A Byzantine-Tolerant State Machine over a PoB Hypergraph

A Quorum Ledger without SMR

Ryuji Ishiguro

April 21, 2026

Abstract

We construct a Byzantine-tolerant state machine whose state is a *Proof-of-Balance* (PoB) hypergraph and whose only transition is graph extension by a single transaction. The machine, denoted S_{POB}, is realized on a Byzantine quorum system in the style of bftkv [1], but without replicated state-machine (SMR) consensus. Every committed transaction carries a threshold-signed certificate attesting that a b-masking quorum has validated the extension; each quorum member retains only a local, possibly partial, projection of the committed graph. We prove that under the Malkhi–Reiter $n > 3b$ threshold [2, 3], the global committed graph is at all times a reachable state of S_{POB}: it satisfies the two combinatorial safety invariants of no double spending and no creation from thin air (Theorem 1), and under the soundness of the zkPoB argument it satisfies a commitment-level global balance identity (Corollary 7). Local sparsity is absorbed into a quantifiable quorum-selection liveness term. The resulting ledger has linear one-shot message complexity $\Theta(n + q)$ and is a drop-in substitute for the blockchain backbone in PoB-based payment systems.

1 Introduction

Most blockchains use a replicated state machine (SMR) to serialize transactions into a totally ordered log. SMR-based systems pay a communication cost that grows with the number of consensus rounds, and they require every replica to hold (or reconstruct) the full ledger. For a payment system whose only safety property is the prevention of double spending, SMR is stronger than necessary: the natural object is a per-commitment *write-once* register, not a linear log.

bftkv [1] showed that a Byzantine fault-tolerant key-value store built on the Malkhi–Reiter b-masking quorum model [2] is sufficient to prevent equivocation, and that a Phalanx-style [3] write protocol (collect-timestamps / collect-signatures / store) yields a scalable ledger without SMR. A recap of the bftkv building blocks used in this paper is in Appendix E.

This paper develops an anonymous payment ledger on top of such a quorum system. The payment layer, called *Proof of Balance* (PoB) and defined in §6, represents balances as homomorphic commitments (encrypted balances, *ebl*) and attaches a zero-knowledge proof to every transaction showing that the total balance is conserved and every output balance is non-negative, without revealing any individual balance. The ledger layer, on top of which PoB runs, is the subject of the rest of this paper: we replace the implicit per-key register model with an explicit *hypergraph state machine*. The global state is a directed hypergraph over commitment vertices and transaction hyperedges, and the only state transition is the append of a single admissible hyperedge. We show that this state machine can be realized on a b-masking quorum system using a two-phase protocol (Preflight + Commit) that corresponds directly to the bftkv get-signature / write pair.

Contributions.

1. A precise hypergraph / bipartite-incidence formalization of the PoB ledger (§2).
2. A state-machine definition S_{POB} whose single transition is admissible-hyperedge extension (§2.4).

3. A Preflight + Commit protocol realization on a b-masking quorum system (§4).
4. A security and complexity analysis reducing PoB safety to the Malkhi–Reiter equivocation bound, plus a hypergeometric liveness term (§5).
5. The PoB payment scheme: a zero-knowledge proof of balance conservation and range (zkPoB, §6.1) together with a four-phase MPC for joint transaction construction (§6.2). A commitment-level global balance identity (Corollary 7) lifts per-transaction soundness to the entire hypergraph.
6. Full safety and liveness proofs in the appendix.

2 The PoB Hypergraph

We model the ledger as a directed hypergraph whose vertices are commitments (encrypted balances) and whose hyperedges are transactions.

2.1 Hypergraph model

Let $\mathcal{E}$ be a set of *commitment vertices* (abstract for now; in the payment scheme of §6 these are the encrypted balances *ebl*) and $\mathcal{T}$ a set of *transaction vertices*. Every transaction $t \in \mathcal{T}$ has:

- an input set $\mathrm{In}(t) \subseteq \mathcal{E}$,
- an output set $\mathrm{Out}(t) \subseteq \mathcal{E}$,
- a PoB validity predicate $\mathrm{PoBValid}(t)$.

A *genesis* subset $\mathsf{Gen} \subseteq \mathcal{E}$ distinguishes commitments that arise from mint events.
A transaction is a directed hyperedge

$$t : \mathrm{In}(t) \longrightarrow \mathrm{Out}(t).$$

A *graph* $G = (E_G, T_G)$ has commitment vertices $E_G \subseteq \mathcal{E}$ and transaction vertices $T_G \subseteq \mathcal{T}$, closed under incidence: for every $t \in T_G$, $\mathrm{In}(t) \cup \mathrm{Out}(t) \subseteq E_G$.

2.2 Bipartite incidence encoding

For proofs we use the equivalent directed bipartite incidence graph $B(G)$ with vertex set $E_G \sqcup T_G$, an arc $e \to t$ for every $e \in \mathrm{In}(t)$, and an arc $t \to e$ for every $e \in \mathrm{Out}(t)$. $B(G)$ is bipartite and acyclic, and extension is append-only.

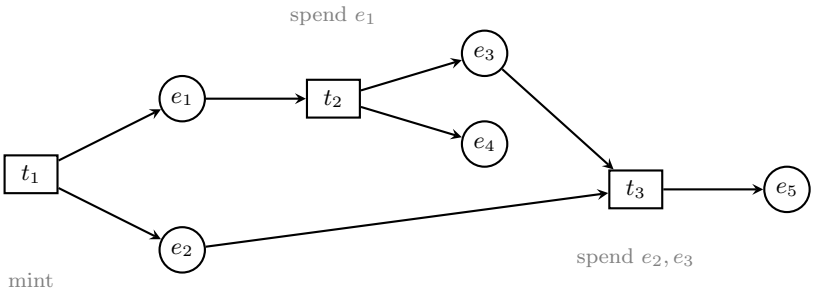

Figure 1: Bipartite incidence graph $B(G)$ after applying t_1, t_2, t_3 to $G_0 = (\emptyset, \emptyset)$. Circles are commitments E_G; squares are transactions T_G. Arcs $t \to e$ indicate creation ($e \in \mathrm{Out}(t)$); arcs $e \to t$ indicate consumption ($e \in \mathrm{In}(t)$). The mint t_1 has no commitment inputs (its outputs e_1, e_2 enter Gen); t_2 consumes e_1 and produces e_3, e_4; t_3 consumes e_2 and e_3 and produces e_5. After these three steps, $\mathrm{Frontier}(G) = \{e_4, e_5\}$: the only commitments produced but not yet consumed.

The causal partial order $\prec_G$ on T_G is the transitive closure of the arcs in Figure 1: $t_1 \prec t_2$ (via e_1), $t_1 \prec t_3$ (via e_2), and $t_2 \prec t_3$ (via e_3), giving the total chain $t_1 \prec t_2 \prec t_3$ here. In general the causal order is only partial: if t_2 and t_3 had disjoint inputs and outputs they would be concurrent, and either $G_0 \oplus t_1 \oplus t_2 \oplus t_3$ or $G_0 \oplus t_1 \oplus t_3 \oplus t_2$ would produce the same final graph.

Define the partial functions $\text{creator}, \text{consumer} : E_G \rightharpoonup T_G$ by:

$$\begin{aligned} \text{creator}(e) = t &\iff e \in \text{Out}(t), \quad t \in T_G, \\ \text{consumer}(e) = t &\iff e \in \text{In}(t), \quad t \in T_G. \end{aligned}$$

"creator(e) is defined" means creator(e) $\in T_G$ exists for e (i.e., some transaction in G produced e); similarly for consumer(e). Define the *frontier*:

$$\text{Frontier}(G) \;=\; \big\{\, e \in E_G \;\big|\; (e \in \mathsf{Gen} \;\vee\; \text{creator}(e) \in T_G) \;\wedge\; \nexists\, t \in T_G \colon e \in \text{In}(t) \,\big\}.$$

In words: e is on the frontier iff e was produced (by a transaction in G or by genesis) and has not yet been consumed by any transaction in G.

2.3 Admissible extension

A candidate transaction t is *admissible* over G, written $\text{Adm}(G, t)$, iff all of the following hold:

1. PoBValid(t);
2. In(t) has no duplicates;
3. Out(t) has no duplicates;
4. $\text{In}(t) \cap \text{Out}(t) = \emptyset$;
5. inputs are on the frontier: $\text{In}(t) \subseteq \text{Frontier}(G)$;
6. outputs are fresh: for every $e \in \text{Out}(t)$, $e \notin E_G$.

By the frontier definition of §2.2, every $e \in \text{In}(t) \setminus \mathsf{Gen}$ is then automatically produced by a unique creator(e) $\in T_G$, so clause 5 alone ensures every non-genesis input is bound to an existing creator in G.

Define the extension operator Extend by

$$\text{Extend}(G, t) = \big(E_G \cup \text{Out}(t),\; T_G \cup \{t\}\big),$$

with the induced incidence arcs.

