Strongly Adaptive Asynchronous Byzantine Agreement with Linear Communication and Almost-Everywhere Termination

Renas Bacho[1], Simon Holmgaard Kamp[1], Julian Loss[1], Jesper Buus Nielsen[2], and Maciej Obremski[3]

[1] Ruhr University Bochum, Germany
[2] Aarhus University, Denmark
[3] National University of Singapore, Singapore

Abstract. It is well known that Byzantine agreement requires $\Omega(t^2)$ messages against a strongly adaptive adversary that may corrupt $t = \Theta(n)$ parties and suppress a party's undelivered messages upon corruption. We circumvent this lower bound by relaxing termination: for any constant $0 < \nu < 1$, at least a $(1 - \nu)$ fraction of the honest parties terminate, while agreement and validity hold for every honest decision. For every constant corruption gap $\delta > 0$ and every $t \leq (1-\delta)n/3$, we construct binary agreement with this guarantee in authenticated asynchronous networks against strongly adaptive corruptions with message takebacks. Except with negligible probability, these guarantees hold and every honest decision carries a transferable justifier. The protocol achieves almost-everywhere termination in expected $O(1)$ rounds, expected total communication $O(n\kappa^2)$ bits, and expected maximum communication $\widetilde{O}(\kappa^2)$ bits over honest parties before corruption, where κ is the computational security parameter.

Our construction combines a recursively shared BLS signing key with certificate dissemination over a sparse graph. The recursive sharing is used to implement threshold signing with distinct safety and liveness thresholds. Dissemination propagates existing certificates to almost all honest parties. The construction uses a trusted setup and is secure in the random oracle model under the algebraic one-more computational Diffie–Hellman assumption.

We further obtain full binary agreement with the same asymptotic communication and round complexity against adaptive corruptions without message takebacks. The extension requires neither atomic multi-send nor private communication channels.

1 Introduction

Byzantine agreement [37] allows n mutually distrustful parties to decide a common value despite arbitrary behavior by corrupted parties. In an asynchronous network, randomized protocols can tolerate $t < n/3$ corruptions. A strongly adaptive adversary may corrupt a party after observing that it sent a message and suppress the message before delivery. This capability, also called after-the-fact message removal or a message takeback, lets the adversary erase messages from, for instance, small committees on which progress depends. Dolev and Reischuk [21] establish an $\Omega(n + t^2)$-message worst-case lower bound for deterministic Byzantine agreement, which is quadratic in n when $t = \Theta(n)$. Abraham et al. [1] establish an expected $\Omega(t^2)$-message lower bound for randomized Byzantine agreement against strongly adaptive corruptions with message takebacks when every honest party must terminate. Subquadratic protocols must therefore change the model or the required guarantees, for example by restricting adaptivity, hiding the communication pattern from the adversary, or relaxing termination.

We relax termination while retaining agreement and validity for every honest party that decides. Specifically, *almost-everywhere termination* guarantees that, for any prescribed constant $0 < \nu < 1$, at least a $(1-\nu)$ fraction of honest parties terminate, except with negligible probability. The remaining honest parties may remain undecided. Every honest decision carries a transferable certificate, called a *justifier*, that any party can verify.[4] Classical almost-everywhere agreement, introduced by Dwork, Peleg, Pippenger, and Upfal [23], guarantees agreement and validity only for a large subset of honest parties and allows the remaining honest parties to decide inconsistently. Subsequent work develops this notion for scalable agreement and secure computation [35,18,6].

[4] For instance in a step where an honest party needs to see $t + 1$ pre-votes for 0 to vote 0, a justifier could be $t + 1$ signed pre-votes on 0 from distinct parties. If the signature scheme is threshold, this justifier can be made compact. Justified outputs are used to justify inputs when protocols are composed.

Boyle, Goldwasser, and Tessaro [13] already use *almost-everywhere certified agreement* followed by dissemination to obtain full termination in synchronous networks with static corruptions. Their core guarantee lets almost all honest parties learn the value without allowing any honest party to accept an incorrect value. Our core protocol provides this guarantee with asynchronous communication and strongly adaptive corruptions with message takebacks; extending it to full termination requires excluding takebacks.

Technical contribution. Our threshold signing construction starts from a BLS key shared recursively down a constant-depth tree. Signatures are reconstructed upward using sparse forwarding, with distinct safety and liveness thresholds. The public sampling guarantees hold simultaneously for all relevant party sets, allowing corruptions and message takebacks to be chosen adaptively after setup. We combine this construction with sparse certificate dissemination to implement the common coin [15], input validation [14], and justified proxcensus [30] used in the agreement reduction of [24]. The composition explicitly accommodates missing honest inputs and outputs: we choose the signing and dissemination thresholds so that every subsequent step retains enough honest contributors to make progress.

1.1 Our Results

Let κ be the computational security parameter, set the statistical security parameter $\lambda = \Theta(\kappa)$, and let $t \leq (1-\delta)n/3$ for a constant $\delta > 0$. In authenticated asynchronous networks, we construct binary agreement against a strongly adaptive adversary that may suppress undelivered messages sent before corruption. Except with negligible probability, all honest decisions agree and satisfy validity, and at most an arbitrarily small constant fraction of honest parties remain undecided. Almost-everywhere termination takes an expected constant number of asynchronous rounds. The expected total communication is $O(n\kappa^2)$ bits, and the expected maximum outgoing communication over all parties, counting each party only before its corruption, is $\widetilde{O}(\kappa^2)$ bits. Theorem 3 gives the full statement.

The construction uses a trusted setup that fixes public sampling structures and privately distributes shares of recursively shared BLS signing keys. Security is proved in the random oracle model under the algebraic one-more computational Diffie–Hellman assumption [4]. Corollary 4 extends this result to full binary agreement against adaptive corruptions without message takebacks. This model does not grant an atomic multi-send operation: the adversary may corrupt a party between any two point-to-point sends, but it cannot suppress a send that completed before corruption. The compiler from almost-everywhere to full termination preserves expected constant asynchronous round complexity and expected $O(n\kappa^2)$ communication in the same parameter regime. It uses neither private online channels nor a fixed dissemination committee.

Table 1 compares our guarantees and complexity bounds with prior asynchronous constructions. We discuss related work in more detail in Appendix A.

2 Technical Overview

The BA construction follows the reduction in [24] and has three components: a common coin, initial input validation, and justified proxcensus (a variant of graded agreement). Input validation first supplies *justifiers* proving that a bit was proposed by an honest party. Parties then alternate justified proxcensus and a common coin until a decision is justified. The common coin [15] and input justifiers [14] are implemented using threshold signatures with appropriate thresholds. Justified proxcensus [30] (full version [29]) relies on a doubling step which takes at most two justified inputs and either establishes a uniqueness certificate for one of them, or distributes the justifiers for both of them. This is instantiated by a combination of threshold signing and dissemination which we call sign-or-disseminate.

2.1 Byzantine Agreement Components

Threshold signing. Our threshold signing protocol has a safety threshold q and a liveness threshold h. A party contributes a signature by releasing its own leaf partial signatures for the instance. Producing a threshold signature requires at least $q - t$ honest parties to contribute a signature, while at least h honest contributors to the signing instance for the same message guarantee that *almost all* honest parties output its threshold signature. This output guarantee permits a small constant fraction of honest parties to receive

Table 1. Comparison of asynchronous agreement protocols for $t < (1-\delta)n/3$ and short values. The Primitive column identifies the stated primitive: binary agreement (BA), total order broadcast (TOB), or validated asynchronous Byzantine agreement (VABA); every row yields an asynchronous Byzantine agreement protocol. The BKLL20 communication bound includes setup refresh for unbounded reuse, but the cited analysis gives no round bound for the refresh; the linear DKLN25 communication bound is amortized over sufficiently large batches. All displayed bounds are in expectation. We use κ and λ for the computational and statistical security parameters, respectively. $\mathrm{BA}_{\mathrm{AET}}$ denotes BA with almost-everywhere termination; AMS, SE, and NT denote atomic multi-send, secure erasures, and no message takebacks, respectively, while Unrestricted denotes strong adaptivity with message takebacks.

		Communication			
Work	Primitive	Total	Maximum outgoing per party	Rounds	Adaptivity
BKLL20 [7]	BA	$O(n\,\mathrm{poly}(\kappa))$	$O(n\,\mathrm{poly}(\kappa))$	—	AMS, SE, NT
DKLN25 [20]	TOB	$O(n\,\mathrm{poly}(\kappa))$	$O(n\,\mathrm{poly}(\kappa))$	$O(1)$	AMS, SE, NT
ACGSY25 [2]	VABA	$O(n\,\mathrm{poly}(\kappa))$	$O(n\,\mathrm{poly}(\kappa))$	$O(1)$	AMS, SE, NT
FMT26 [25]	BA	$O(n^{3/2}(\lambda+\log n)\kappa)$	$O(\sqrt{n}(\lambda+\log n)\kappa)$	$O(\lambda)$	NT
This work	BA	$O(n\kappa^2)$	$\widetilde{O}(\kappa^2)$	$O(1)$	NT
This work	$\mathrm{BA}_{\mathrm{AET}}$	$O(n\kappa^2)$	$\widetilde{O}(\kappa^2)$	$O(1)$	Unrestricted

no signature, even when the liveness threshold is met. When the signature is needed to justify an input to the next step, those parties may therefore be unable to contribute input to that step. We must consequently allow for missing honest inputs as well as missing honest outputs when composing the three components. Our implementation also requires $h-q$ to be a constant fraction of n. For each component, we choose the thresholds so that the available honest inputs suffice for liveness despite this margin, and the guaranteed outputs leave enough parties able to continue.

Coin flipping. The simplest case is a common coin: all honest contributors to the coin instance contribute a signature for the same fixed message. Hashing the unique signature gives the coin bit, and the signature serves as a transferable justifier for it. A safety threshold $q > t$ keeps the coin hidden until an honest party contributes a signature. Since almost all honest parties contribute input when a coin is needed for progress, the available support is close to $n-t$, leaving room for both the signing protocol's liveness margin and missing input contributions. This step alone requires only an honest majority with a margin proportional to n: $n-2t$ must leave a constant fraction of n between adversarial support and available honest support.

Initial input validation. Each party may receive an unjustified binary input, and almost-everywhere termination is guaranteed when all but ϵn honest parties receive inputs, for a sufficiently small constant $\epsilon > 0$. Input validation obtains a certificate showing that a bit was proposed by an honest party. Each party with an input contributes a signature only for its own bit, using a safety threshold $q = t+1$. Honest support may now be split between two values, so only the more common bit is guaranteed support from at least half of the honest parties with an input. Ignoring missing inputs for the moment, the available support is therefore at least $(n-t)/2$, which must exceed t by a constant fraction of n, as required for liveness. This requires $n-3t$ to be a constant fraction of n, which also leaves room for missing inputs. Either bit may be justified by an input validation certificate, and every bit justified in this way was proposed by an honest party. This step is run only once, before the justified agreement iterations begin.

Justified proxcensus. Justified G-proxcensus takes justified binary inputs and produces justified grades in $\{0,\ldots,G-1\}$, each accompanied by a publicly verifiable justifier. Justified graded agreement requires any two justified output grades to differ by at most one. Justified validity requires that, if b is the only justified input bit, then every justified output grade equals $b(G-1)$. Our almost-everywhere variant guarantees outputs to almost all honest parties whenever almost all honest parties have justified inputs.

In the recursive construction, each step turns an s-grade protocol into a $(2s-1)$-grade protocol by using the preceding outputs as candidate grades. By justified graded agreement, there are at most two candidates, and they are adjacent. A candidate z can be promoted to the even grade $2z$ by a uniqueness certificate,

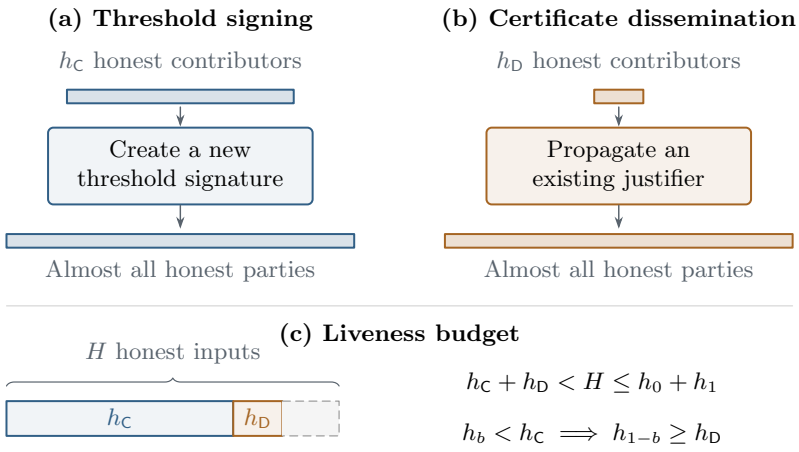


Fig. 1. Sign-or-disseminate. The support bar shows the two thresholds side by side: together they take up less than the full width, and the gap that remains (dashed) is positive in our instantiation once n is large enough. That gap is what makes the two cases exhaustive: if one value falls below the signing threshold, the other meets the dissemination threshold, and if both fall below it, both meet the dissemination threshold. The upper bars show the liveness thresholds and the lower bars the guaranteed output sets, all drawn to a common schematic scale. The signing threshold is large, while a small number of honest contributors is enough for dissemination, and both primitives reach almost all honest parties. Lemma 4 gives the exact condition, including rounding, and the bounds on missing outputs.

with the requirement that certificates promoting distinct candidates cannot both exist. Alternatively, the justifiers for two adjacent candidates $z-1$ and z together justify the odd grade $2z-1$. Thus at most one even output grade can be justified, and any justified odd output grade is adjacent to it, preserving justified graded agreement. If only one candidate is justified, only its doubled grade can be justified as an output, preserving justified validity.