Theorem 1 (Graph Safety)**.** *Let $G_0 = (\emptyset, \emptyset)$. If for every $i \geq 0$ we have $G_{i+1} = \text{Extend}(G_i, t_i)$ with $\text{Adm}(G_i, t_i)$, then every G_i satisfies:*

1. ***No double spending:*** *every $e \in E_{G_i}$ is consumed by at most one transaction in T_{G_i};*
2. ***No creation from thin air:*** *every non-genesis input of every $t \in T_{G_i}$ is a commitment already present in E_{G_i} and produced by some transaction in T_{G_i}.*

Theorem 1 is proved by induction in Appendix A. It is an invariant of the graph model alone, independent of any quorum system or of any cryptographic property of PoBValid. A third invariant, global balance conservation, does depend on the algebraic structure of the PoB commitments and the soundness of zkPoB: it is stated as Corollary 7 in §6.

2.4 The S_{POB} state machine

Definition 2. *The* PoB state machine *is the transition system*

$$S_{POB} = (\mathcal{G},\ G_0,\ \mathrm{Step}), \qquad G_0 \ =\ (\emptyset,\ \emptyset),$$

with transition function

$$\mathrm{Step}(G,t) \ =\ \begin{cases} \mathrm{Extend}(G,t) & \text{if } \mathrm{Adm}(G,t), \\ G & \text{otherwise.} \end{cases}$$

The reachable-state set $\mathcal{G}$ is defined recursively as the smallest set of graphs satisfying

$$G_0 \in \mathcal{G}, \qquad G \in \mathcal{G} \ \wedge\ \mathrm{Adm}(G,t) \implies \mathrm{Extend}(G,t) \in \mathcal{G}.$$

We introduce the partial binary operator $\oplus : \mathcal{G} \times \mathsf{Tx} \rightharpoonup \mathcal{G}$ defined by $G \oplus t = \mathrm{Extend}(G,t)$ whenever $\mathrm{Adm}(G,t)$ holds, and undefined otherwise. Every $G \in \mathcal{G}$ is then obtained from G_0 by a finite sequence of $\oplus$ applications:

$$G \ =\ G_0 \oplus t_1 \oplus t_2 \oplus \cdots \oplus t_k,$$

with each t_i admissible against $G_0 \oplus t_1 \oplus \cdots \oplus t_{i-1}$. The operator $\oplus$ is not *commutative in general: if t_j consumes an output produced by t_i, then $G \oplus t_j$ is undefined (via* Adm*) while $G \oplus t_i \oplus t_j$ is defined. Commutativity holds only between* causally independent *transactions — pairs (t_i, t_j) with $\mathrm{In}(t_i) \cap \mathrm{In}(t_j) = \emptyset$, $\mathrm{Out}(t_i) \cap \mathrm{In}(t_j) = \emptyset$, and $\mathrm{Out}(t_j) \cap \mathrm{In}(t_i) = \emptyset$.*

Consequently $\mathcal{G}$ does not carry a total order on transactions: the trajectory prefixes $G_0 \oplus t_1 \oplus \cdots$ and $G_0 \oplus t'_1 \oplus \cdots$ that reorder causally independent transactions yield the same *final graph. Formally, every $G \in \mathcal{G}$ induces a partial order $\prec_G$ on T_G given by the transitive closure of "$t_i \prec t_j$ if $\mathrm{Out}(t_i) \cap \mathrm{In}(t_j) \neq \emptyset$" (causal dependence). Any topological linearization of $\prec_G$ is a valid construction sequence for G under $\oplus$; concurrent transactions are free to be ordered arbitrarily.*

Genesis commitments are not part of the initial state; they enter E_G exclusively as outputs of admissible mint *transactions, which are transactions with $\mathrm{In}(t) = \emptyset$ and $\mathrm{Out}(t) \subseteq \mathsf{Gen}$. All other ("ordinary") transactions have non-empty inputs that are already on the frontier (clause 5) and outputs disjoint from Gen. Once a genesis commitment is consumed, it is consumed as an ordinary frontier element; a transaction cannot simultaneously be a mint and consume a genesis input. Any structure involving multiple genesis vertices is therefore an emergent property of* Adm *applied to G_0, not a built-in feature of the initial graph.*

This section is pure graph theory: $\mathcal{G}$ is a mathematical object defined purely by the admissibility predicate Adm of §2.3, with no reference to how graphs are stored or which party validates transactions. Realization of S_{POB} by a distributed protocol is the subject of §3 onwards.

By Theorem 1, every reachable state of S_{POB} is a valid PoB hypergraph. The rest of this paper is about realizing S_{POB} on a Byzantine quorum system.

3 Quorum Model

We adopt the Byzantine quorum setting of bftkv [1] and the b-masking quorum system of Malkhi–Reiter [2, 3].

3.1 Fault model and quorum sizes

Let U be a finite set of quorum members with $|U| = n$. At most b members are Byzantine; the remaining $n - b$ are honest. We assume:

- $n \geq 3b + 1$ (equivalently $n > 3b$);
- authenticated communication between any client and any member;
- durable local storage at each honest member;

- *partial synchrony* [6] (*liveness assumption only*): after some unknown global stabilization time (GST), message delivery between correct parties is bounded. This matches the standard deployment setting for practical BFT protocols (PBFT, HotStuff, bftkv) and rules out indefinite network outages from the liveness analysis. Safety (§5.1) does *not* require partial synchrony and holds under arbitrary asynchrony;
- *client persistence* (*liveness assumption only*): a correct client that initiates Commit for t retries until q acks are collected. Combined with partial synchrony, this implies every signed $(t, \sigma(t))$ eventually reaches q durable copies;
- a threshold-signature scheme that produces, from the signed contributions of a set $S \subseteq U$, a single short signature that verifies iff $|S| \geq \tau$, and that is existentially unforgeable under chosen-message attack against any coalition of strictly fewer than τ corrupted contributors. In particular, the b Byzantine members alone cannot forge $\sigma(t)$, and a valid threshold signature implies at least $\tau - b$ distinct honest contributors.

A *quorum* is any subset $Q \subseteq U$ with $|Q| = q$, where we fix the quorum size

$$q \;=\; n - b.$$

By the b-masking intersection property, any two quorums satisfy

$$|Q_1 \cap Q_2| \;\geq\; 2q - n \;=\; n - 2b \;\geq\; b + 1,$$

so every pair of quorums contains at least one honest member in common [2]. In the canonical BFT case $n = 3b + 1$ this gives $q = 2b + 1$ and $|Q_1 \cap Q_2| \geq b + 1$, matching the thresholds used by bftkv [1, §4].

3.2 Global and local graphs

Each honest member r maintains a *local graph* G_r. The *global graph* realized by the quorum protocol is the union

$$G^* \;=\; \bigcup_{r \in H} G_r,$$

over the set H of honest members. G^* is not stored at any single node; a transaction appears in G^* as soon as at least one honest member has stored it.

The central claim of this paper is that G^* is a reachable state of the abstract state machine S_{POB} of §2.4, i.e.,

$$G^* \in \mathcal{G}.$$

Since $\mathcal{G}$ is defined recursively as graphs reachable from G_0 by admissible extension, $G^* \in \mathcal{G}$ means the sequence of transactions appearing in G^* corresponds to an admissible trajectory of S_{POB}. This identifies the distributed realization with the abstract state machine and is proved as Theorem 3 (§5.1) under the partial-synchrony and client-persistence assumptions of §3.

4 Protocol

The protocol has two phases. In Preflight the client collects a threshold-signed admissibility certificate from a signing quorum Q_p. In Commit the client disseminates the signed transaction to a store quorum Q_c. These phases mirror the get-signature and write operations of bftkv [1, §4], but operate on graph extensions rather than per-key writes.

4.1 Notation

Let C denote the client, Q a quorum, and $r \in Q$ a quorum member. We write $C \to Q$ for a client-to-quorum broadcast and $C \leftarrow Q_i$ for the reply of a single member. We write $\sigma_r(m)$ for r's partial signature on message m, and $\sigma(m) = \mathsf{Combine}(\{\sigma_r(m)\}_{r \in S})$ for the threshold signature obtained from a set S of at least τ partial signatures.

4.2 Preflight

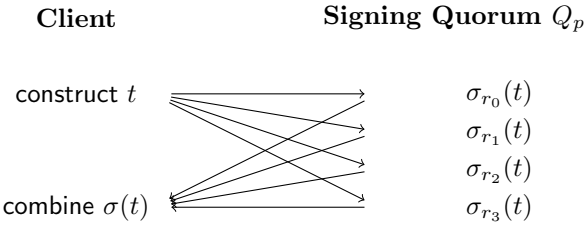

Figure 2: Preflight protocol.

Client side: choose $Q_p \subseteq U$ with $|Q_p| = q$, send t to every $r \in Q_p$, collect partial signatures from at least τ honest members, and combine them into the threshold signature $\sigma(t)$. If fewer than τ partial signatures are collected, Preflight aborts and the client may retry with a freshly chosen quorum.

Member side is given by Algorithm 1. The member checks that t is admissible against its local graph G_r before signing.

Algorithm 1: Member r on Preflight(t)

on Preflight(t)**:**
  verify PoBValid(t);
  verify clauses 2–4 of Adm (local to t);
  if *not (*$\mathrm{In}(t) \subseteq \mathrm{Frontier}(G_r)$*)* **then**
    return reject; `// input missing, consumed, or not produced in` G_r
  end
  if *some* $e \in \mathrm{Out}(t)$ *already in* E_{G_r} **then**
    return reject; `// output collision`
  end
  mark $\mathrm{In}(t)$ as tentatively consumed in G_r;
  return $\sigma_r(t)$;
end

The tentative mark on $\mathrm{In}(t)$ prevents r from signing any conflicting transaction while t's commit is still in flight. Marks are durable: they survive restarts and are cleared only on receipt of a committed transaction that consumes or supersedes the same input, or by an out-of-band recovery procedure (§4.3). They are *not* cleared by a local timeout; any such timeout would reopen the double-sign window that Appendix B closes.

Mark retention is an operational assumption. Because marks are not expired automatically, a malicious client can saturate an honest member's mark table by running Preflight on many distinct transactions with overlapping inputs that it never commits, and an honest client that crashes between Preflight and Commit may leave marks that only clear via peer-driven recovery (§4.3). We do not analyse these behaviours in the safety or liveness theorems; the assumption is that the deployment imposes operational limits — per-client rate limiting, bounded in-flight signed candidates per member, and a recovery protocol that eventually finishes every Preflight that collected τ contributions — so that the mark table remains bounded in the honest-client case and DoS by a Byzantine client is rate-limited at admission. All such policies preserve safety (they only *add* reasons to reject a candidate); they affect liveness only.

The signing threshold τ is fixed at

$$\tau \;=\; q \;=\; n - b.$$

§5 and Appendix B.1 show that any strictly smaller threshold is unsafe. This matches the bftkv write threshold.

4.3 Commit

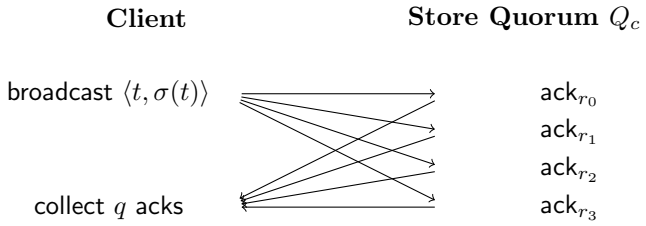

Figure 3: Commit protocol.

Client side: choose $Q_c \subseteq U$ with $|Q_c| = q$, broadcast $\langle t, \sigma(t) \rangle$ to every $r \in Q_c$, and wait for q acknowledgements. If fewer than q acks arrive before a client-side retry deadline, the client resends $\langle t, \sigma(t) \rangle$ to a freshly chosen Q_c. Because $\sigma(t)$ is non-expiring, the client may retry indefinitely; members that already stored t reply ack idempotently. Commit is declared successful only after q acks have been received.

Member side is given by Algorithm 2.

Algorithm 2: Member r on Commit(t, σ)

```
on Commit(t, σ):
    verify σ is a valid threshold signature on t;
    if t ∈ T_{G_r} then
        return ack;                                          // idempotent
    end
    verify PoBValid(t);
    append t to G_r: set E_{G_r} ← E_{G_r} ∪ Out(t), T_{G_r} ← T_{G_r} ∪ {t} durably;
    clear any tentative marks on In(t);
    return ack;
end
```

Members appending t on Commit do *not* re-check admissibility against their own local graph. The threshold signature certifies that at least $\tau = n - b$ members each verified admissibility of t against their own local G_r during Preflight; of these, at least $\tau - b = n - 2b$ are honest. The global safety theorem (§5.1) establishes that this local-per-member guarantee extends to admissibility against G^* itself. Commit-time members therefore trust threshold-signed history without re-verifying against their own G_r.

Why Commit acks are safety-critical. Safety requires that every committed transaction be durably stored at at least one honest member before any conflicting transaction can be signed. Since $|Q_c| = q = n - b$, among the q acks at least $q - b = n - 2b \geq b + 1$ come from honest members, so at least one honest member has durably stored t. By the b-masking intersection property, any future signing quorum Q'_p intersects Q_c in at least one honest member, who now holds t in its local graph and will reject any transaction conflicting with t under Algorithm 1. Thus once Commit succeeds, no future conflicting transaction can pass Preflight.

Client failure between Preflight and Commit. If the client collects $\sigma(t)$ but never completes Commit (e.g., the client crashes or a network partition isolates it), the honest members that signed Preflight retain their tentative marks on $\mathrm{In}(t)$. Any other party holding $\sigma(t)$ can act as the committer: Commit is a pure dissemination of the signed object and requires no state from the original client. In particular, members in Q_p may themselves re-broadcast $\langle t, \sigma(t) \rangle$ to finish the transaction and clear their marks, once they combine the partial signatures they have overheard from peers in Q_p (or are handed by a peer). The paper does not specify a particular recovery policy; any policy that eventually produces q durable copies of t is safe.

5 Security and Complexity Analysis

We analyze safety, liveness, and communication complexity.

5.1 Safety

Safety follows from the Malkhi–Reiter equivocation argument applied to the threshold-signature step of Preflight.

Theorem 3 (Global Graph Safety)**.** *Assume $n \geq 3b + 1$, $q = n - b$, $\tau = n - b$, and $G_0 = (\emptyset, \emptyset)$. Under the protocol of §4 and the partial-synchrony plus client-persistence assumptions of §3, the global graph $G^* = \bigcup_{r \in H} G_r$ is a reachable state of S_{POB}:*

$$G^* \in \mathcal{G}.$$

In particular, G^* satisfies the two graph-safety invariants of Theorem 1, and under the soundness of zkPoB also the commitment-level global balance of Corollary 7.

The proof is in Appendix B. The core is the bftkv equivocation bound [1, §5]: for two conflicting transactions t_1, t_2, the honest signers split disjointly into two sets whose sizes sum to at most $n - b$, while each side must also satisfy the threshold τ. This yields $2\tau \leq n + b$, contradicting $\tau = n - b$ under $n > 3b$. The choice $\tau = n - b$ is tight (Appendix B.1).

5.2 Liveness

Liveness has two failure modes. The first is *Preflight failure*: an honest member cannot sign t because, for some non-genesis input $e \in \mathrm{In}(t)$, $e \notin \mathrm{Frontier}(G_r)$ — the creator transaction of e has not yet propagated to r's local graph. The second is *Commit failure*: the client cannot reach q members in Q_c (e.g., a transient partition). Safety does not depend on either phase succeeding, but a transaction is considered committed only after Commit produces q durable acks. Under total network unavailability, liveness is zero until the network heals; we restrict the analysis to the *reachable* setting where the client can deliver and receive messages.

Preflight failure

For a valid candidate t, let $F_t \subseteq U$ be the set of members that cannot sign t: the union of Byzantine members that refuse, and honest members r for which $\mathrm{In}(t) \not\subseteq \mathrm{Frontier}(G_r)$ (at least one input's creator transaction has not yet propagated to r). Write $u_t = |F_t|$.

If the client chooses the signing quorum Q_p uniformly at random among size-q subsets of U, the number of non-signers in Q_p is hypergeometric:

$$Y_t \sim \mathrm{Hypergeometric}(n, u_t, q).$$

Theorem 4 (Preflight Liveness)**.** *Under the assumptions of Theorem 3, for every valid candidate t the probability that a single Preflight attempt fails is*

$$\Pr[\textit{pre-fail at } t] \;=\; \sum_{i=q-\tau+1}^{\min(u_t,\, q)} \frac{\binom{u_t}{i}\binom{n-u_t}{q-i}}{\binom{n}{q}}.$$

In particular, if $u_t \leq q - \tau$ then $\Pr[\textit{pre-fail at } t] = 0$.

Proof in Appendix C.

Figure 4 visualizes Theorem 4: as the quorum size q grows past $\tau + u_t$, the probability of failure collapses to zero because every size-q subset must contain at least τ signers. In the canonical BFT parameterization with $\tau = q = n - b$ the slack is zero, so reliability is bought entirely by pushing u_t down via wider dissemination of committed transactions.