We implement the recursive step using binary sign-or-disseminate. Each party contributes its own candidate grade and the justifier obtained from the preceding step as input to threshold signing and certificate dissemination. A threshold signature serves as the uniqueness certificate. We choose the signing safety threshold $q > (n+t)/2$, so any two sets of q contributors intersect in more than t parties and therefore contain a common honest party. Since each honest party contributes a signature for only one candidate, conflicting threshold certificates cannot exist, as formalized in Corollary 2.

For termination, we choose the thresholds so that the number of honest parties with inputs is at least the sum of the signing and dissemination thresholds. If either candidate has too little honest support for dissemination, the other therefore has enough support for threshold signing. Otherwise, both candidates have enough support for dissemination. Thus almost all honest parties obtain either a uniqueness certificate for one candidate or justifiers for both candidates. The difference $(n-t)-(n+t)/2 = (n-3t)/2$ leaves room for missing inputs, support for the less common candidate, and the signing protocol's liveness margin. Section 5.2 gives the parameter constraints for this step and initial input validation. Figure 1 illustrates how the two thresholds ensure progress.

Three recursive steps produce the nine grades used by the binary agreement protocol, with almost-everywhere termination at each step. The initial input validation certificates ensure that validity ultimately refers to an input of an honest party. Extreme grades justify a decision, grades adjacent to an extreme preserve the corresponding bit, and middle grades adopt the common coin. A party that obtains a decision certificate contributes it to certificate dissemination and terminates; every recipient forwards its first decision certificate in the same way. The binary sign-or-disseminate argument ensures progress: either enough parties disseminate the decision, or enough hold the adjacent grade to support the decided bit in the next iteration. Extreme grades do not justify inputs to later iterations. The coin makes all justified inputs for the next iteration coincide with probability at least $1/2$, while the liveness analysis preserves enough justified inputs despite missing outputs. This yields almost-everywhere termination in expected constant asynchronous rounds, as proved in Section 6.

2.2 Implementing the Components

Implementing threshold signing. The same recursive threshold signing construction implements the thresholds needed by all three components. It combines standard BLS signatures [9], threshold sharing [8], and a recursively sampled access structure. The root signing key is shared down a constant-depth tree of random committees. At the leaves, parties produce verifiable BLS partial signatures only for their own justified inputs. Committees reconstruct signatures locally and forward them through sparse random edges toward the root, where reconstruction yields a unique transferable threshold certificate. Every transmitted partial or reconstructed signature carries an input justifier, allowing recipients to verify it and help reconstruct and forward even if their own input differs or is absent. The children of each committee are overlapping random subsets of that committee, which provides many independently representative samples while increasing the total committee membership by only a constant replication factor at each level.

For unforgeability, induction over the levels shows that the shares held by any set of fewer than q parties are insufficient to reconstruct the root signing key. For liveness, induction shows that any set of at least h honest contributors supplies enough partial signatures for reconstruction and forwarding to reach the root. Both properties hold simultaneously for every relevant party, sender, and recipient set, so one final union bound gives a public setup that remains valid even when the adversary chooses its corruptions and message takebacks adaptively after seeing the setup. The minimum committee size makes the probability of any forwarding failure small enough for this union bound even when each forwarding list has length proportional to the tree's branching factor, yielding the communication bound $O\big(\big[n\lambda + n(n/\lambda)^{1/L}\big]\,\kappa\big)$ for constant recursion depth L. Section 4 gives the construction and analysis.

Implementing certificate dissemination. Certificate dissemination uses a public disperser sampled once during setup. Each party has a short public list of randomly sampled recipients and sends its justifier to those recipients. A union bound shows that every sufficiently large source set reaches all but the prescribed fraction of parties, simultaneously for all possible source sets. After the required honest parties contribute input, dissemination takes at most one asynchronous time unit and uses $O((n+\lambda)(B+\kappa+\log n))$ bits when certificates have B bits. Subsection 5.1 gives the construction and analysis.

2.3 From Almost-Everywhere to Full Termination

Following Boyle, Goldwasser, and Tessaro [13], parties holding the certified output disseminate it to the remaining parties. Against adaptive corruptions without message takebacks, almost-everywhere termination can be upgraded to full termination without a dissemination committee or atomic multi-send. Every party that obtains the justified decision samples $O(\lambda+\log n)$ recipients using fresh private randomness and sends to them sequentially. Because each next recipient remains unknown until its point-to-point send is irrevocable, the adversary cannot preemptively corrupt all useful senders toward a particular recipient. A final union bound shows that every honest party receives the unique justified decision. We tolerate missing inputs throughout the core protocol and use dissemination to all parties only after a decision, so the core retains almost-everywhere termination even with message takebacks. The additional communication is $O(n\kappa^2)$ in the stated parameter regime, with at most one additional asynchronous time unit in the model without message takebacks.

Paper Organization. Section 3 specifies the model, notation, setup assumptions, and protocol conventions. Section 4 presents recursive threshold signing. Section 5 constructs certificate dissemination, establishes binary sign-or-disseminate, and develops the input validation, justified almost-everywhere proxcensus, and almost-everywhere common coin components. Section 6 combines these components into an asynchronous binary agreement protocol with almost-everywhere termination and analyzes its security and complexity. Subsection 6.2 shows how to obtain full termination when message takebacks are disallowed.

3 Preliminaries and Model

Notation. For a positive integer k, write $[k] := \{1, \ldots, k\}$, and identify each party P_i with its index $i \in [n]$ when writing sets of parties. The computational and statistical security parameters are κ and λ, respectively. We assume that n is polynomially bounded in κ; thus, after increasing the exponent if necessary, fix a positive

integer constant $c \geq 1$ such that $n \leq \kappa^c$ for all sufficiently large κ. Whenever we state the simplified $O(n\kappa^2)$ total and $\widetilde{O}(\kappa^2)$ maximum per-party bounds, the fixed integer recursion depth is chosen with $L \geq c$. We write $\mathrm{negl}(\kappa)$ for a function smaller than the reciprocal of every positive polynomial for all sufficiently large κ. Statistical sampling failures are bounded explicitly in terms of λ, and computational failure probabilities are negligible in κ. We use log for logarithms to base 2 and ln for natural logarithms. The notation $\widetilde{O}$ suppresses only factors polynomial in logarithms of the parameters; full powers of κ and λ remain explicit.

Network and corruptions. The n parties $P_1, \ldots, P_n$ communicate over authenticated asynchronous point-to-point channels. A probabilistic polynomial-time adversary controls message scheduling and permanently corrupts at most t parties during an execution. A party is *honest* if it is never corrupted in the execution, equivalently, if it remains uncorrupted at the end of the execution. Unless explicitly qualified, honesty means remaining uncorrupted throughout the execution. When a party may be corrupted later, we instead say that it is *uncorrupted at a specified instant* or refer to its actions *before corruption*. Messages between honest parties are eventually delivered, but there is no bound on their elapsed delivery time. With message takebacks, corruption permits the adversary to suppress the corrupted sender's undelivered messages; without takebacks, a send completed before corruption remains irrevocable; if its recipient remains honest, the message is eventually delivered to that recipient. Each point-to-point send is one atomic action, and a party can be corrupted between any two such actions. In the compiler for full agreement, sampling a fresh uniform recipient from private local randomness and issuing the corresponding point-to-point send occur within one such atomic action. The adversary learns the recipient only when the send has become irrevocable. After that point, the message and its endpoints are visible; no privacy of the online channels is assumed.

Setup. A trusted dealer privately distributes setup shares and erases its secret working state before the online execution, without requiring erasures by the protocol parties. The setup is independent of the inputs and of the subsequent protocol instances.

Justifiers and protocol instances. A protocol that accepts justified inputs specifies an input justification predicate J_{IN}, which may be supplied as a parameter. A protocol that produces justified outputs defines an output justification predicate J_{OUT}. An input or output is accompanied by a justifier for its value under the corresponding predicate. Justifiers are *transferable*: any party can verify the same value and justifier using the public parameters. We use $\top$ and $\bot$ for predicate acceptance and rejection, respectively.

Definition 1 (Justification predicate). *A* justification predicate *is an efficiently computable public predicate $\mathcal{J} : \{0,1\}^* \times \{0,1\}^* \to \{\top, \bot\}$ whose evaluation depends only on its arguments and the public parameters, including the protocol instance identifier, and not on the verifier's local protocol state. A string π is a* justifier *for a value m with respect to $\mathcal{J}$ if $\mathcal{J}(m, \pi) = \top$.*

Following [24, Definitions 2 and 3], a value m is *$\mathcal{J}$-justified* with respect to an execution if, at some point in that execution, an honest party has m and a justifier π satisfying $\mathcal{J}(m, \pi) = \top$ in its view. Statements about $\mathcal{J}$-justified values refer to the execution under consideration, which we leave implicit. For a $\mathcal{J}$-justified value m, an associated justifier is a string π that occurs together with m in an honest party's view and satisfies $\mathcal{J}(m, \pi) = \top$. Security properties of $\mathcal{J}$-justified values hold except with their stated failure probabilities.

All protocol instances, messages, and justifiers are domain-separated under composition; we leave their identifiers implicit except where needed in cryptographic expressions. Each party receives at most one input and produces at most one output per instance, and each protocol rule is executed at most once per indexing tuple unless repetition is explicit. For a protocol instance Π, we write $\Pi.\mathsf{contribute}(m, \pi)$ for a party contributing its own input value m and justifier π to that instance. All parties execute the prescribed rules of every instance, whether or not they contribute input to it. An honest party contributes to signing only for its own input; after verifying a justifier for another party's input, it may reconstruct and forward for that value without acquiring an input itself. Outputting from a component does not stop prescribed reconstruction and forwarding; the final BA decision rule specifies when a party terminates the session.

Communication complexity. We count only online messages prescribed by the protocol while their sender is uncorrupted. For each party P_i, let C_i be the total number of bits it sends before corruption, or throughout the execution if it is never corrupted. Total communication is $\sum_i C_i$, and maximum communication is $\max_i C_i$. Communication used to generate and distribute the trusted setup is not included in these bounds.

Asynchronous time. Following Canetti and Rabin [16], we measure asynchronous time (or rounds) in units of the actual maximum message delay in a finite execution. With takebacks, the maximum is over messages between honest parties; without takebacks, it includes all messages sent before the sender's corruption to an honest recipient. Local computation and any finite sequence of prescribed point-to-point send actions requiring no intervening message delivery consume no asynchronous time in this measure. Thus, a party completes such sends as part of the event in which it contributes input or receives the message that enables them, although the adversary may still corrupt the party between two sends.

Definition 2 (BA with almost-everywhere termination). *Fix constants $0 \le \epsilon, \nu < 1$. Each party P_i may receive a binary input $b_i \in \{0,1\}$, and the protocol defines an output justification predicate $J_{\text{OUT}}^{\mathsf{BA}}$ for its certified decisions. Let $\mathcal{H} \subseteq [n]$ be the set of parties that remain uncorrupted throughout the execution. The protocol satisfies the following properties.*

1. ***Justified agreement.*** *Every honest decision is $J_{\text{OUT}}^{\mathsf{BA}}$-justified and carries a transferable justifier, and any two $J_{\text{OUT}}^{\mathsf{BA}}$-justified decision bits are equal.*
2. ***Honest-input validity.*** *Every honestly decided bit is the input of a party in $\mathcal{H}$. In particular, if every input held by a party in $\mathcal{H}$ equals b, then every honest party that decides outputs b.*
3. ***Almost-everywhere termination.*** *If all but ϵn parties in $\mathcal{H}$ receive an input, then at least $(1-\nu)|\mathcal{H}|$ parties in $\mathcal{H}$ eventually decide.*

Definition 3 (Binary agreement). *A protocol satisfies binary agreement if it satisfies Definition 2 with $\epsilon = 0$ and $\nu = 0$. Thus, whenever every honest party receives an input, every honest party eventually decides.*

We reduce binary agreement to a justified variant in which inputs carry justifiers. Input validation (Section 5.3) supplies an input justification predicate under which every justified bit was the input of an honest party, so justified validity then implies honest-input validity.