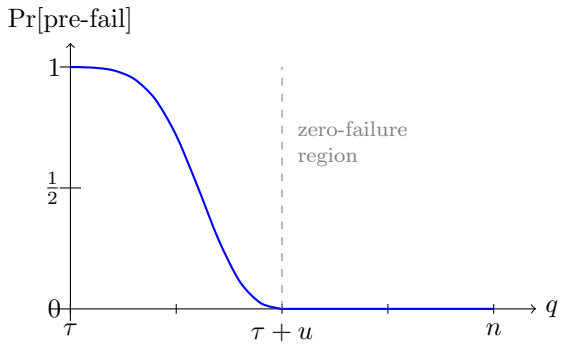

Figure 4: Preflight failure probability as a function of quorum size q, for fixed $n = 40$, $u = 10$ non-signers, threshold $\tau = 20$. The curve is the hypergeometric upper tail $\Pr[Y > q-\tau]$ with $Y \sim \text{Hypergeometric}(n, u, q)$. It has the characteristic S-shape: slow decay while the slack $q-\tau$ is small, sharp inflection around $q \approx \tau+u/2$, and exact zero once $q \geq \tau + u$ since no size-q subset can contain more than u non-signers. As $q \to n$ the curve stays pinned at zero. This is the threshold phenomenon of the b-masking quorum system.

Commit failure

Let $\mu \subseteq U$ denote the set of members that are currently *unreachable* to the client, with $|\mu| = m$. Among these, at most b may be Byzantine (silently refusing), and the remaining $m-b$ (possibly zero) are honest but unreachable due to partition, crash, or load. Under a uniformly random choice of Q_c of size q, the number of reachable members in Q_c is

$$R_t \sim \text{Hypergeometric}(n, n-m, q),$$

and Commit succeeds in one attempt iff $R_t \geq q$, i.e., iff $Q_c \subseteq U \setminus \mu$.

Theorem 5 (Commit Liveness)**.** *A single Commit attempt succeeds with probability*

$$\Pr[\textit{commit-ok}] \;=\; \frac{\binom{n-m}{q}}{\binom{n}{q}}$$

if $n - m \geq q$, and with probability 0 otherwise. In particular if $m > n - q = b$ no single quorum can be fully reached and Commit liveness is zero for the duration of the partition.

Note $n-q = b$: the system tolerates the loss of up to b unreachable members for both safety *and* liveness, matching the Byzantine bound. Beyond that threshold, safety is preserved (no conflicting t' can obtain $\sigma(t')$ because Preflight also requires q signers, and safety of the committed prefix is unaffected by failure to commit new transactions) but progress stalls until enough members become reachable.

End-to-end success

The probability that one end-to-end attempt (Preflight then Commit) succeeds is

$$\Pr[\text{ok}] \;=\; \big(1 - \Pr[\text{pre-fail}]\big) \cdot \Pr[\text{commit-ok}],$$

assuming independent fresh quorum draws — formally, that each retry samples Q_p (and Q_c) uniformly and independently from size-q subsets of U, without conditioning on previously drawn sets. In practice the client obtains independence by rerandomizing over the full membership U on each attempt; if retries reuse or partially overlap prior quorums, the success probabilities become correlated and the single-attempt bound above is only a lower bound on the per-attempt success rate. The expected number of end-to-end attempts until success is $1/\Pr[\text{ok}]$ whenever $\Pr[\text{ok}] > 0$. If the network is partitioned such that $m > b$, $\Pr[\text{ok}] = 0$ and liveness is zero for the partition duration; safety is unaffected. Because $\sigma(t)$ does not expire, a client holding $\sigma(t)$ can always complete Commit once enough members become reachable.

Theorem 4 and Theorem 5 give a clean separation: safety depends only on the $n > 3b$ threshold and is unconditional; liveness is a pair of hypergeometric tails that the client amortizes by retry with fresh random quorums.

A commit-coverage bound. Let k be a lower bound on the number of members that receive and append every committed transaction. Then the number of honest members missing a direct parent of t is at most $n - k$, and a worst-case bound on u_t is

$$u_t \;\le\; (n-k) + b.$$

Plugging into Theorem 4 gives an *a priori* liveness guarantee: if the store quorum consistently reaches a majority of members, u_t stays small and Pr[fail at t] remains bounded.

5.3 Communication complexity

All bounds are in messages; round complexity is 2 on the success path.

Proposition 6 (One-shot complexity)**.** *Under the parameters $n \ge 3b+1$, $|Q_p| = |Q_c| = q = n - b$, and $\tau = n - b$, one successful transaction commit costs*

$$M_{\text{one}}(t) \;=\; 2q + 2q \;=\; 4q$$

messages on the client-to-quorum path: q Preflight requests, q Preflight replies, q Commit broadcasts, and q Commit acknowledgements (required for safety, §4.3).

In the canonical BFT case $n = 3b+1$, $q = 2b+1$:

$$M_{\text{one}}(t) \;=\; 4(2b+1) \;=\; 8b+4.$$

Asymptotically, since $q \le n$,

$$M_{\text{one}}(t) \;=\; \Theta(n+q) \;=\; \Theta(n).$$

Let $p_t = 1 - \Pr[\text{ok}]$ be the end-to-end attempt-failure probability from §5.2. Under independent fresh quorum draws,

$$\mathbb{E}[M_{\text{total}}(t)] \;=\; \frac{4q}{1-p_t}.$$

Bandwidth. Per-message cost is dominated by the transaction payload. The Preflight request and the Commit broadcast each carry t including $|\pi_t|$, the size of the zkPoB proof. With d-bit balance ranges, $|\operatorname{In}(t)| + |\operatorname{Out}(t)|$ commitments, and a BP++-family argument [8], $|\pi_t|$ is $O(d + \log(|\operatorname{In}(t)| + |\operatorname{Out}(t)|))$ group elements. One-shot bandwidth is therefore $O\big(n \cdot |\pi_t|\big)$ bits; Preflight replies and Commit acks are $O(1)$ per member.

6 The PoB Payment Scheme

The *Proof of Balance* (PoB) payment scheme runs on top of the hypergraph state machine S_{POB}. An account holds a homomorphic commitment

$$ebl \;=\; h^{\beta} g^{bl}$$

to a balance $bl \in \mathbb{Z}_{\ge 0}$ under a blinding factor $\beta \in \mathbb{Z}_p$, where g, h are fixed generators of a prime-order group $\mathbb{G}$ of order p, with $\log_g h$ unknown. A transaction transitions a set of *input commitments* $\{ebl_i\}_i$ into a set of *output commitments* $\{ebl'_i\}_i$, reallocating value without revealing any individual balance. Anonymity follows because every commitment is perfectly hiding and no long-lived identity is attached to any individual ebl; double-spending is prevented at the ledger layer by the frontier condition (clause 5 of §2.3).

The commitment vertex set $\mathcal{E}$ of §2 is the set of encrypted balances $\{ebl\}$; each transaction hyperedge is the pair of commitment sets plus a zero-knowledge proof π_t that binds them. The predicate PoBValid(t) of §2 is instantiated here as the verifier's acceptance of π_t against $(\mathrm{In}(t), \mathrm{Out}(t))$; throughout §§4–5 and the appendix, any algorithmic check "verify PoBValid(t)" means "run the zkPoB verifier on π_t with inputs $(\mathrm{In}(t), \mathrm{Out}(t))$ and accept iff the verifier accepts". We spell out the proof relation in §6.1, describe how a transaction is jointly constructed by payer and payee in §6.2, and submit the finished transaction to Preflight + Commit (§4) for admission to the global graph.

6.1 Zero-Knowledge Proof of Balance (zkPoB)

PoBValid(t) is a zero-knowledge proof that a transaction conserves value across its inputs and outputs and that every output balance is non-negative, without revealing any individual balance. Write the inputs of t as $\{ebl_i = h^{\beta_i} g^{bl_i}\}_i$ and its outputs as $\{ebl'_i = h^{\beta'_i} g^{bl'_i}\}_i$. The PoB relation is

$$\begin{aligned}\mathcal{R} := \big\{ \big((bl_i, bl'_i, \beta_i, \beta'_i)_i,\ (ebl_i, ebl'_i)_i\big) : \\ ebl_i = h^{\beta_i} g^{bl_i},\ ebl'_i = h^{\beta'_i} g^{bl'_i}, \\ 0 \le bl'_i < 2^d,\ \text{and}\ \sum_i bl_i = \sum_i bl'_i \big\}.\end{aligned}$$

By homomorphism,

$$(ebl_i, ebl'_i) \in L_{\mathcal{R}} \iff \prod_i ebl_i \,/\, \prod_i ebl'_i \;=\; \prod_i h^{\beta_{\triangle i}}, \quad \beta_{\triangle i} = \beta_i - \beta'_i,$$

iff $\sum_i bl_i = \sum_i bl'_i$, so it suffices to prove knowledge of the blinding deltas and a valid range decomposition of each output balance. As an explanatory derivation this can be cast as a sigma protocol on the relation

$$\begin{aligned}\mathcal{R}' := \big\{ \big((\beta_{\triangle i}, bl'_i, \gamma_i),\ (u_j, w_i, b_{ij}, c_{ij})\big) : \\ w_i = h^{\beta_{\triangle i}},\ c_{ij} = u_j^{\gamma_i} g^{b_{ij}} \big\},\end{aligned}$$

where $u_j = g^{\alpha_j}$, $\alpha_j \xleftarrow{\mathcal{R}} \mathbb{Z}_p$, subject to $\prod_i ebl_i / \prod_i ebl'_i = \prod_i w_i$ and $bl'_i = \sum_{j=0}^{d-1} 2^j b_{ij}$. The prover shows knowledge of the witness via the three-move protocol in Figure 5. The verifier concludes by checking $\prod_i ebl_i / \prod_i ebl'_i \stackrel{?}{=} \prod_i w_i$.

In practice the range subclaim is instantiated with a Bulletproofs-family argument: a BP++-style reciprocal argument [8] over an arithmetic circuit, together with a WNLA inner argument [7]. The prover commits to each output balance, proves the committed value lies in $[0, 2^d)$, and binds the range proof to the encrypted balance commitment carried in the transaction. The sigma-protocol derivation above explains the algebraic relation; the concrete proof object π_t attached to t is a compact non-interactive argument obtained via Fiat–Shamir.

Cryptographic assumptions. Soundness and zero-knowledge of zkPoB rest on the discrete-logarithm assumption in $\mathbb{G}$ (where g, h are group generators with no known relation, i.e., $\log_g h$ is not known to any party) and the random-oracle heuristic for the Fiat–Shamir transformation. Knowledge-soundness follows from the standard rewinding argument for sigma protocols composed with a Bulletproofs-family range proof; soundness error is negligible in the security parameter. Corollary 7 and Theorem 3's appeal to zkPoB soundness are both under these assumptions.

A transaction is therefore abstractly

$$t \;=\; \big(\{ebl_i\}_{i \in \mathrm{In}(t)},\ \{ebl'_i\}_{i \in \mathrm{Out}(t)},\ \pi_t\big),$$

and PoBValid(t) is the verifier's acceptance of π_t against the stated input / output commitments. Any ebl_i or ebl'_i may be nil, treated as $bl = 0$.

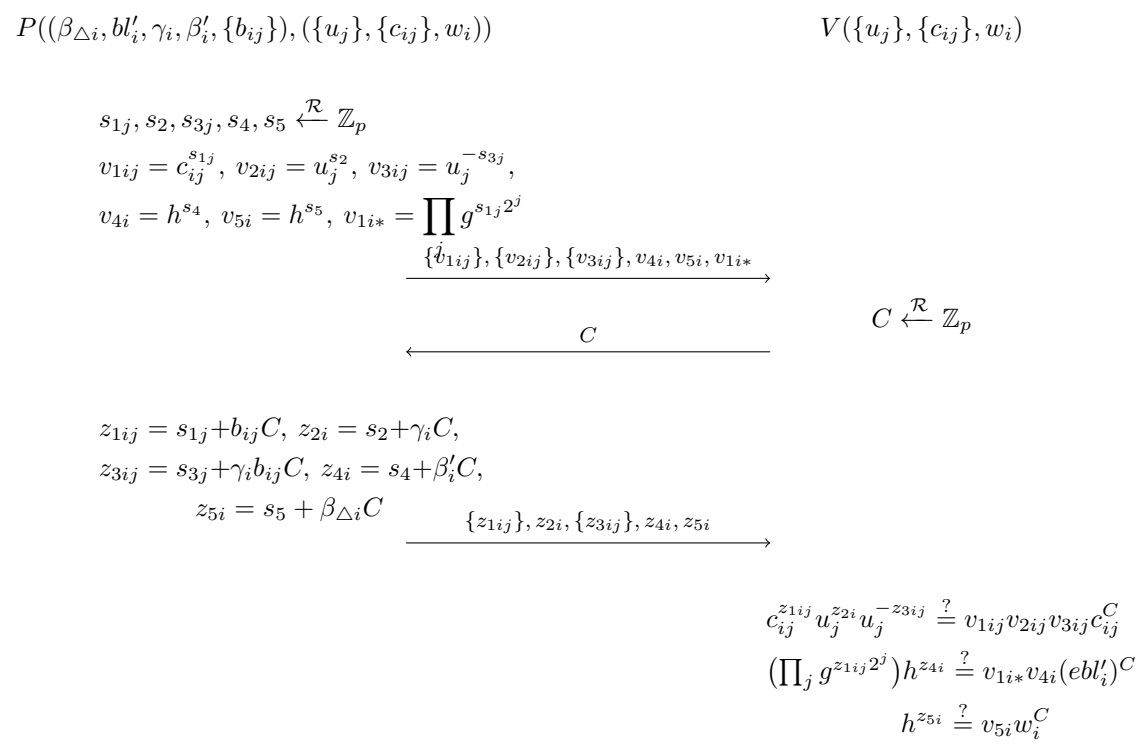

Figure 5: Sigma-protocol derivation of the PoB relation. In practice the range subclaim is instantiated with a Bulletproofs-family proof and the whole protocol is compiled to a non-interactive argument via Fiat–Shamir.

Global balance consistency. The PoB relation of §6.1 is multiplicatively composable on commitments, so the per-transaction identity of Theorem 1 lifts to a statement about the *global* graph at the cryptographic level. Define

$$\Pi(S) \;=\; \prod_{e \in S} e \qquad (S \subseteq \mathcal{E}),$$

the product of commitments over a subset S; note $\Pi(S) = h^{\sum \beta_e} g^{\sum \mathrm{bal}(e)}$ by the homomorphism of Pedersen commitments.

We write $\mathrm{bal}(e)$ and β_e for the balance and blinding factor of a commitment $e = h^{\beta_e} g^{\mathrm{bal}(e)}$. Recall the mint / ordinary distinction from §2.4: mints have $\mathrm{In}(t) = \emptyset$ and $\mathrm{Out}(t) \subseteq \mathsf{Gen}$; ordinary transactions have $\mathrm{In}(t) \subseteq \mathrm{Frontier}(G)$ and $\mathrm{Out}(t) \cap \mathsf{Gen} = \emptyset$, and a genesis commitment consumed as an input is consumed as an ordinary frontier element. For a mint, $\mathrm{PoBValid}(t)$ carries range proofs on $\mathrm{Out}(t)$ only; value conservation is vacuously true.

Corollary 7 (Commitment-level global balance)**.** *Let G_i be a reachable state of S_{POB} obtained along the trajectory $G_0 \oplus t_0 \oplus \cdots \oplus t_{i-1}$, and assume* $\mathrm{PoBValid}(t_j)$ *holds for every $j < i$ (under soundness of the zkPoB argument of §6.1). Define*

$$\Delta_i \;=\; \sum_{\substack{j<i \\ t_j \ ordinary}} \Big(\sum_{e' \in \mathrm{Out}(t_j)} \beta_{e'} - \sum_{e \in \mathrm{In}(t_j)} \beta_e \Big).$$

Then, at the commitment level,

$$\Pi\big(\mathrm{Frontier}(G_i)\big) \;=\; \Pi\big(\mathsf{Gen} \cap E_{G_i}\big) \cdot h^{\Delta_i}, \tag{1}$$

and equivalently, at the balance level,

$$\sum_{e \in \mathrm{Frontier}(G_i)} \mathrm{bal}(e) \;=\; \sum_{e \in \mathsf{Gen} \cap E_{G_i}} \mathrm{bal}(e). \tag{2}$$

Proof. Write

$$N_i \;=\; \frac{\Pi(\mathrm{Frontier}(G_i))}{\Pi(\mathsf{Gen} \cap E_{G_i})}.$$

We show by induction on i that $N_i = h^{\Delta_i}$, which is (1).

Base case. $G_0 = (\emptyset, \emptyset)$, both products are the empty product 1, and $\Delta_0 = 0$, so $N_0 = 1 = h^0$.

Inductive step: mint transition. Suppose t_i is a mint, so $\mathrm{In}(t_i) = \emptyset$ and $\mathrm{Out}(t_i) \subseteq \mathsf{Gen}$. By clause 6 of Adm (outputs fresh), $\mathrm{Out}(t_i) \cap E_{G_i} = \emptyset$. Therefore

$$\begin{aligned} \mathrm{Frontier}(G_{i+1}) &= \mathrm{Frontier}(G_i) \sqcup \mathrm{Out}(t_i), \\ \mathsf{Gen} \cap E_{G_{i+1}} &= (\mathsf{Gen} \cap E_{G_i}) \sqcup \mathrm{Out}(t_i). \end{aligned}$$

Hence $\Pi(\mathrm{Frontier}(G_{i+1}))/\Pi(\mathsf{Gen} \cap E_{G_{i+1}})$ is unchanged: $N_{i+1} = N_i$. Since mints contribute nothing to Δ, $\Delta_{i+1} = \Delta_i$, so $N_{i+1} = h^{\Delta_i} = h^{\Delta_{i+1}}$.