Definition 4 (Justified BA with almost-everywhere termination). *Fix constants $0 \le \epsilon, \nu < 1$ and an input justification predicate J_{IN}. Each party P_i may receive a J_{IN}-justified binary input, and the protocol defines an output justification predicate $J_{\text{OUT}}^{\mathsf{BA}}$ for its certified decisions. The protocol satisfies justified agreement as in Definition 2 and the following properties.*

1. ***Justified validity.*** *If b is the only J_{IN}-justified bit, then every $J_{\text{OUT}}^{\mathsf{BA}}$-justified decision bit equals b.*
2. ***Almost-everywhere termination.*** *If all but ϵn parties in $\mathcal{H}$ receive a J_{IN}-justified input, then at least $(1-\nu)|\mathcal{H}|$ parties in $\mathcal{H}$ eventually decide.*

4 Recursive Threshold Signing

Threshold signing creates a signature certifying contributions from a prescribed number of parties. We call a threshold signature used as a justifier a *threshold certificate*. Each instance is associated with an input justification predicate J_{IN} and value m and accepts an input (m, π) whenever $J_{\text{IN}}(m, \pi) = \top$. A party *contributes a signature* for an instance by releasing its own leaf partial signatures for that instance. For a threshold signing instance Π, the operation $\Pi.\mathsf{contribute}(m, \pi)$ prescribes this contribution for m with justifier π. Parties without an input still participate in the prescribed reconstruction and forwarding after verifying a justifier π for the value m with $J_{\text{IN}}(m, \pi) = \top$. We require the following properties against strongly adaptive corruptions with message takebacks, except with negligible probability.

Definition 5 (Threshold signing). *Let $q \in \{t+1, \ldots, n\}$ be an integer threshold, and let $0 < a, \eta < 1$ be constants with $q + \lceil an \rceil \le n - t$. A protocol is a (q, a, η)-threshold signing protocol with input justification predicate J_{IN} if it satisfies the following properties. Its output signatures are publicly verifiable under the verification key fixed by setup; q is the safety threshold and $q + \lceil an \rceil$ is the liveness threshold.*

1. ***Liveness.*** *If at least $q + \lceil an \rceil$ honest parties contribute input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$ to the instance for m, then all but $\lfloor \eta n \rfloor$ honest parties output a valid transferable q-threshold signature σ for the step identifier and m.*

2. ***Unforgeability.*** *A valid q-threshold signature for the step identifier* sid *and m can be produced only if at least $q - t$ honest parties have contributed a signature for (sid, m). An honest party contributes a signature only for its own input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$.*
3. ***Uniqueness.*** *Any two honest parties that output a valid signature in the same instance output the same signature.*

We construct a threshold signing protocol from a single BLS key shared through a recursively sampled hierarchy of committees, which we call *bins*. The hierarchy serves two purposes. First, it defines a monotone access structure in which the shares of fewer than q parties are insufficient to produce the unique BLS signature on the message, where q is the safety threshold. Second, it routes reconstructed bin signatures toward the root with subquadratic communication while ensuring reconstruction from the liveness threshold of honest contributors. A signature reconstructed for a non-root bin is an intermediate partial signature toward the root signature, while the root signature is the transferable threshold certificate.

Proofs for this section are in Appendix C.

4.1 Parameters and the Committee Hierarchy

Safety and liveness thresholds. Fix an integer safety threshold q and constants $0 < a, \eta < 1$. The parameter a determines the gap between the liveness and safety thresholds, while ηn bounds the number of honest parties that may receive no output. Define the global liveness threshold $h := q + \lceil an \rceil$ and assume $t < q < h \leq n - t$.

Density margins. Define the normalized thresholds $s := \frac{q-1}{n}$ and $\ell := \frac{h}{n}$ and the gap $g := \ell - s$. The gap satisfies $g \geq a$. Set $\phi := \frac{s+\ell}{2}$ and $\alpha := \frac{g}{16}$. Thus, we have $s < \phi - 4\alpha < \phi + 4\alpha < \ell$.

Recursion depth and level-dependent densities. Let $L \geq 1$ be a fixed integer recursion depth. The following lower and upper density sequences distribute the gap between the safety and liveness thresholds across these levels:

$$d_j^- := s + \frac{j}{L}(\phi - 4\alpha - s), \qquad j = 0, \ldots, L, \tag{1}$$

$$d_j^+ := \ell - \frac{j}{L}(\ell - \phi - 4\alpha), \qquad j = 0, \ldots, L. \tag{2}$$

Consequently, $d_0^- = s$, $d_L^- = \phi - 4\alpha$, $d_0^+ = \ell$, $d_L^+ = \phi + 4\alpha$. Both consecutive density gaps equal $\zeta := \frac{g}{4L}$. For the forwarding analysis, set

$$\theta := (1 - \eta)(\phi + 4\alpha) \qquad \text{and} \qquad \beta := \alpha\eta(\phi + 4\alpha)/2.$$

Replication and leaf size. Set $\bar{\lambda} := \lambda + \log n$ and choose

$$\rho = \Omega\left(\frac{1}{\zeta^2 \min\{\phi, 1 - \phi - 2\alpha\}}\right), \qquad m_* := c_* \frac{\bar{\lambda}}{\zeta^2},$$

where c_* is sufficiently large as a function of the fixed normalized thresholds s, ℓ, the constants η, L, and the preceding choice of ρ.

If $m_* \geq n$, use direct q-out-of-n Shamir sharing and let $\mathcal{G}$ be the complete directed sender–recipient graph on $[n] \times [n]$. Each contributor sends its partial signature and input justifier to every party, which reconstructs after receiving q valid shares. Under $t < q < h \leq n - t$, this satisfies Definition 5 with safety threshold q, liveness threshold h, no missing honest outputs, and one asynchronous time unit after the required input contributions. For justifiers of length at most B, its communication is $O(n^2(\kappa + B)) \leq O(nm_*(\kappa + B))$. Henceforth, the recursive hierarchy assumes $m_* < n$. Set

$$b := \left\lceil \rho \left(n/m_*\right)^{1/L} \right\rceil, \qquad p := \frac{\rho}{b}, \qquad f := \lceil c_f b \rceil,$$

where the forwarding constant c_f is sufficiently large.

The replication parameter ρ is the expected number of children of a bin that contain any fixed member of its parent committee. The choice of b makes the nominal leaf committee size $\Theta(m_*)$ and produces the $n^{1/L}$ factor in the forwarding cost.

Recursive hierarchy. Let T be a rooted b-ary tree of depth L, let its root committee be $Q_r = [n]$, and sample the remaining committees top-down. For every child u of an internal bin v, include each member of Q_v in Q_u independently with probability p.

Conditioned on Q_v, every child size is distributed as $\mathrm{Bin}(|Q_v|, p)$, and different children may overlap. During setup, committees and Shamir key shares are generated downward from the root. During the online protocol, reconstructed bin signatures travel in the opposite direction from the leaves toward the root.

Forwarding maps. For every edge $u \to v$ toward the root and every $P_i \in Q_u$, independently sample an ordered forwarding tuple $R_{u\to v}(i)$ of f recipients, each chosen independently and uniformly from Q_v with replacement. Duplicate recipients in a tuple are suppressed when messages are sent.

Every party that reconstructs the signature of a child bin sends the same signature to the recipients in its tuple. The fanout $f = \Theta(b)$ compensates for the ratio between the sizes of a child committee and its parent, and the forwarding analysis holds simultaneously for every admissible set of honest senders and recipients. Both recursive forwarding and certificate dissemination use disperser properties. We analyze them separately because forwarding goes from a child committee Q_u to its parent committee Q_v, whose cardinalities generally differ, whereas certificate dissemination uses all n parties as both senders and recipients.

Reconstruction thresholds. Define the reconstruction threshold of every bin by

$$k_v := \begin{cases} \lfloor \phi b \rfloor + 1, & v \text{ is internal,} \\ \lfloor \phi |Q_v| \rfloor + 1, & v \text{ is a leaf.} \end{cases}$$

Thus, a leaf reconstructs from partial signatures held by k_v parties, while an internal bin reconstructs from k_v child bin signatures.

Strongly adaptive corruptions. The committee hierarchy and forwarding maps are public, and the adversary may corrupt a sender after observing its messages and suppress any messages that have not yet been delivered. The liveness analysis therefore fixes the final corruption set and conservatively ignores every message sent by a party in that set. The sampling and forwarding properties hold simultaneously for every admissible final corruption set, so the analysis remains valid when the corruptions are chosen adaptively after the hierarchy is known.

The public recursive signing graph is $\mathcal{G} := (T, \{Q_v\}_{v\in T}, \{R_{u\to v}\}_{u\to v})$. Figure 2 illustrates the downward sharing hierarchy and the upward flow of reconstructed bin signatures.

4.2 Open and Live Bins

An *open* bin is one whose secret exponent can be reconstructed from the scalar shares held by a specified party set. A *live* bin has sufficient density of honest contributors and sufficiently many live children to propagate a reconstructed signature toward the root.

For a set of parties S, a leaf is S-open if it contains at least k_v members of S, and an internal bin is called S-open if at least k_v of its children are S-open. For a set A of honest contributors, a leaf v at level L is called A-live if $|A \cap Q_v| \geq d_L^+ |Q_v|$. An internal bin v at level $j < L$ is A-live if $|A \cap Q_v| \geq d_j^+ |Q_v|$ and at least a $(\phi + 2\alpha)$-fraction of its children are A-live. For an A-live internal bin v, let $\mathcal{L}_v$ denote its set of A-live children.

Fix a constant $c_{\mathsf{occ}} > 0$, chosen sufficiently large as a function of the constant ρ. Moreover, call a pair (C, A) *admissible* if $C \subseteq [n]$, $|C| \leq t$, and $A \subseteq [n] \setminus C$ with $|A| \geq h$. Here, C represents a possible final corruption set and A a set of uncorrupted contributors.

Definition 6 (Good recursive signing graph). *The recursive signing graph $\mathcal{G}$ is (q, h, t, η)-good if the following properties hold simultaneously.*

1. ***Root safety.*** *For every $S \subseteq [n]$ with $|S| < q$, the root is not S-open.*
2. ***Root liveness.*** *For every admissible pair (C, A), the root is A-live.*

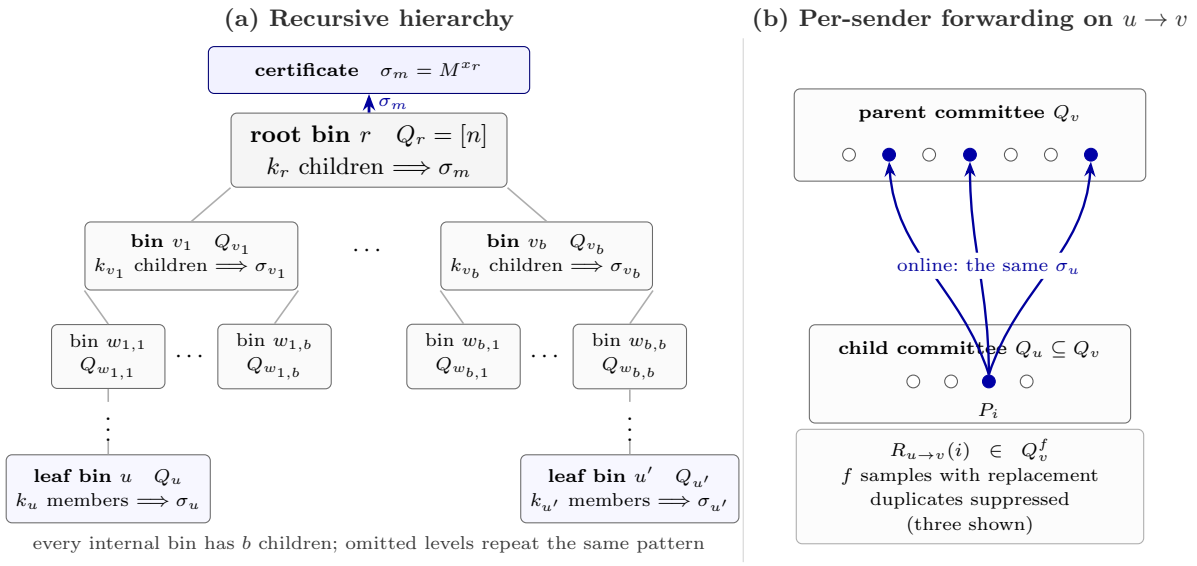


Fig. 2. Recursive sharing and forwarding for threshold signing. Panel (a) shows representative branches of the b-ary hierarchy. An internal bin reconstructs σ_v from k_v child bin signatures, while a leaf reconstructs σ_u from k_u member partial signatures. The root signature σ_m is the transferable certificate. Panel (b) magnifies one edge $u \to v$. Setup samples $Q_u \subseteq Q_v$ and shares downward. Online, every $P_i \in Q_u$ that reconstructs σ_u forwards it to the distinct recipients in $R_{u\to v}(i)$. Three recipient messages of one sender are shown. Independently sampled sibling committees may overlap.

3. ***Robust forwarding.*** *For every admissible pair (C, A), every A-live internal bin v, and every family $(S_u)_{u\in\mathcal{L}_v}$ satisfying $S_u \subseteq Q_u \setminus C$ and $|S_u| \geq (1-\eta)|Q_u \setminus C|$ for every $u \in \mathcal{L}_v$, define*

$$\mathsf{Recv}_C(u, v) := \{P_i \in Q_v \setminus C : \exists P_j \in S_u \text{ such that } P_i \in R_{u\to v}(j)\}.$$

Then all but an η-fraction of $Q_v \setminus C$ belong to $\mathsf{Recv}_C(u, v)$ for at least k_v distinct children $u \in \mathcal{L}_v$.

4. ***Committee load.*** *Writing $\bar{\lambda} := \lambda + \log n$, every party belongs to at most $(c_{\mathsf{occ}}\bar{\lambda})^j$ committees at level j.*

Lemma 1 (Recursive sampler for safety and liveness thresholds). *For the hierarchy sampled in Section 4.1, with $m_* < n$ and constant recursion depth L, sufficiently large choices of ρ, c_*, c_f, and c_{occ} make the hierarchy (q, h, t, η)-good in the sense of Definition 6, except with probability $\exp(-\lambda)$. The root liveness and robust forwarding conclusions also hold for every disjoint pair (C, A) with $|A| \geq h$, without restricting $|C|$. On the same event, every child u of every internal bin v satisfies*

$$\frac{1}{2}p|Q_v| \leq |Q_u| \leq \frac{3}{2}p|Q_v|.$$

Write $\mathcal{E}_{\mathsf{samp}}$ for this joint event. The constants depend only on s, ℓ, η, and L.

4.3 Recursive BLS Sharing

Let $\Gamma = (\mathbb{G}, \mathbb{G}_T, \mathfrak{p}, g, e) \leftarrow \mathsf{BGrpGen}(1^\kappa)$ be bilinear group parameters, where $\mathbb{G}$ and $\mathbb{G}_T$ have prime order $\mathfrak{p}$, g generates $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable, bilinear, and nondegenerate map. We assume $\mathfrak{p} > \max\{n, b\}$. All BLS signatures and partial signatures lie in $\mathbb{G}$, and the pairing permits their public verification against the corresponding published key. The dealer samples $x_r \leftarrow_\$ \mathbb{Z}_\mathfrak{p}$ and publishes $X_r := g^{x_r}$. For every internal bin v, canonically enumerate its children by distinct nonzero evaluation points $\xi_u \in \mathbb{Z}_\mathfrak{p}$, independently sample fresh coefficients

$$(a_{v,1}, \ldots, a_{v,k_v-1}) \leftarrow_\$ \mathbb{Z}_\mathfrak{p}^{k_v-1}, \qquad \text{set} \qquad F_v(X) := x_v + \sum_{d=1}^{k_v-1} a_{v,d}X^d,$$

and define $x_u := F_v(\xi_u)$ for every child u. For every leaf v, assign its members distinct nonzero evaluation points $\xi_{v,i}$, independently sample fresh coefficients

$$(a_{v,1}^{\mathsf{leaf}}, \ldots, a_{v,k_v-1}^{\mathsf{leaf}}) \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}^{k_v-1}, \qquad \text{set} \qquad F_v^{\mathsf{leaf}}(X) := x_v + \sum_{d=1}^{k_v-1} a_{v,d}^{\mathsf{leaf}} X^d,$$

and give $x_{v,i} := F_v^{\mathsf{leaf}}(\xi_{v,i})$ to $P_i \in Q_v$. The dealer publishes $X_v := g^{x_v}$ and $X_{v,i} := g^{x_{v,i}}$ for every bin and every occurrence of a party in a leaf committee, and then erases every scalar secret except the shares delivered to parties.

Lemma 2 (Access structure for safety and liveness thresholds). *For every bin v and party set $S \subseteq [n]$, the exponent x_v is a public linear combination of the scalar leaf shares held by S below v if and only if v is S-open.*

4.4 Protocol

Model $H_{\mathsf{BLS}} : \{0,1\}^* \to \mathbb{G}$ as a random oracle. For an instance identified by (sid, m), define $M := H_{\mathsf{BLS}}(\mathsf{sid}, m) \in \mathbb{G}$. For every non-root bin v, write $\sigma_v := M^{x_v}$ for its bin signature, which is an intermediate partial signature toward the root signature.

A party contributes its own input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$ by executing $\Pi.\mathsf{contribute}(m, \pi)$ for the signing instance Π. A leaf signature share is *authorized* if its owner releases it as part of contributing a signature for its own input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$ for the signed value m. Every partial or reconstructed signature is sent together with a justifier π satisfying $J_{\text{IN}}(m, \pi) = \top$, which the recipient verifies before processing the signature. All parties execute the reconstruction and forwarding rules after this verification, including those with a different input or no input. For forwarding, a party retains any one verified input justifier for m; receiving it does not authorize releasing its own leaf partial signatures.

Signature contribution and leaf reconstruction. To contribute a signature, party P_i sends $\sigma_{v,i} := M^{x_{v,i}}$ to every member of each leaf v containing that party. A recipient accepts this partial signature only if $e(\sigma_{v,i}, g) = e(M, X_{v,i})$. After accepting valid leaf partial signatures from k_v distinct members, it interpolates them in the exponent to obtain $\sigma_v := M^{x_v}$.

Recursive forwarding and reconstruction. Whenever $P_i \in Q_u$ first reconstructs a child signature σ_u, it sends that signature to every recipient in $R_{u \to v}(i)$, where v is the parent of u. A member of Q_v accepts σ_u only if $e(\sigma_u, g) = e(M, X_u)$, and after accepting valid signatures from k_v distinct children it interpolates them in the exponent to obtain σ_v.

Root output. At the root, a party outputs the unique transferable signature $\sigma_m := M^{x_r}$. Anyone verifies it by checking $e(\sigma_m, g) = e(H_{\mathsf{BLS}}(\mathsf{sid}, m), X_r)$.

4.5 Computational Assumption

We use the algebraic one-more computational Diffie–Hellman (AOMCDH) assumption for $\mathsf{BGrpGen}$ [5,4]. For a polynomially bounded integer $N \geq 1$, the challenger samples $\Gamma \leftarrow \mathsf{BGrpGen}(1^\kappa)$, $H \leftarrow_{\$} \mathbb{G}$, and $z_0, \ldots, z_N \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$, and gives $(\Gamma, H, g^{z_0}, \ldots, g^{z_N})$ to the adversary. The adversary may make at most N queries to a discrete logarithm oracle that, on input $(a_0, \ldots, a_N) \in \mathbb{Z}_{\mathsf{p}}^{N+1}$, returns $\sum_{j=0}^{N} a_j z_j$. The assumption states that no probabilistic polynomial-time adversary can output $H^{z_0}, \ldots, H^{z_N}$ except with negligible probability.

Remark 1 (Relation to algebraic one-more discrete logarithm). Bacho et al. [4] give an AGM proof for the nontrivial implication that hardness of algebraic one-more discrete logarithm (AOMDL) implies hardness of AOMCDH. The converse reduction is immediate: after recovering $z_0, \ldots, z_N$, one can compute $H^{z_0}, \ldots, H^{z_N}$. Thus, for matching games and parameters, AOMDL and AOMCDH are equivalent as hardness assumptions for algebraic adversaries. Here AOMDL denotes the algebraic one-more discrete logarithm problem, a falsifiable variant of the one-more discrete logarithm problem.

Our AOMCDH assumption is formulated directly for $\Gamma \leftarrow \mathsf{BGrpGen}(1^\kappa)$ and therefore already gives the adversary access to the pairing. In the bilinear AGM, pairing evaluations produce elements of $\mathbb{G}_T$

and, absent an efficient map back to $\mathbb{G}$, do not introduce additional source group elements. In particular, $e(H, g^{z_j}) = e(H^{z_j}, g)$ allows one to verify a candidate H^{z_j} but does not compute $H^{z_j} \in \mathbb{G}$. Thus, the source group AGM argument remains applicable when AOMDL and discrete logarithm are assumed hard under the same pairing access.

4.6 Security and Complexity

Theorem 1 (Recursive threshold signing). *Under the setup and computational assumptions of this section, fix parameters satisfying $t < q < h = q + \lceil an \rceil \le n - t$, a constant $L \ge 1$, and $\lambda = \Omega(\kappa + \log n)$. There exists a protocol satisfying Definition 5 against strongly adaptive corruptions with message takebacks, except with probability $\exp(-\lambda) + \mathrm{negl}(\kappa)$. For input justifiers of at most B bits, all required outputs occur within $L + 1$ asynchronous time units after h honest input contributions. On the setup event $\mathcal{E}_{\mathsf{samp}}$, defined in the proof and having probability at least $1 - \exp(-\lambda)$, the total outgoing communication before corruption, in bits, is*

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right](\kappa + B)\right).$$

The expected maximum outgoing communication per party before corruption, in bits and without conditioning on $\mathcal{E}_{\mathsf{samp}}$, is

$$\widetilde{O}\left(\left[\lambda + \left(\frac{n}{\lambda}\right)^{1/L}\right](\kappa + B)\right).$$

Corollary 1 (Existential contributor set). *Under the assumptions of Theorem 1, except with probability $\exp(-\lambda) + \mathrm{negl}(\kappa)$, the following holds simultaneously for every tagged message $\mu = (\mathsf{sid}, m)$. Let T be the first execution event at which a valid root signature for μ is output by a protocol party or presented by the adversary. Let $C(T) \subseteq [n]$ be the set of parties corrupted by time T, and let $I_\mu(T) \subseteq [n] \setminus C(T)$ be the set of parties uncorrupted at T that have released at least one of their own authorized leaf signature shares for μ by that time. Then there exists a set $S_\mu(T) \subseteq C(T) \cup I_\mu(T)$ with $|S_\mu(T)| = q$. We call $S_\mu(T)$ an* existential contributor set*. It is only an analytical witness and need not be recoverable from the signature. Members of $S_\mu(T) \cap C(T)$ need not have contributed a signature through the protocol, since the adversary has their secret shares; every member of $S_\mu(T) \setminus C(T)$ has released an authorized leaf partial signature. If C is the final corruption set and $c := |C|$, then at least $q - c$ members of $S_\mu(T)$ remain honest throughout the execution and released an authorized share for μ by time T.*

Corollary 2 (Exclusion of conflicting signatures). *Suppose that $q > \frac{n+t}{2}$, and consider two differently tagged value-specific signing instances at the same logical step. If every honest party contributes a signature for at most one of these instances, then valid root signatures cannot be produced for both instances, except with probability $\exp(-\lambda) + \mathrm{negl}(\kappa)$.*

5 Byzantine Agreement Components

We present and analyze the building blocks for our Byzantine agreement protocol. Proofs for this section are in Appendix D.

5.1 Certificate Dissemination

A public disperser disseminates a transferable certificate from a constant fraction of the parties to all but a prescribed constant fraction.

Each instance is associated with an input justification predicate J_{IN} and a value m, and disseminates certificates π satisfying $J_{\text{IN}}(m, \pi) = \top$. The following definition holds except with negligible probability against strongly adaptive corruptions with message takebacks.

Definition 7 (Certificate dissemination). *Let $0 < a, \eta < 1$ be constants with $\lceil an \rceil \le n - t$. A protocol is an (a, η)-certificate dissemination protocol with input justification predicate J_{IN} if it satisfies the following properties.*

1. ***Liveness.*** *If at least $\lceil an \rceil$ honest parties contribute input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$ to the instance for m, then all but $\lfloor \eta n \rfloor$ honest parties output a pair (m, π') satisfying $J_{\text{IN}}(m, \pi') = \top$.*
2. ***Justified output.*** *Every value m output by an honest party is J_{IN}-justified and is accompanied by a justifier π' satisfying $J_{\text{IN}}(m, \pi') = \top$.*

A Public Disperser. Fix constants $0 < a, \eta < 1$ and a statistical security parameter $\lambda \geq 1$. Let $D = ([n], [n], E)$ be a directed bipartite graph with one left and one right vertex for each party. For $A \subseteq [n]$, write

$$N_D(A) := \{j \in [n] : \exists i \in A \text{ s.t. } (i, j) \in E\}.$$

Definition 8 (Disperser). *The graph D is an (a, η)-dissemination disperser if every $A \subseteq [n]$ with $|A| \geq \lceil an \rceil$ satisfies $|[n] \setminus N_D(A)| \leq \lfloor \eta n \rfloor$.*

Define $d_{\mathsf{D}} := \min\{n, \lceil (\ln 4 + \lambda/n)/(a\eta) \rceil\}$.

$\mathsf{Setup}_{\mathsf{dissem}}(1^\lambda, n, a, \eta)$

Construct the public graph D as follows.

1. If $d_{\mathsf{D}} = n$, set D to the complete directed bipartite graph on $[n] \times [n]$.
2. Otherwise, independently for every party P_i, sample d_{D} recipients uniformly from $[n]$ with replacement, suppress duplicate recipients, and publish the resulting list as the out-neighborhood of i.

Fig. 3. Setup for certificate dissemination.

Lemma 3 (Sampled disperser). *The graph output by $\mathsf{Setup}_{\mathsf{dissem}}(1^\lambda, n, a, \eta)$ is an (a, η)-dissemination disperser except with probability at most $\exp(-\lambda)$. Its maximum left degree satisfies*

$$d_{\mathsf{D}} \leq \frac{\ln 4 + \lambda/n}{a\eta} + 1.$$

Since a and η are fixed constants, this is $d_{\mathsf{D}} = O(1 + \lambda/n)$.

The public graph is reused across instances; its disperser property holds for every sufficiently large source set, including sets chosen adaptively after setup.

Protocol and Security. Each instance disseminates certificates for one value m.

$\Pi_{\mathsf{dissem}}(a, \eta, J_{\mathbf{in}})$

Each party P_i executes the following instructions.