Inductive step: ordinary transition. Suppose t_i is ordinary, so $\mathrm{In}(t_i) \subseteq \mathrm{Frontier}(G_i) \setminus \mathsf{Gen}$ (clause 5 plus the mint convention) and $\mathrm{Out}(t_i) \cap (E_{G_i} \cup \mathsf{Gen}) = \emptyset$ (clause 6 and $\mathrm{Out}(t_i) \cap \mathsf{Gen} = \emptyset$). Then

$$\begin{aligned} \mathrm{Frontier}(G_{i+1}) &= \big(\mathrm{Frontier}(G_i) \setminus \mathrm{In}(t_i)\big) \sqcup \mathrm{Out}(t_i), \\ \mathsf{Gen} \cap E_{G_{i+1}} &= \mathsf{Gen} \cap E_{G_i}. \end{aligned}$$

Hence

$$N_{i+1} \;=\; N_i \cdot \frac{\Pi(\mathrm{Out}(t_i))}{\Pi(\mathrm{In}(t_i))}.$$

By soundness of zkPoB, π_{t_i} witnesses the relation $\mathcal{R}$ on $(\mathrm{In}(t_i), \mathrm{Out}(t_i))$; in particular, $\sum_{e \in \mathrm{In}(t_i)} \mathrm{bal}(e) = \sum_{e' \in \mathrm{Out}(t_i)} \mathrm{bal}(e')$. By the homomorphism of Pedersen commitments,

$$\frac{\Pi(\mathrm{Out}(t_i))}{\Pi(\mathrm{In}(t_i))} \;=\; h^{\,\delta_i} \cdot g^{\,\sum_{\mathrm{Out}(t_i)} \mathrm{bal}(e') - \sum_{\mathrm{In}(t_i)} \mathrm{bal}(e)} \;=\; h^{\,\delta_i},$$

where $\delta_i = \sum_{e' \in \mathrm{Out}(t_i)} \beta_{e'} - \sum_{e \in \mathrm{In}(t_i)} \beta_e$ is exactly the increment $\Delta_{i+1} - \Delta_i$ from the definition of Δ. Therefore $N_{i+1} = h^{\Delta_i} \cdot h^{\delta_i} = h^{\Delta_{i+1}}$.

Balance identity. Applying the Pedersen homomorphism to both sides of (1),

$$h^{\sum_{\mathrm{Frontier}(G_i)} \beta_e}\, g^{\sum_{\mathrm{Frontier}(G_i)} \mathrm{bal}(e)} \;=\; h^{\sum_{\mathsf{Gen} \cap E_{G_i}} \beta_e + \Delta_i}\, g^{\sum_{\mathsf{Gen} \cap E_{G_i}} \mathrm{bal}(e)}.$$

Under the discrete-log assumption on $\mathbb{G}$, the exponents of g (which is linearly independent of h in the exponent for any party that does not know $\log_g h$) must match, yielding (2). □ □

Corollary 7 is the balance-consistency counterpart of Theorem 1. Theorem 1 covers the two purely combinatorial invariants (no double spending, no creation from thin air), which hold for S_{POB} regardless of any cryptographic property of PoBValid. Global balance conservation, by contrast, is an algebraic statement about commitments and requires the soundness of zkPoB: it holds at the level of encrypted aggregates over $\mathrm{Frontier}(G_i)$ and is what makes the ledger auditable without revealing any individual balance — the product of the frontier's encrypted balances matches the product of all genesis mints, up to a blinding factor whose discrete log in h is witnessed across the accumulated zkPoB proofs.

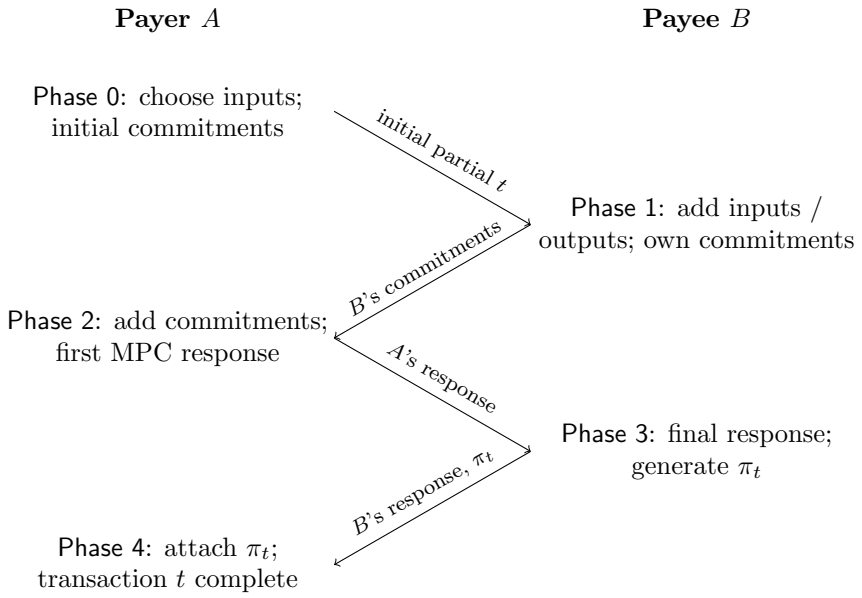

Figure 6: Transaction construction as an MPC ping-pong between payer and payee. Phase 0 is local to the payer; Phases 1–3 interleave commitments and responses to the sigma-protocol derivation of Figure 5; Phase 3 produces the non-interactive PoB proof π_t; Phase 4 is the payer's local finalisation. The submitted t is then input to Preflight (§4.2).

6.2 Transaction construction

Generating a joint transaction is a four-phase MPC between the payer and the payee (and generalises to multi-party transactions by repeating the same pattern). No part of the quorum protocol is involved during construction; the result of the MPC is a complete transaction t with its PoB proof π_t, which is then submitted to Preflight + Commit. Let A denote the payer and B the payee.

The resulting t is a single hyperedge with fixed $\mathrm{In}(t), \mathrm{Out}(t)$ and a non-interactive proof π_t, identical in shape to the abstract transaction of §2. The quorum protocol of §4 has no notion of "originator": the *client* role is simply "any party holding the finished t" (and later $\sigma(t)$). Either the payer or the payee may drive Preflight+Commit; moreover, if the initial client crashes, any peer that has received t (or $\sigma(t)$) may take over, since both phases are stateless dissemination of a signed object. Client persistence (§3) ensures that once Preflight yields $\sigma(t)$, t eventually reaches q durable copies.

7 Conclusion

S_{POB} shows that a Byzantine-tolerant payment ledger does not need SMR. A hypergraph state machine with a single extension transition, realized via a Phalanx-style two-phase protocol on a b-masking quorum system, is sufficient to provide the double-spend prevention that blockchains typically obtain from full consensus. Under the Malkhi–Reiter $n > 3b$ threshold and signing threshold $\tau = n - b$ matching bftkv, global safety is unconditional and liveness is a hypergeometric tail captured in closed form.

Comparison with SMR

The key structural distinction from classical SMR is at the storage/ordering level: the global graph G^* carries a partial order via $\prec_{G^*}$ on the causal DAG of transactions, but each individual member's local graph G_r carries *neither a global order nor the full state*. No replica is required to hold G^*, and no replica is required to agree on a linearization of causally independent transactions. Classical SMR replicates a totally ordered log in full at every replica; S_{POB} realized via Preflight+Commit does neither.

Per-transaction message complexity differs accordingly. SMR-based protocols vary sharply between common-case and worst-case execution:

- PBFT [4] is $\Theta(n^2)$ in the common case (all-to-all PREPARE and COMMIT) and $\Theta(n^3)$ during view-change, with further degradation under cascading view-changes.
- HotStuff [5] reduces the common case to $\Theta(n)$ via a star topology through the leader and threshold signatures, and keeps view-change at $\Theta(n)$. Under adversarial leader rotation or asynchrony, however, successive views must be re-run, each still incurring $\Theta(n)$.

Our protocol is $\Theta(n+q) = \Theta(n)$ per commit attempt in *every* execution: both Preflight and Commit are star-shaped around the client, there is no leader and no view-change, and a failed attempt costs another $\Theta(n)$ rather than triggering a quadratic or cubic fallback. The expected total is $4q/(1-p_t)$ from the hypergeometric analysis, which remains linear in n for any fixed success margin.

Beyond ordering and complexity, S_{POB} differs from SMR in failure behaviour: above the safety threshold, SMR loses both safety and liveness, while b-masking quorums still allow *detection* of equivocation and revocation of equivocating members [1, §5]. Concretely, a Byzantine r who contributes partial signatures $\sigma_r(t_1)$ and $\sigma_r(t_2)$ for conflicting t_1, t_2 is exposed whenever any honest member r' observes both partial signatures — for instance, when r' lies in the intersection of two signing quorums that produced $\sigma(t_1)$ and $\sigma(t_2)$. The pair $\big(\sigma_r(t_1), \sigma_r(t_2)\big)$ is a non-repudiable proof of equivocation by r, which honest members use to revoke r from U as in bftkv [1, §5].