1. To contribute input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$, P_i sends (m, π) to every out-neighbor of i in D.
2. Upon receiving (m, π) with $J_{\text{IN}}(m, \pi) = \top$, P_i outputs (m, π).

Receiving an output before its own input does not prevent a party from subsequently executing the send rule.

Fig. 4. One-round certificate dissemination.

Theorem 2 (Certificate dissemination). *For constants $0 < a, \eta < 1$ with $\lceil an \rceil \leq n - t$, there exists a protocol satisfying Definition 7 against strongly adaptive corruptions with message takebacks, except with probability $\exp(-\lambda)$ over the setup. All required outputs occur within one asynchronous time unit after $\lceil an \rceil$ honest input contributions. For input certificates of at most B bits, the total outgoing communication before corruption is $O((n + \lambda)(B + \kappa + \log n))$ bits.*

5.2 Binary Sign-or-Disseminate

Recursive proxcensus advances using either a threshold certificate for one value or justifiers for two adjacent values [30]. We mimic this logic through *threshold signing* and *certificate dissemination*, with explicit liveness thresholds and bounds on missing outputs. Each primitive accepts arbitrary values, while their combined *binary sign-or-disseminate* guarantee applies when almost all honest parties hold one of at most two justified candidate values: either almost all obtain a new threshold signature for one candidate, or almost all obtain existing certificates for both candidates. The alternatives are inclusive, and the candidates need not be bits; in recursive proxcensus they are at most two adjacent grades.

In the quadratic protocol for $n > 3t$, each party has a binary input and a corresponding validity justifier. If both justified inputs occur among honest parties, all-to-all communication distributes both justifiers to every honest party and thereby certifies that both inputs are justified. Otherwise, at least $n-t$ honest parties have the same input, and their contributions form an $(n-t)$-threshold certificate for that input.

These two types of certificates serve as justifiers for the following protocol step. Quorum intersection ensures that a threshold certificate can form for at most one input, while dissemination preserves at most the two justified inputs from the preceding step. The same binary step can therefore be composed recursively and used to reach agreement.

Parameters for the BA instantiations. For the instantiations in this section and Section 6, fix constants $\delta, \epsilon > 0$ satisfying $t \leq (1-\delta)n/3$ and $6\epsilon < \delta$. We assume throughout these instantiations that n is sufficiently large whenever additive rounding terms are absorbed by these constant margins. The quantity ϵn bounds missing honest inputs between steps, sets the dissemination threshold, and gives the difference between the signing protocol's liveness and safety thresholds for input validation and proxcensus, up to rounding.

Every certificate dissemination instance below is instantiated as an $(\epsilon, \eta_{\mathsf{D}})$-certificate dissemination protocol using Theorem 2. The threshold signing instances used for input validation and recursive proxcensus are instantiated as $(q, \epsilon, \eta_{\mathsf{C}})$-threshold signing protocols using Theorem 1, for the corresponding values of q. The common coin uses the separate parameters a_{coin} and η_{coin} defined in Section 5.5. Define $\eta_{\mathsf{P}} := \max\{\eta_{\mathsf{C}}, 2\eta_{\mathsf{D}}\}$, and choose the parameters bounding missing outputs so that $\eta_{\mathsf{P}} \leq \epsilon$.

To ensure that conflicting threshold certificates cannot both be produced, set

$$q_{\mathsf{uniq}} := \left\lfloor \frac{n+t}{2} \right\rfloor + 1, \qquad h_{\mathsf{uniq}} := q_{\mathsf{uniq}} + \lceil \epsilon n \rceil.$$

Since

$$q_{\mathsf{uniq}} = \left\lfloor \frac{n+t}{2} \right\rfloor + 1 > \frac{n+t}{2},$$

Corollary 2 ensures that conflicting q_{uniq}-threshold signatures cannot both be produced when every honest party releases its shares for at most one candidate at that logical step. This is an existential consequence of unforgeability and does not require the threshold signature to reveal which parties released shares.

For each candidate value m_b, run a threshold signing instance and a certificate dissemination instance with the same input justification predicate J_{IN}. A party with its own J_{IN}-justified input value m_b and an associated justifier π contributes input (m_b, π) to both instances. The signing instance produces a new certificate of support for m_b; the dissemination instance distributes m_b together with its associated justifier. The following lemma derives the required output using only the liveness properties in Definitions 5 and 7. Figure 1 illustrates how the two thresholds ensure progress.

Lemma 4 (Binary sign-or-disseminate). *Let m_0 and m_1 be two distinct candidate values, and let h_0 and h_1 count the honest parties with J_{IN}-justified inputs that contribute them to both primitives for these respective values, with each counted honest party contributing to at most one candidate. A candidate to which no honest party contributes input has count zero. Suppose $h_0 + h_1 \geq H$. Let the threshold signing instances have parameters $(q, a_{\mathsf{C}}, \eta_{\mathsf{C}})$, let the dissemination instances implement $(a_{\mathsf{D}}, \eta_{\mathsf{D}})$-certificate dissemination, and define $h_{\mathsf{C}} := q + \lceil a_{\mathsf{C}} n \rceil$, $h_{\mathsf{D}} := \lceil a_{\mathsf{D}} n \rceil$. If $H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}$, then at least one of the following holds for an index $b \in \{0,1\}$ maximizing h_b.*

1. *All but at most $\lfloor \eta_{\mathsf{C}} n \rfloor$ honest parties output a valid q-threshold signature for m_b.*

2. *All but at most* $2\lfloor \eta_{\mathsf{D}} n \rfloor$ *honest parties output both* J_{IN}*-justified values* m_b *and* m_{1-b} *together with associated justifiers.*

For recursive proxcensus, take $H := n - t - \epsilon n$, $h_{\mathsf{D}} := \lceil \epsilon n \rceil$, and $h_{\mathsf{C}} := h_{\mathsf{uniq}}$. The difference $(n-t) - (n+t)/2 = (n-3t)/2$ leaves room for missing inputs, support for the less common value, and the signing protocol's liveness margin. Using

$$h_{\mathsf{uniq}} \leq (n+t)/2 + \epsilon n + 2, \qquad t \leq (1-\delta)n/3, \qquad \text{and} \qquad 6\epsilon < \delta$$

gives

$$n - t - 2\epsilon n - h_{\mathsf{uniq}} \geq (\delta/2 - 3\epsilon)n - 2 > 0.$$

Thus $H - h_{\mathsf{D}} + 1 \geq n - t - 2\epsilon n > h_{\mathsf{uniq}}$, which verifies $H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}$ in Lemma 4 and also gives $h_{\mathsf{uniq}} \leq n - t$.

Under this instantiation, either conclusion of Lemma 4 makes the required threshold signature, or both required disseminated certificates, available to all but $\eta_{\mathsf{P}} n$ honest parties.

5.3 Input Validation

The input validation protocol converts an unjustified binary input into a binary input justified to have been proposed by an honest party [14]. Define the trivial input justification predicate by $J_{\text{IN}}^{\mathsf{raw}}(m, \pi) := \top$ for all $m, \pi \in \{0,1\}^*$. Since this predicate is always true, raw inputs require no justifier; we use $\perp$ as their canonical justifier and omit it from messages. Input validation represents every raw justifier as $\perp$ when contributing, reconstructing, or forwarding signatures. Set $q_{\mathsf{valid}} := t + 1$ and $h_{\mathsf{valid}} := q_{\mathsf{valid}} + \lceil \epsilon n \rceil$. The safety threshold ensures that every valid certificate has an honest contributor. Honest support may be split between the two bits, so the liveness threshold must be reached by the more common bit even when some honest inputs are missing.

For each bit b, parties use a separate $(q_{\mathsf{valid}}, \epsilon, \eta_{\mathsf{C}})$-threshold signing instance with input justification predicate $J_{\text{IN}}^{\mathsf{raw}}$. Upon receiving input b_i, party P_i contributes $(b_i, \perp)$ only to the signing instance for b_i. Upon receiving its first valid q_{valid}-threshold certificate σ_b, it outputs (b, σ_b). Parties without an input still perform the prescribed reconstruction and forwarding and may output a validated pair. The output justification predicate satisfies $J_{\text{OUT}}^{\mathsf{valid}}(b, \sigma_b) = \top$ if and only if σ_b is a valid q_{valid}-threshold certificate for the input validation step and b.

Lemma 5 (Input validation). *Suppose that all but* ϵn *honest parties have a binary input. Then all but* $\eta_{\mathsf{C}} n$ *honest parties output a* $J_{\text{OUT}}^{\mathsf{valid}}$*-justified bit together with its justifier. Moreover, every* $J_{\text{OUT}}^{\mathsf{valid}}$*-justified bit was the input of at least one honest party.*

If both bits have sufficient support, certificates for both bits may be produced, and each certificate separately establishes that its bit was proposed by an honest party.

5.4 Justified Almost-Everywhere Proxcensus

We retain the justified graded agreement and justified validity properties of justified proxcensus [30], while relaxing its termination guarantee to almost-everywhere termination.

Definition 9 (Almost-everywhere proxcensus). *The protocol takes binary inputs with input justification predicate* J_{IN} *and defines an output justification predicate* $J_{\text{OUT}}^{(G)}$ *for grades in* $\{0, \ldots, G-1\}$*. It satisfies the following properties except with negligible probability.*

1. ***Almost-everywhere termination.*** *If all but* ϵn *honest parties have a* J_{IN}*-justified binary input, then all but* $\eta_{\mathsf{P}} n$ *honest parties produce a* $J_{\text{OUT}}^{(G)}$*-justified output.*
2. ***Justified graded agreement.*** *Any two* $J_{\text{OUT}}^{(G)}$*-justified grades* y *and* y' *satisfy* $|y - y'| \leq 1$*.*
3. ***Justified validity.*** *If* b *is the only* J_{IN}*-justified bit, then every* $J_{\text{OUT}}^{(G)}$*-justified output has value* $b(G-1)$*.*

Recursive Construction. We construct protocols $\Pi_{\mathsf{prox}(G)}(J_{\text{IN}})$ for $G \in \{2^k + 1 : k \geq 1\}$ using a two-grade identity protocol as the recursion base. The base protocol $\Pi_{\mathsf{prox}(2)}(J_{\text{IN}})$ outputs its J_{IN}-justified input unchanged and sets $J_{\text{OUT}}^{(2)} := J_{\text{IN}}$. At most ϵn honest parties lack an input and hence an output; the first recursive step reduces the number of missing honest outputs to at most $\eta_{\mathsf{P}} n$. For the recursive step, suppose that $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$ and its output justification predicate $J_{\text{OUT}}^{(s)}$ have been defined. We construct $\Pi_{\mathsf{prox}(2s-1)}(J_{\text{IN}})$ as follows.

$\Pi_{\mathsf{prox}(2s-1)}(J_{\mathbf{in}})$

The input justification predicate J_{IN} is a parameter of the protocol; the underlying s-grade protocol uses the same predicate and defines $J_{\text{OUT}}^{(s)}$. For every $z \in \{0, \ldots, s-1\}$, let Dissem_z be the value-specific $(\epsilon, \eta_{\mathsf{D}})$-certificate dissemination instance and let Sign_z be the value-specific $(q_{\mathsf{uniq}}, \epsilon, \eta_{\mathsf{C}})$-threshold signing instance, both with input justification predicate $J_{\text{OUT}}^{(s)}$. Each party P_i executes the following logic.

- Upon receiving its J_{IN}-justified binary input b and associated justifier π_b, P_i contributes input (b, π_b) to $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$.
- Upon obtaining its own output (z, π_z) from $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$, P_i executes $\mathsf{Dissem}_z.\mathsf{contribute}(z, \pi_z)$ and then executes $\mathsf{Sign}_z.\mathsf{contribute}(z, \pi_z)$.
- Upon obtaining a valid threshold certificate σ_z for Sign_z, including one received as a disseminated grade justifier, P_i outputs $(2z, \sigma_z)$ if it has not yet output.
- Upon obtaining outputs $(z-1, \pi_{z-1})$ and (z, π_z) from Dissem_{z-1} and Dissem_z, respectively, P_i outputs $(2z-1, \mathsf{Compact}_s(z-1, \pi_{z-1}, z, \pi_z))$ if it has not yet output, where $\mathsf{Compact}_s$ is defined in Appendix B.

Justification predicates. The predicate $J_{\text{OUT}}^{(2s-1)}$ accepts (y, π) iff either:

- $y = 2z$ for some $z \in \{0, \ldots, s-1\}$ and π is a valid q_{uniq}-threshold certificate for the value-specific instance Sign_z at this recursion level; or
- $y = 2z - 1$ for some $z \in \{1, \ldots, s-1\}$ and π is a canonical compact justifier accepted by $\mathsf{Compact}_s$ for the two adjacent lower-level grades $z-1$ and z.

Fig. 5. Recursive construction of $(2s-1)$-proxcensus from s-proxcensus.

The compact relation is defined and proved sound in Appendix B.