References

[1] Ishiguro, R. *A Byzantine Fault-tolerant KV Store for Decentralized PKI and Blockchain*. `https://github.com/yahoo/bftkv/blob/master/docs/bftkv.pdf`

[2] Malkhi, D. and Reiter, M. (1998) *Byzantine Quorum Systems*.

[3] Malkhi, D. and Reiter, M. (1998) *Secure and Scalable Replication in Phalanx*.

[4] Castro, M. and Liskov, B. (1999) *Practical Byzantine Fault Tolerance*.

[5] Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., and Abraham, I. (2019) *HotStuff: BFT Consensus with Linearity and Responsiveness*. PODC '19.

[6] Dwork, C., Lynch, N., and Stockmeyer, L. (1988) *Consensus in the Presence of Partial Synchrony*. J. ACM 35(2).

[7] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., and Maxwell, G. (2018) *Bulletproofs: Short Proofs for Confidential Transactions and More*.

[8] Eagen, L., Kanjalkar, S., Ruffing, T., and Nick, J. (2022) *Bulletproofs++: Next Generation Confidential Transactions via Reciprocal Set Membership Arguments*. `https://eprint.iacr.org/2022/510`.

A Proof of Theorem 1 (Graph Safety)

By induction on the sequence $G_0, G_1, \ldots$.

Base case. $G_0 = (\emptyset, \emptyset)$ trivially satisfies properties (1)–(2): there are no edges and no inputs to fabricate.

Inductive step. Suppose G_i satisfies properties (1)–(2) of Theorem 1. Let $G_{i+1} = \mathrm{Extend}(G_i, t_i)$ with $\mathrm{Adm}(G_i, t_i)$.

No double spending. We show that every $e \in E_{G_{i+1}}$ is consumed by at most one transaction in $T_{G_{i+1}}$. Consider $e \in E_{G_{i+1}}$ and distinguish two cases.

Case 1: $e \in E_{G_i}$. The consumers of e in $T_{G_{i+1}}$ are the consumers of e in T_{G_i} (at most one, by the inductive hypothesis) plus possibly t_i if $e \in \mathrm{In}(t_i)$. Two sub-cases:

- If e is unconsumed in G_i, then t_i is the unique consumer of e in G_{i+1} (if $e \in \mathrm{In}(t_i)$) or e has no consumer (otherwise). In either sub-case the count is ≤ 1.
- If e is consumed in G_i by some $t \in T_{G_i}$, then we must show $e \notin \mathrm{In}(t_i)$. But clause 5 of Adm requires $\mathrm{In}(t_i) \subseteq \mathrm{Frontier}(G_i)$, and the frontier excludes commitments already consumed in G_i. Hence $e \notin \mathrm{In}(t_i)$ and t remains the unique consumer.

Case 2: $e \in \mathrm{Out}(t_i) \setminus E_{G_i}$. The only candidate consumer of e in $T_{G_{i+1}}$ would be t_i itself, but $e \notin \mathrm{In}(t_i)$ by clause 4 ($\mathrm{In}(t_i) \cap \mathrm{Out}(t_i) = \emptyset$). Hence e has no consumer in G_{i+1}.

No creation from thin air. For every $t \in T_{G_{i+1}}$ and every $e \in \mathrm{In}(t) \setminus \mathsf{Gen}$, clause 5 of Adm applied at the step that added t requires $e \in \mathrm{Frontier}(G_j)$ for some $j \leq i$, and by definition $\mathrm{Frontier}(G_j) \subseteq E_{G_j} \subseteq E_{G_{i+1}}$, so e is a commitment already present in the graph. By the inductive hypothesis, e was produced by some transaction in T_{G_j} (the creator exists because $\mathrm{Frontier}(G_j)$ contains only commitments that are either in Gen or have a creator in T_{G_j}). Hence no non-genesis input of any transaction is fabricated.

Therefore G_{i+1} satisfies (1)–(2) and the induction closes. □

B Proof of Theorem 3 (Global Graph Safety under the Quorum Protocol)

Let $G^* = \bigcup_{r \in H} G_r$ be the global graph of §3.2. We show $G^* \in \mathcal{G}$ — that is, the sequence of transactions accumulated in G^* corresponds to an admissible trajectory of S_{POB} starting from $G_0 = (\emptyset, \emptyset)$. We prove three sub-claims: *equivocation safety* (no two conflicting transactions can both carry a valid threshold signature, so Adm's frontier clause is never violated as G^* grows), *ancestry safety* (every non-genesis input of every $t \in T_{G^*}$ lies in $\mathrm{Frontier}(G^*)$, so clause 5 of Adm is satisfied when t is appended to G^*), and *durability* (once Commit succeeds with q acks, t is durably stored at $\geq b+1$ honest members, so every future signing quorum sees t and cannot equivocate on it). Together with Theorem 1 these imply $G^* \in \mathcal{G}$.

Equivocation safety

Say two transactions t_1, t_2 *conflict* if $\mathrm{In}(t_1) \cap \mathrm{In}(t_2) \neq \emptyset$ or $\mathrm{Out}(t_1) \cap \mathrm{Out}(t_2) \neq \emptyset$. Suppose for contradiction that both t_1 and t_2 carry valid threshold signatures.

By threshold-signature soundness, for $i \in \{1, 2\}$ there is a set $S_i \subseteq U$ of contributing signers with $|S_i| \geq \tau$. Decompose

$$S_i \;=\; H_i \sqcup B_i, \qquad H_i \subseteq H,\; B_i \subseteq B,$$

where H is the set of honest members ($|H| = n - b$) and B the set of Byzantine members ($|B| \leq b$). Byzantine members may sign both; honest members sign at most one of any conflicting pair by Algorithm 1 (the tentative mark ensures that once r signs t_1, the input overlap causes r to reject t_2). Hence $H_1 \cap H_2 = \emptyset$, so

$$|H_1| + |H_2| \;\leq\; |H| \;=\; n - b.$$

From $|S_i| \geq \tau$ and $|B_i| \leq b$, we get $|H_i| \geq \tau - b$, and therefore

$$2(\tau - b) \;\leq\; n - b, \qquad \text{i.e.,} \qquad \tau \;\leq\; \frac{n+b}{2}.$$

But by hypothesis $\tau = n - b$, so $n - b \leq (n+b)/2$, which rearranges to $n \leq 3b$, contradicting $n \geq 3b + 1$.

Ancestry safety

Let $t \in G^*$, so $t \in T_{G_r}$ for some honest r. Then r accepted t via Algorithm 2, which verifies $\sigma(t)$ via threshold signature check. By threshold soundness the signing set contains at least $\tau - b \geq (n - b) - b = n - 2b \geq 1$ honest members. Let $r' \in H$ be one. By Algorithm 1, $\mathrm{In}(t) \subseteq \mathrm{Frontier}(G_{r'})$, so every non-genesis $e \in \mathrm{In}(t)$

is produced by a unique creator$_{G_{r'}}(e) \in T_{G_{r'}}$ and not yet consumed in $G_{r'}$. Since $G_{r'} \subseteq G^*$, creator$_{G_{r'}}(e) \in T_{G^*}$, so e is also produced in G^*.

It remains to show e is unconsumed in G^*. Suppose for contradiction that some honest $r'' \in H$ holds $t' \in T_{G_{r''}}$ with $t' \neq t$ and $e \in \mathrm{In}(t')$. Then t' was committed to $G_{r''}$ via Algorithm 2, so t' carries a valid threshold signature $\sigma(t')$. But t and t' both consume e, so they conflict in the sense of the equivocation-safety argument above. Hence two conflicting transactions t, t' carry valid $\sigma(t), \sigma(t')$, contradicting equivocation safety. Therefore no honest member has recorded any consumer of e, so $e \in \mathrm{Frontier}(G^*)$.

Durability

Commit succeeds only after q acks. Among the q members responding, at most b are Byzantine, so at least $q - b = n - 2b \geq b + 1$ are honest and have durably stored t. Any future signing quorum Q'_p satisfies $|Q'_p \cap Q_c| \geq n - 2b \geq b + 1$, so at least one honest member in the intersection already holds t and will reject any transaction conflicting with t under Algorithm 1. Thus equivocation safety extends beyond t's commit to all future candidates, and every subsequent extension of G^* continues to satisfy Adm.