Lemma 6 (Recursive proxcensus). *Suppose that $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$ satisfies justified graded agreement and justified validity, and supplies an output to all but ϵn honest parties whenever all but ϵn honest parties have a J_{IN}-justified input. Assume also that all value-specific threshold signing and certificate dissemination instances used in Figure 5 satisfy, respectively, the liveness and unforgeability properties in Definition 5 and the liveness and justified output properties in Definition 7, with the parameters fixed in Section 5.2. Then $\Pi_{\mathsf{prox}(2s-1)}(J_{\text{IN}})$ implements justified almost-everywhere $(2s-1)$-proxcensus as in Definition 9.*

Starting from $\Pi_{\mathsf{prox}(2)}$, repeated application of Lemma 6 constructs justified almost-everywhere G-proxcensus for every $G = 2^k + 1$ with $k \geq 1$, because $\eta_{\mathsf{P}} \leq \epsilon$. Composing the input validation protocol with this construction yields proxcensus outputs whose validity ultimately traces to an input of an honest party.

Compact justifiers. The compact justifier representation is adapted from [29, Appendix A]. Under the two-value anchoring property of Definition 11, input justifiers of at most B bits yield compact grade justifiers of length $B + O(\kappa)$. Appendix B gives a self-contained definition of $\mathsf{Compact}_s$ and proves Lemma 7.

5.5 Common Coin

Definition 10 (Almost-everywhere common coin). *An almost-everywhere common coin Π_{coin} defines an output justification predicate $J_{\text{OUT}}^{(\mathsf{coin},r)}$ for each iteration r and outputs a $J_{\text{OUT}}^{(\mathsf{coin},r)}$-justified bit c_r with an associated transferable justifier π_r^{coin}. It satisfies the following properties.*

1. ***Almost-everywhere termination.*** *All but $\eta_{\mathsf{coin}} n$ honest parties obtain an output whenever all but $\eta_{\mathsf{P}} n$ honest parties contribute input.*

2. ***Justified agreement.*** *Any two $J_{\mathrm{OUT}}^{(\mathsf{coin},r)}$-justified bits are equal.*
3. ***Uniformity and unpredictability.*** *The bit remains hidden until an honest party has contributed input. For the analysis, consider an experiment in which the first exposure of the coin reveals an independent uniform bit; the real execution differs only with negligible probability over each polynomial execution prefix.*

We model $H_{\mathsf{coin}} : \{0,1\}^* \to \{0,1\}$ as an independent bit-valued random oracle. In the construction below, $H_{\mathsf{coin}}(\mathsf{sid}, \sigma)$ is the binary coin value. Choose η_{coin} so that $\eta_{\mathsf{P}} + \eta_{\mathsf{coin}} \le \epsilon$.

All honest contributors to the coin instance contribute a signature for the same fixed message. The safety threshold must exceed t to keep the coin hidden before an honest signature contribution, while its liveness threshold must accommodate the parties missing a proxcensus output. The corruption bound leaves an $\Omega(n)$ gap between the safety threshold $q_{\mathsf{coin}} = t + 1$ and the number $n - t - \eta_{\mathsf{P}} n$ of honest contributors available for liveness.

Corollary 3 (Common coin from threshold signing). *Fix parameters (q, a, η) satisfying the hypotheses of Theorem 1. For each coin instance, set $m := \texttt{coin}$ and let sid encode the BA session and iteration. Run threshold signing on (sid, m), with signed group element $M := H_{\mathsf{BLS}}(\mathsf{sid}, m)$ as in Section 4.4. Then $c := H_{\mathsf{coin}}(\mathsf{sid}, \sigma_m) \in \{0,1\}$ is an almost-everywhere common coin as in Definition 10, with liveness threshold $h = q + \lceil an \rceil$ and at most ηn missing honest outputs. Its value remains computationally hidden until at least $q - t > 0$ honest parties have contributed a signature. The signature σ_m is a transferable justifier for c. The protocol takes at most $L + 1$ asynchronous time units and has the communication complexity stated in Theorem 1.*

For BA, set

$$q_{\mathsf{coin}} := t + 1, \qquad h_{\mathsf{coin}} := \lceil n - t - \eta_{\mathsf{P}} n \rceil, \qquad a_{\mathsf{coin}} := \frac{h_{\mathsf{coin}} - q_{\mathsf{coin}}}{n}.$$

Since $\eta_{\mathsf{P}} \le \epsilon$ and $t \le (1-\delta)n/3$,

$$\begin{aligned} h_{\mathsf{coin}} - q_{\mathsf{coin}} &\ge n - 2t - \eta_{\mathsf{P}} n - 1 \\ &\ge \left(\frac{1+2\delta}{3} - \eta_{\mathsf{P}}\right) n - 1 \\ &\ge \left(\frac{1+2\delta}{3} - \epsilon\right) n - 1 \\ &= \Omega(n). \end{aligned}$$

Moreover, $h_{\mathsf{coin}} \le n - t$. Hence, for all sufficiently large n, $t < q_{\mathsf{coin}} < h_{\mathsf{coin}} \le n - t$, and $a_{\mathsf{coin}} = (h_{\mathsf{coin}} - q_{\mathsf{coin}})/n$ is bounded below by a positive constant. Instantiate Corollary 3 with $(q, a, \eta) = (q_{\mathsf{coin}}, a_{\mathsf{coin}}, \eta_{\mathsf{coin}})$. Whenever all but $\eta_{\mathsf{P}} n$ honest parties contribute input, their number is at least h_{coin}, even when the final corruption set has size t. The common coin bit is $c_r := H_{\mathsf{coin}}(\mathsf{sid}, \sigma_m) \in \{0,1\}$, and the root signature σ_m is its transferable justifier. The predicate $J_{\mathrm{OUT}}^{(\mathsf{coin},r)}(c, \sigma)$ accepts if and only if σ is a valid root signature on $(\mathsf{sid}, \texttt{coin})$ for iteration r and $c = H_{\mathsf{coin}}(\mathsf{sid}, \sigma)$. The first query by any party or the adversary to the independent coin oracle at this root signature exposes the coin; the safety threshold $q_{\mathsf{coin}} > t$ ensures that producing the signature requires an honest signature contribution.

6 Byzantine Agreement

We combine the components of Section 5 into binary agreement, using the parameters of Section 5.2. Proofs for this section are in Appendix E.

6.1 The Agreement Protocol

We follow the reduction from justified proxcensus and a common coin [24], accounting for missing component outputs in the almost-everywhere setting.

Fix $G \geq 6$ such that $\Pi_{\mathsf{prox}(G)}$ is available. For the recursive construction in Section 5.4, we may take $G = 9$. Grades 0 and $G-1$ justify decisions, grades 1 and $G-2$ preserve the corresponding bit, and the nonempty set of middle grades $\{2, \ldots, G-3\}$ uses the common coin. The condition $G \geq 6$ ensures that there are at least two middle grades.

Decision dissemination. A *decision certificate* is a pair (b, ζ) satisfying the output justification predicate $J^{\mathsf{BA}}_{\mathrm{OUT}}(b, \zeta) = \top$ defined in Figure 6. For each bit b, use one instance of $\Pi_{\mathsf{dissem}}(\epsilon, \eta_{\mathsf{D}}, J^{\mathsf{BA}}_{\mathrm{OUT}})$ for the BA session on the existing public dissemination graph D. A party obtaining a decision certificate contributes no further signatures; it disseminates the certificate once and terminates, as specified below.

If a decision is justified in iteration r, justified graded agreement restricts every $J^{(G,r)}_{\mathrm{OUT}}$-justified grade to $\{0, 1\}$ or to $\{G-2, G-1\}$. Neither pair contains a middle grade, so no party needs the coin for that iteration to decide or to contribute input to the next iteration.

Combining this protocol with input validation gives binary agreement with almost-everywhere termination and the following complexity bounds.

Theorem 3 (Binary agreement with almost-everywhere termination). *Fix constants $\delta > 0$, $0 < \nu < 1$, and an integer $c \geq 1$. There is a constant $\epsilon > 0$ such that, for all sufficiently large $n \leq \kappa^c$ and $t \leq (1-\delta)n/3$, there exists a protocol satisfying Definition 2 with parameters (ϵ, ν) against strongly adaptive corruptions with message takebacks, except with negligible probability. The protocol assumes authenticated reliable asynchronous channels, the setup and computational assumptions of Section 4, and the independent random oracle of Section 5.5. If all but ϵn honest parties receive an input, its expected complexity is $O(1)$ asynchronous rounds, $O(n\kappa^2)$ total bits, and $\widetilde{O}(\kappa^2)$ maximum outgoing bits per party. Communication is measured in bits and counted before each party's corruption, and expectations are conditioned on $\mathcal{E}_{\mathsf{BA}}$ defined in the proof of Lemma 10, which has probability $1 - \mathrm{negl}(\kappa)$.*

6.2 From Almost-Everywhere to Full Termination

We now consider adaptive corruptions without message takebacks, as defined in Section 3. In this model, sending decision certificates to additional random recipients extends termination to every honest party while preserving the asymptotic complexity bounds. Each party samples each recipient using fresh private randomness and reveals it only when sending, at which point the send is irrevocable. The adversary may corrupt a party between any two sends.

Let $\alpha_{\mathsf{out}} := 1 - \frac{t}{n} - \eta_{\mathsf{P}} > 0$ and set

$$d := \left\lceil \frac{\lambda + \ln n}{\alpha_{\mathsf{out}}} \right\rceil .$$

Each party runs the binary agreement protocol Π_{BA} of Figure 6. Upon first obtaining a $J^{\mathsf{BA}}_{\mathrm{OUT}}$-justified decision bit b and associated justifier π, whether locally or in a received message, it outputs (b, π), contributes (b, π) to the decision dissemination instance for b, and performs d sequential point-to-point sends of (b, π), sampling a fresh uniform recipient privately for each send. It completes both decision dissemination and these additional sends before terminating.

Corollary 4 (Linear full binary agreement). *Under the assumptions of Theorem 3, there exists a protocol satisfying Definition 3 against adaptive corruptions without message takebacks, except with negligible probability. When every honest party receives an input, it has expected $O(1)$ asynchronous rounds, $O(n\kappa^2)$ total bits, and $\widetilde{O}(\kappa^2)$ maximum outgoing bits per party before corruption. These expectations are conditioned on $\mathcal{E}_{\mathsf{full}}$ defined in the proof of Corollary 4, which has probability $1 - \mathrm{negl}(\kappa)$.*

References

1. Abraham, I., Chan, T.H., Dolev, D., Nayak, K., Pass, R., Ren, L., Shi, E.: Communication complexity of byzantine agreement, revisited. In: PODC. pp. 317–326. ACM (2019)
2. Abraham, I., Chouatt, E., Gilad, Y., Stern, G., Yakoubov, S.: Asynchronous algorand: Reaching agreement with near linear communication and constant expected time. In: PODC. pp. 28–38. ACM (2025). https://doi.org/10.1145/3732772.3733533

$\Pi_{\mathsf{BA}}(J^{(0)}_{\mathbf{in}})$

The initial input justification predicate $J^{(0)}_{\text{IN}}$ is a parameter of the protocol. For each iteration r, the proxcensus instance with input predicate $J^{(r)}_{\text{IN}}$ defines the output predicate $J^{(G,r)}_{\text{OUT}}$. Party P_i uses its $J^{(0)}_{\text{IN}}$-justified input value x^0_i and associated justifier π^0_i when available. Write (g^r_i, ζ^r_i) for its own proxcensus output and $(c_r, \pi^{\mathsf{coin}}_r)$ for its coin output in iteration r. While it has no decision certificate, P_i executes the following rules separately for each r, including when it lacks an input to that iteration.

- When its own $J^{(r)}_{\text{IN}}$-justified estimate x^r_i and associated justifier π^r_i are available, P_i contributes input (x^r_i, π^r_i) to $\Pi_{\mathsf{prox}(G)}(J^{(r)}_{\text{IN}})$.
- Upon obtaining its own $J^{(G,r)}_{\text{OUT}}$-justified grade g^r_i and associated justifier ζ^r_i, P_i acts as follows. If $g^r_i = 0$ or $g^r_i = G-1$, it applies the decision rule below to $(g^r_i/(G-1), \zeta^r_i)$. Otherwise, it contributes the fixed coin message as input to Π_{coin} for iteration r with justifier (g^r_i, ζ^r_i).
- If $g^r_i = 1$, then P_i sets $x^{r+1}_i := 0$ justified by $(1, \zeta^r_i)$; if $g^r_i = G-2$, then P_i sets $x^{r+1}_i := 1$ justified by $(G-2, \zeta^r_i)$.
- If $2 \leq g^r_i \leq G-3$, then upon obtaining its coin output $(c_r, \pi^{\mathsf{coin}}_r)$, P_i sets $x^{r+1}_i := c_r$ justified by $(g^r_i, \zeta^r_i, \pi^{\mathsf{coin}}_r)$.

Decision rule. Upon first obtaining (b, ζ) with $J^{\mathsf{BA}}_{\text{OUT}}(b, \zeta) = \top$, either locally or in a received message, P_i outputs (b, ζ), contributes it to the decision dissemination instance for b, and terminates all instances of the BA session.

Justification predicates. The input justification predicate for the next iteration $J^{(r+1)}_{\text{IN}}$ accepts (b, π) exactly in the following cases.