Combining equivocation safety, ancestry safety, and durability with Theorem 1, the sequence of transactions accumulated in G^* is an admissible trajectory of S_{POB} from G_0, so $G^* \in \mathcal{G}$. □

B.1 Tightness of $\tau = n - b$

Suppose $\tau \leq n - b - 1$. The b Byzantine members sign both t_1 and t_2. Partition the $n - b$ honest members into H_1, H_2 with $|H_1| = \lceil (n-b)/2 \rceil$, $|H_2| = \lfloor (n-b)/2 \rfloor$; present t_1 only to $H_1 \cup B$ and t_2 only to $H_2 \cup B$. Then the attacker collects at least

$$b + \left\lfloor \frac{n-b}{2} \right\rfloor$$

valid partial signatures on each of t_1, t_2. This exceeds τ whenever

$$\tau \;\leq\; b + \left\lfloor \frac{n-b}{2} \right\rfloor \;=\; \left\lfloor \frac{n+b}{2} \right\rfloor.$$

For $n = 3b + 1$, $\lfloor (n+b)/2 \rfloor = 2b$, and $\tau \leq n - b - 1 = 2b$ indeed satisfies this bound, so two conflicting threshold signatures are simultaneously achievable. Hence $\tau \geq n - b$ is necessary.

Why $n = 3b$ fails. With $n = 3b$ and $\tau = n - b = 2b$, the b-masking intersection $|Q_1 \cap Q_2| \geq n - 2b = b$ is no longer strictly greater than b, so every intersection member could be Byzantine. The equivocation-safety argument of Appendix B requires at least one *honest* intersection member to carry the tentative mark, and thus fails. The condition $n \geq 3b + 1$ is the minimum that yields $|Q_1 \cap Q_2| \geq b + 1$, guaranteeing an honest member in every pair of quorums.

C Proofs of Theorems 4 and 5 (Liveness)

Preflight Liveness

Let $F_t \subseteq U$ be the non-signer set, $u_t = |F_t|$, and $Q_p \subseteq U$ a size-q subset chosen uniformly at random. Let $Y_t = |Q_p \cap F_t|$. Since Q_p is uniform over size-q subsets and F_t is a fixed set of size u_t, Y_t is hypergeometric:

$$\Pr[Y_t = i] \;=\; \frac{\binom{u_t}{i}\binom{n-u_t}{q-i}}{\binom{n}{q}},$$

for $\max(0, q - (n - u_t)) \leq i \leq \min(u_t, q)$.

The Preflight succeeds iff at least τ members of Q_p sign, equivalently iff at most $q - \tau$ members of Q_p lie in F_t, i.e., $Y_t \leq q - \tau$. Therefore

$$\Pr[\text{pre-fail at } t] \;=\; \Pr[Y_t > q - \tau] \;=\; \sum_{i=q-\tau+1}^{\min(u_t,q)} \Pr[Y_t = i],$$

which is the claimed expression. If $u_t \le q - \tau$, then $Y_t \le u_t \le q - \tau$ almost surely, so Pr[pre-fail at t] = 0.

Commit Liveness

Let $\mu \subseteq U$ be the unreachable set with $|\mu| = m$, and Q_c a size-q subset chosen uniformly at random. Commit succeeds in one attempt iff every $r \in Q_c$ is reachable and responds with ack. Byzantine members in $Q_c \setminus \mu$ may refuse, but honest members in $Q_c \setminus \mu$ always ack by Algorithm 2 (verification of $\sigma(t)$ succeeds by assumption, and the append is unconditional).

Safety of Algorithm 2 means a Byzantine member cannot forge an ack without $\sigma(t)$; however it may silently drop its reply, contributing to apparent unreachability. Collapsing Byzantine silence into the unreachable set μ (the worst case), a single Commit attempt succeeds iff $Q_c \subseteq U \setminus \mu$:

$$\Pr[\text{commit-ok}] \;=\; \begin{cases} \binom{n-m}{q} / \binom{n}{q} & \text{if } n - m \ge q, \\ 0 & \text{if } n - m < q. \end{cases}$$

For $n = 3b + 1$, $n - q = b$, so Commit liveness is zero when more than b members are unreachable. □

D Proof of Proposition 6 (Complexity)

Preflight sends q requests to Q_p and receives q partial-signature replies on success: $2q$ messages. Commit sends q broadcasts and receives q acks (required for safety, see §4.3): $2q$ messages. Total on the success path: $M_{\text{one}}(t) = 4q$. Asymptotically $M_{\text{one}}(t) = \Theta(n + q) = \Theta(n)$ since $q \le n$. The retry analysis follows from the geometric distribution with mean $1/(1 - p_t)$ where $p_t = 1 - \Pr[\text{ok}]$. □

E Recap of bftkv

For self-containedness we reproduce the building blocks of bftkv [1] that carry over to this paper's quorum model. The read / write protocols of bftkv are not reproduced here because they are replaced by Preflight + Commit of §4.

Quorum Cliques. bftkv builds its quorum system from a trust graph (a Web of Trust). The fundamental building block is a *Quorum Clique*: a subgraph $G_C = (V, E)$ such that

1. for all $v_i, v_j \in V$, both (v_i, v_j) and (v_j, v_i) lie in E;
2. $|V| \ge 4$;
3. quorum cliques partition the trust-graph vertices: each vertex belongs to at most one quorum clique.

Under these conditions a Sybil attack tends to split a clique rather than silently capture it: if a node is compromised and the attacker adds colluding peers, the compromised node cannot simultaneously belong to a second clique, so the attack severs trust links and yields two separated cliques instead of a silent takeover.

Quorum Certificates. A *quorum certificate* is a collective signature produced by the members of a quorum clique. When a node joins the network, its quorum issues a certificate that represents the trust graph rooted at that node; this certificate authenticates the node to the rest of the system without relying on a centralised PKI.

KV Quorum. Once a tuple is signed by a quorum clique, it is stored on nodes outside the clique. These storage nodes form a second quorum system, the *KV quorum*, which also serves as an auditor: each node checks for equivocation before accepting a tuple and, on detecting malicious behaviour, revokes the offender and propagates a proof of the revocation. A revoked member has no path back into the network.

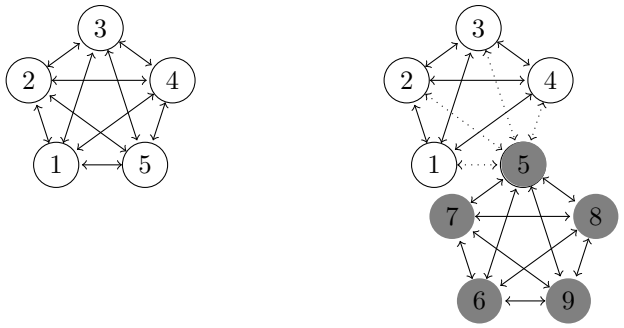

Figure 7: A legitimate quorum clique (left). If node 5 is compromised and the attacker introduces colluding nodes (6–9) to outnumber the legitimate members, node 5 cannot simultaneously belong to another clique: property (3) above forces the attacker to sever trust links. The result is two separated cliques (right) rather than a silent takeover of the original clique.

Revocation and equivocation detection. Equivocation is the core misbehaviour a BFT quorum must detect. Two kinds arise. The first is variable-level equivocation: a malicious signer produces $\langle x, t, v\rangle$ and $\langle x, t, v'\rangle$ with $v \neq v'$, which is caught at the KV quorum because any two stored tuples for the same $\langle x, t\rangle$ can be compared. The second is graph-level equivocation: the malicious node presents inconsistent views of the trust graph to different peers. Every node therefore keeps its own view of the trust graph and, on receiving a quorum certificate, checks that the certificate is consistent with its local view, updating the view with any new information the certificate carries.

Security analysis of the quorum bound. We sketch the standard split-clique argument of [1] that gives $n > 3b$. Consider an attacker who wants $\langle x, t, v\rangle$ and $\langle x, t, v'\rangle$ (with $v \neq v'$) to both pass the signing threshold. The worst case is a clique split into two honest groups $\mathcal{H}_1, \mathcal{H}_2$ of roughly equal size — the attacker presents v to $\mathcal{H}_1$ and v' to $\mathcal{H}_2$, then combines their partial signatures with the b Byzantine members. The maximum number of signatures obtainable on either value is

$$b \;+\; \frac{n-b}{2},$$

the b Byzantine members plus half of the honest population. For the threshold $\tau = n - b$ to remain unachievable by the attacker we need

$$n - b \;>\; b + \frac{n-b}{2} \quad\Longleftrightarrow\quad n \;>\; 3b,$$

which is the Malkhi–Reiter b-masking bound used throughout this paper. The algebraic version of this argument, specialised to transactions and threshold-signed aggregation, appears in Appendix B (equivocation safety) and its tightness is proved in Appendix B.1.

In this paper, Byzantine equivocation is prevented *structurally* by the threshold-signature mechanism of §5: no two conflicting transactions can both obtain a valid $\sigma(t)$ once the $\tau = n - b$ threshold is in force, so variable-level equivocation on any $e \in \mathcal{E}$ is ruled out at the signing stage. The trust-graph construction above is still used to define the membership set U and to revoke provably Byzantine members.