- The justifier π contains grade $g = 1$ and a string ζ satisfying $J^{(G,r)}_{\text{OUT}}(g, \zeta) = \top$, and $b = 0$.
- The justifier π contains grade $g = G-2$ and a string ζ satisfying $J^{(G,r)}_{\text{OUT}}(g, \zeta) = \top$, and $b = 1$.
- The justifier π contains a grade $g \in \{2, \ldots, G-3\}$, a string ζ satisfying $J^{(G,r)}_{\text{OUT}}(g, \zeta) = \top$, and a string π^{coin}_r satisfying $J^{(\mathsf{coin},r)}_{\text{OUT}}(b, \pi^{\mathsf{coin}}_r) = \top$.

The BA output justification predicate $J^{\mathsf{BA}}_{\text{OUT}}$ accepts (b, ζ) if and only if $b \in \{0, 1\}$ and $J^{(G,r)}_{\text{OUT}}(b(G-1), \zeta) = \top$ for the iteration r encoded in ζ.

Fig. 6. Binary agreement from proxcensus and a common coin, with almost-everywhere termination.

3. Augustine, J., King, V., Molla, A.R., Pandurangan, G., Saia, J.: Scalable and secure computation among strangers: Message-competitive byzantine protocols. In: 34th International Symposium on Distributed Computing (DISC 2020). pp. 31:1–31:19 (2020). https://doi.org/10.4230/LIPIcs.DISC.2020.31
4. Bacho, R., Loss, J., Tessaro, S., Wagner, B., Zhu, C.: Twinkle: Threshold signatures from ddh with full adaptive security. In: Advances in Cryptology – EUROCRYPT 2024: 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26–30, 2024, Proceedings, Part I. p. 429–459. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-58716-0_15, https://doi.org/10.1007/978-3-031-58716-0_15
5. Bauer, B., Fuchsbauer, G., Plouviez, A.: The one-more discrete logarithm assumption in the generic group model. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 587–617. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92068-5_20
6. Bhangale, A., Liu-Zhang, C.D., Loss, J., Nayak, K., Yandamuri, S.: Leader election with poly-logarithmic communication per party. In: Advances in Cryptology – CRYPTO 2025, Part II. pp. 37–68 (2025). https://doi.org/10.1007/978-3-032-01878-6_2, https://eprint.iacr.org/2025/1016
7. Blum, E., Katz, J., Liu-Zhang, C., Loss, J.: Asynchronous byzantine agreement with subquadratic communication. In: TCC (1). Lecture Notes in Computer Science, vol. 12550, pp. 353–380. Springer (2020)
8. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Berlin, Heidelberg (Jan 2003). https://doi.org/10.1007/3-540-36288-6_3
9. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. Journal of Cryptology **17**(4), 297–319 (Sep 2004). https://doi.org/10.1007/s00145-004-0314-9
10. Boyle, E., Cohen, R., Goel, A.: Breaking the $O(\sqrt{(}n))$-bit barrier: Byzantine agreement with polylog bits per party. In: Miller, A., Censor-Hillel, K., Korhonen, J.H. (eds.) 40th ACM PODC. pp. 319–330. ACM (Jul 2021). https://doi.org/10.1145/3465084.3467897
11. Boyle, E., Cohen, R., Goel, A.: Breaking the $O(\sqrt{n})$-bit barrier: Byzantine agreement with polylog bits per party. Journal of Cryptology **37**(1), 2 (Jan 2024). https://doi.org/10.1007/s00145-023-09484-0
12. Boyle, E., Cohen, R., Meyer, P.: Maintaining sublinear locality over time: Adaptively secure MPC on a reusable hidden graph. In: Advances in Cryptology – EUROCRYPT 2026, Part III. pp. 546–572 (2026). https://doi.org/10.1007/978-3-032-25324-8_19, https://eprint.iacr.org/2026/905
13. Boyle, E., Goldwasser, S., Tessaro, S.: Communication locality in secure multi-party computation - how to run sublinear algorithms in a distributed setting. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 356–376. Springer, Berlin, Heidelberg (Mar 2013). https://doi.org/10.1007/978-3-642-36594-2_21
14. Cachin, C., Kursawe, K., Petzold, F., Shoup, V.: Secure and efficient asynchronous broadcast protocols. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 524–541. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_31
15. Cachin, C., Kursawe, K., Shoup, V.: Random oracles in constantipole: practical asynchronous byzantine agreement using cryptography (extended abstract). In: Neiger, G. (ed.) 19th ACM PODC. pp. 123–132. ACM (Jul 2000). https://doi.org/10.1145/343477.343531
16. Canetti, R., Rabin, T.: Fast asynchronous byzantine agreement with optimal resilience. In: 25th ACM STOC. pp. 42–51. ACM Press (May 1993). https://doi.org/10.1145/167088.167105
17. Chandran, N., Chongchitmate, W., Garay, J.A., Goldwasser, S., Ostrovsky, R., Zikas, V.: The hidden graph model: Communication locality and optimal resiliency with adaptive faults. In: Roughgarden, T. (ed.) ITCS 2015. pp. 153–162. ACM (Jan 2015). https://doi.org/10.1145/2688073.2688102
18. Chandran, N., Forghani, P., Garay, J.A., Ostrovsky, R., Patel, R., Zikas, V.: Universally composable almost-everywhere secure computation. In: Dachman-Soled, D. (ed.) ITC 2022. LIPIcs, vol. 230, pp. 14:1–14:25. Schloss Dagstuhl (Jul 2022). https://doi.org/10.4230/LIPIcs.ITC.2022.14
19. Chandran, N., Garay, J.A., Misra, A.K., Ostrovsky, R., Zikas, V.: Adaptive security, erasures, and network assumptions in communication-local MPC. In: Boyle, E., Mahmoody, M. (eds.) TCC 2024, Part IV. LNCS, vol. 15367, pp. 293–326. Springer, Cham (Dec 2024). https://doi.org/10.1007/978-3-031-78023-3_10
20. Damgård, I.B., Kamp, S.H., Loss, J., Nielsen, J.B.: Asynchronous YOSO a la paillier. Cryptology ePrint Archive, Report 2025/128 (2025), https://eprint.iacr.org/2025/128
21. Dolev, D., Reischuk, R.: Bounds on information exchange for byzantine agreement. J. ACM **32**(1), 191–204 (1985)
22. Ducros, C., Loss, J., Rambaud, M.: Strong efficiency lower bounds for byzantine agreement. In: International Symposium on Distributed Computing (DISC) (2026), https://eprint.iacr.org/2026/479, full version: Cryptology ePrint Archive, Paper 2026/479
23. Dwork, C., Peleg, D., Pippenger, N., Upfal, E.: Fault tolerance in networks of bounded degree. SIAM J. Comput. **17**(5), 975–988 (1988)
24. Elsheimy, F., Kamp, S.H., Loss, J., Nielsen, J.B.: Optimal best-of-both-worlds consensus. In: CRYPTO (10). Lecture Notes in Computer Science, vol. 16809, pp. 3–34. Springer (2026), https://eprint.iacr.org/2026/239

25. Feng, H., Mai, T., Tang, Q.: Balanced and adaptively secure asynchronous common coin and byzantine agreement with sub-quadratic communication. In: PODC. pp. 271–281. ACM (2026). https://doi.org/10.1145/3796701.3815954
26. Feng, H., Mai, T., Tang, Q.: Balanced and adaptively secure asynchronous common coin and byzantine agreement with sub-quadratic communication. Cryptology ePrint Archive, Report 2026/998 (2026), https://eprint.iacr.org/2026/998, full version
27. Fernando, R., Gelles, Y., Komargodski, I.: Scalable distributed agreement from LWE: Byzantine agreement, broadcast, and leader election. In: Guruswami, V. (ed.) ITCS 2024. vol. 287, pp. 46:1–46:23. LIPIcs (Jan / Feb 2024). https://doi.org/10.4230/LIPIcs.ITCS.2024.46
28. Gelles, Y., Komargodski, I.: Optimal load-balanced scalable distributed agreement. In: Mohar, B., Shinkar, I., O'Donnell, R. (eds.) 56th ACM STOC. pp. 411–422. ACM Press (Jun 2024). https://doi.org/10.1145/3618260.3649736
29. Kamp, S.H.: A new way to achieve round-efficient asynchronous byzantine agreement. Cryptology ePrint Archive, Report 2025/143 (2025), https://eprint.iacr.org/2025/143
30. Kamp, S.H.: Short paper: A new way to achieve round-efficient asynchronous byzantine agreement. In: Garman, C., Moreno-Sanchez, P. (eds.) FC 2025, Part II. LNCS, vol. 15752, pp. 160–169. Springer, Cham (Apr 2025). https://doi.org/10.1007/978-3-032-07035-7_10
31. Kapron, B.M., Kempe, D., King, V., Saia, J., Sanwalani, V.: Fast asynchronous byzantine agreement and leader election with full information. ACM Transactions on Algorithms **6**(4), 68:1–68:28 (2010). https://doi.org/10.1145/1824777.1824788
32. King, V., Lonargan, S., Saia, J., Trehan, A.: Load balanced scalable byzantine agreement through quorum building, with full information. In: Distributed Computing and Networking (ICDCN 2011). pp. 203–214 (2011). https://doi.org/10.1007/978-3-642-17679-1_18
33. King, V., Saia, J.: From almost everywhere to everywhere: Byzantine agreement with $\tilde{o}(n^{3/2})$ bits. In: DISC. Lecture Notes in Computer Science, vol. 5805, pp. 464–478. Springer (2009)
34. King, V., Saia, J.: Breaking the $O(n^2)$ bit barrier: Scalable byzantine agreement with an adaptive adversary. J. ACM **58**(4), 18:1–18:24 (2011)
35. King, V., Saia, J., Sanwalani, V., Vee, E.: Scalable leader election. In: SODA. pp. 990–999. ACM Press (2006)
36. Kumar, M., Molla, A.R.: Sublinear message bounds of authenticated implicit byzantine agreement. arXiv:2307.05922 (2023), https://arxiv.org/abs/2307.05922
37. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (Jul 1982). https://doi.org/10.1145/357172.357176

A Related Work

Lower bounds. Dolev and Reischuk [21] prove an $\Omega(n+t^2)$-message worst-case lower bound for deterministic Byzantine agreement. Abraham, Chan, Dolev, Nayak, Pass, Ren, and Shi [1] prove an expected $\Omega(t^2)$-message lower bound for randomized agreement against strongly adaptive corruptions with message takebacks when every honest party must terminate.

Setup assumptions also constrain subquadratic communication against adaptive adversaries without message takebacks. Blum, Katz, Liu-Zhang, and Loss [7] prove an expected $\Omega(t^2)$-message lower bound for setup-free asynchronous BA, even with secure channels and erasures, so some form of setup is necessary for (non-amortized) subquadratic communication when $t = \Theta(n)$. Ducros, Loss, and Rambaud [22] prove an expected $\Omega(t^2)$-message lower bound for synchronous Byzantine broadcast when setup is limited to a common reference string. They further prove an expected $\Omega(nt)$ lower bound on the total number of plain-message and signature transmissions for partially synchronous BA with a common reference string and idealized signatures, even in fault-free executions that are synchronous from the start and have identical inputs. The latter bound treats signatures as incompressible tokens, so it does not rule out subquadratic communication using threshold signatures as setup.

Almost-everywhere agreement. Almost-everywhere agreement was introduced by Dwork, Peleg, Pippenger, and Upfal [23] for fault tolerance in bounded-degree networks. Agreement and validity are guaranteed only for a large subset of honest parties, while the remaining honest parties may produce conflicting outputs. Chandran, Forghani, Garay, Ostrovsky, Patel, and Zikas [18] give a universally composable formulation of almost-everywhere secure computation, allowing unprivileged honest parties to lose correctness and privacy guarantees. In complete networks, King, Saia, Sanwalani, and Vee [35] develop synchronous almost-everywhere agreement and leader election with polylogarithmic per-party communication against static corruptions. Bhangale, Liu-Zhang, Loss, Nayak, and Yandamuri [6] repair gaps in the leader election analysis while retaining polylogarithmic per-party communication and rounds. A different relaxation preserves agreement for every honest decision but allows some honest parties to remain undecided. This distinction appears in implicit agreement as defined by Augustine, King, Molla, Pandurangan, and Saia [3] and in implicit Byzantine agreement as defined by Kumar and Molla [36], whose termination requirement guarantees only one honest decision maker.

Balanced synchronous agreement. A central approach to scalable full agreement is to first obtain almost-everywhere agreement and then propagate the result to the remaining parties. King and Saia [33] obtain synchronous full agreement with $\widetilde{O}(n^{3/2})$ total communication against static corruptions in the full-information model, and ask whether the cost can be balanced and improved using cryptography. King, Lonargan, Saia, and Trehan [32] achieve $\widetilde{O}(\sqrt{n})$ outgoing bits per party. Gelles and Komargodski [28] remove a superpolynomial local computation requirement in that construction and obtain computationally efficient protocols with $\widetilde{O}(\sqrt{n})$ bits sent and processed per party. Boyle, Goldwasser, and Tessaro [13] first obtain almost-everywhere certified agreement: almost all honest parties learn the value sent by the supreme committee, and certificates prevent any honest party from accepting an incorrect value. Their subsequent dissemination stage forwards the certified value to recipient sets derived from a certified seed for a pseudorandom function, so that every honest party receives the value with high probability. Together, these stages provide agreement and delivery to all honest parties in their synchronous model with static corruptions, yielding MPC with polylogarithmic communication locality from PKI and CRS setup. This is a direct precedent for our composition of almost-everywhere termination without disagreement and final dissemination of the certified decision. Locality counts communication partners rather than bits, and information identifying signers in their certificates can grow linearly with n. Boyle, Cohen, and Goel [10,11] overcome this cost using succinctly reconstructed distributed signatures, obtaining full BA with $\mathrm{poly}(\kappa, \log n)$ bits per party and polylogarithmic rounds. Their instantiations use one-way functions with trusted PKI, or collision-resistant hashing and SNARKs with linear extraction with bare PKI and a CRS. Fernando, Gelles, and Komargodski [27] obtain the same form of communication and round bounds from LWE, PKI and CRS setup using multi-hop batch arguments. These cryptographic constructions tolerate a constant gap below $n/3$ static online corruptions; their small per-party bounds include polynomial factors in the computational security parameter.

Adaptive security and hidden communication. For adaptive adversaries, hiding communication patterns provides a way to protect sparse communication from targeted corruptions. King and Saia [34] achieve

synchronous full agreement with $\widetilde{O}(\sqrt{n})$ bits sent per party against an adaptive adversary under a private-channel assumption that hides the fact that an honest party is sending a message to another honest party, not only message contents. This prevents the adversary from targeting a party for corruption upon observing that it was queried. In the hidden-graph model, Chandran, Chongchitmate, Garay, Goldwasser, Ostrovsky, and Zikas [17] obtain low-locality synchronous MPC using PKI, symmetric-key setup and atomic erasures. Chandran, Garay, Misra, Ostrovsky, and Zikas [19] subsequently construct low-locality communication with erasures but without atomic multi-send, and all-to-one reliable communication without either assumption using additional cryptography and hidden-graph setup. Boyle, Cohen, and Meyer [12] address reuse across executions, obtaining polylogarithmic locality against semi-honest corruptions and $\widetilde{O}(n^{2/3})$ locality against fail-stop corruptions. These results concern communication locality under hidden communication patterns, which is distinct from balanced bit complexity with publicly observable traffic.

Asynchronous agreement. With full information, Kapron, Kempe, King, Saia, and Sanwalani [31] achieve asynchronous full agreement against static corruptions with $\widetilde{O}(n^2)$ communication; their polylogarithmic-time variant has inverse-polylogarithmic failure probability. The almost-everywhere approach also has asynchronous extension procedures [33,32], which start from partial agreement on suitable committee information. In the cryptographic setting, classical threshold-based protocols achieve optimal resilience and expected constant rounds with quadratic communication [15]. Blum, Katz, Liu-Zhang, and Loss [7] obtain subquadratic asynchronous BA against adaptive corruptions for $t < (1-\delta)n/3$. Their single-shot protocol has an expected $O(1)$ number of iterations. For repeated executions, they refresh the setup using adaptively secure MPC, obtaining $n\,\mathrm{poly}(\kappa)$ expected communication per execution; the cited analysis does not state an end-to-end round bound for this refresh. Damgård, Kamp, Loss, and Nielsen [20] develop asynchronous YOSO MPC using threshold Paillier encryption, with subquadratic total order broadcast and linear amortized communication for sufficiently large batches. Asynchronous Algorand [2] obtains near-linear total communication and expected constant rounds for validated BA using a setup for verifiable random functions. These three constructions assume atomic multi-send, secure erasures and no message takebacks, and use small committees whose members may communicate with everyone. Feng, Mai, and Tang [25] obtain balanced asynchronous BA with agreement, validity, and termination for every honest party, except with negligible probability. Their adversary may corrupt parties adaptively during execution but cannot retract messages sent before corruption; the protocol requires neither atomic multi-send nor erasures. Their BA has expected total communication $O(\kappa n^{3/2}(\lambda+\log n))$, maximum outgoing per-party communication $O(\kappa\sqrt{n}(\lambda+\log n))$, and $O(\lambda)$ rounds [25]. The aggregation approach of Feng, Mai, and Tang is similar to ours: public samplers define overlapping communities robust to adaptively chosen corruptions, and intermediate BLS signatures are combined into a global signature. Their common coin gives full output in two deterministic asynchronous rounds with a constant corruption gap below $n/2$. They raise the possibility of reducing per-party communication through deeper recursive aggregation [25]. Their full version [26, Section 5.7] discusses the limitations of their explicit samplers for this purpose and suggests probabilistic sampler constructions as a possible way to overcome them. Their BA uses local multi-valued agreement within each community to select its next vote, followed by signature aggregation and flooding outputs to locally sampled random recipients. Our construction instead uses global support thresholds and binary sign-or-disseminate to implement justified proxcensus, accounting for missing inputs throughout the computation. The recursive hierarchy and forwarding analysis give explicit communication bounds for arbitrary constant depth and preserve almost-everywhere liveness under message takebacks. Table 1 summarizes these asynchronous results in comparison with ours.

B Compact Proxcensus Justifiers

The following compact justifier representation is adapted from [29, Appendix A]; we give a self-contained definition and proof for the representation used here.

Definition 11 (Two-value anchoring). *Call a justifier for a binary input* threshold-anchored *if it consists of a constant number of threshold certificates and coin certificates and contains no recursively nested grade justifier. We say that J_{IN} has the* two-value anchoring property *if, whenever both bits are J_{IN}-justified, there exists a J_{IN}-justified bit $b \in \{0,1\}$ whose associated justifiers are all threshold-anchored and have $O(\kappa)$ bits.*

All justification predicates accept only the canonical encodings specified below and reject redundant fields.

An *uncompressed recursive justifier* is a justifier π for an input bit b satisfying $J_{\text{IN}}(b, \pi) = \top$ at the two-grade base level, a threshold certificate for an even grade $2z$, or a pair of uncompressed justifiers for the lower-level grades $z-1$ and z for an odd grade $2z-1$. An uncompressed recursive justifier must pass the corresponding J_{IN} or threshold signature verification at each leaf of this recursive representation, together with the protocol instance, recursion level, and value checks in Figure 5.

The compact representation is defined recursively together with $J^{(s)}_{\text{OUT}}$. At the base level, a compact justifier for an input bit b is a string π satisfying $J_{\text{IN}}(b, \pi) = \top$. An even grade $2z$ is justified by the q_{uniq}-threshold certificate σ_z produced at that recursion level. For an odd grade $2z-1$, let e be the even member of $\{z-1, z\}$ and let o be the odd member. Above the base level, the compact proof contains the threshold certificate justifying e and follows the child of the decomposition of o other than $e/2$. The branch for $e/2$ is omitted because $q_{\mathsf{uniq}} > t$ and Corollary 1 imply that the existence of the threshold certificate for e entails the existence of an honest contributor that released an authorized share after obtaining the J_{OUT}-justified value $e/2$ together with an associated justifier. Let r be the retained child in the decomposition of o. If r is odd, then $x := e/2$ is even. Thus, the existentially supplied justifier for x is itself a threshold certificate. Applying Corollary 1 to this certificate supplies a justifier for $x/2$, the child shared by r and x; this branch is omitted and the verifier follows the other child of r.

This step is repeated until the retained child is even or the first recursive level is reached. The sequence of retained children is determined by the output grade and the recursion levels, so it need not be included in the justifier. If the retained child is even, its threshold certificate justifier is included as the terminal proof. At the first recursive level, the justifier for a $J^{(3)}_{\text{OUT}}$-justified odd grade contains associated justifiers for both J_{IN}-justified input bits. The operation $\mathsf{Compact}_s$ parses the two received child justifiers and produces this representation. The verifier checks the explicitly retained threshold certificates or input justifiers, together with their instance, recursion level, and value tags.

It does not test an existential condition and does not recover an omitted justifier or contributor. Lemma 7 proves that an accepted representation nevertheless has an existential expansion, except on the stated threshold signing failure event. Neither the honest contributors nor the omitted justifiers need to be recovered from the threshold certificates.

Lemma 7 (Compact proxcensus justifiers). *Assume $q_{\mathsf{uniq}} > t$ and that the corresponding threshold signing configurations satisfy the conclusion of Corollary 1 simultaneously for all tagged messages. For every input justification predicate J_{IN} and every $G = 2^k + 1$, compacting an uncompressed recursive justifier for a grade y yields a representation accepted by $J^{(G)}_{\text{OUT}}$ for y. Conversely, conditioned on the conclusion of Corollary 1, every $J^{(G)}_{\text{OUT}}$-justified grade y has an uncompressed recursive justifier obtained by existentially expanding any associated compact justifier.*

Suppose additionally that, for every J_{IN}-justified input bit b, its associated justifiers have at most B bits, that J_{IN} has the two-value anchoring property of Definition 11, and that a threshold certificate has $O(\kappa)$ bits. Then, for every $J^{(G)}_{\text{OUT}}$-justified grade y, its associated compact justifiers have length $B + O(\kappa)$, where the hidden constant is independent of k. At most k applications of Corollary 1 occur on any descent path. In the BA instantiation $G = 9$ and hence $k = 3$.

Proof. We prove simultaneously, by strong induction on the recursion depth, that compacting an uncompressed recursive justifier yields an accepted compact representation and that every $J^{(G)}_{\text{OUT}}$-justified grade y has an uncompressed recursive justifier obtained by existentially expanding any associated compact justifier. Under the additional hypotheses for the length bound, we show that, for every $J^{(G)}_{\text{OUT}}$-justified grade y, each associated compact justifier contains at most one input justifier of length at most B and two constant-size certificate components.

For completeness, an uncompressed justifier for an even grade already consists of one threshold certificate. At the first recursive level, an uncompressed justifier for an odd grade consists of strings π_0 and π_1 satisfying $J_{\text{IN}}(0, \pi_0) = J_{\text{IN}}(1, \pi_1) = \top$. Now consider an odd grade above the first recursive level. Its two child grades consist of an even grade $e = 2x$ and an odd grade o. The uncompressed proof contains a threshold certificate σ_x justifying e, while the proof of o decomposes one level lower into the shared child x and one further child. The compact proof retains σ_x, deletes the explicit proof of the shared child from the proof of o, and follows

the further child. If that child is again odd, its shared child is even; the compact proof omits that shared child's threshold certificate and again follows only the other child. Thus, only the first threshold certificate is retained along this descent. The recursion level strictly decreases, so the path ends either at an even child, whose threshold certificate is included as terminal data, or at a first-level odd child, whose two input justifiers are included. Hence every uncompressed proof can be pruned to an accepted compact proof.

We prove soundness by induction on the recursion depth k. For $k = 1$, a $J^{(3)}_{\mathrm{OUT}}$-justified even grade has an associated justifier consisting of its threshold certificate, and a $J^{(3)}_{\mathrm{OUT}}$-justified odd grade has an associated justifier containing justifiers for both J_{IN}-justified input bits, so the claim is immediate. Assume the claim for every smaller recursion depth and fix a $J^{(G)}_{\mathrm{OUT}}$-justified grade y at depth $k > 1$ together with an associated compact justifier. An even grade is again immediate. Let the grade therefore be odd. Its two child grades are an even grade $e = 2x$ and an odd grade o, and the compact proof explicitly contains a valid certificate σ_x justifying e.

Since $q_{\mathsf{uniq}} > t$, Corollary 1 implies, except on its unforgeability failure event, that an honest authorized contributor to σ_x possessed a lower-level *compact* justifier π_x for x. By the induction hypothesis, π_x has an uncompressed expansion; no such expansion is assumed to be encoded in or extractable from the certificate σ_x.

It remains to expand the compact tail for o. The lower-level decomposition of o has the shared child x and one further child r. If r is even, the compact tail explicitly contains its threshold certificate justifier; combining it with the uncompressed expansion of π_x gives an uncompressed justifier for o. If r is odd, then x is even, so π_x itself is a valid threshold certificate justifying x. Applying Corollary 1 at the first production time of this certificate implies the existence of a compact justifier for the next omitted shared child, and the induction hypothesis supplies its uncompressed expansion. Repeating this step strictly decreases the recursion depth. The descent ends either with an even retained child, whose certificate is explicitly present, or at the first recursive level, where associated justifiers for both J_{IN}-justified input bits are explicitly present. Combining the resulting uncompressed branch justifiers from the bottom upward first reconstructs an uncompressed justifier for o and then, together with σ_x, one for the original odd grade.

All signing instance, recursion level, and value tags are checked, so every existentially supplied compact justifier is for the unique shared child at that point of the path. Neither that compact justifier, its uncompressed expansion, nor the honest contributor needs to be extracted by the verifier. The initial application of Corollary 1 and the induction hypotheses require at most k applications of that corollary along any descent path.

It remains to bound the representation length. An even compact justifier contains one threshold certificate. For a $J^{(3)}_{\mathrm{OUT}}$-justified odd grade, an associated compact justifier contains associated justifiers for both J_{IN}-justified input bits; by Definition 11, one has $O(\kappa)$ bits and the other has at most B bits. Above the first recursive level, an odd compact justifier contains one initial threshold certificate and either one terminal threshold certificate or the same terminal pair of justifiers for the input bits. No intermediate certificate is retained. Its length is therefore $B + O(\kappa)$, with a constant independent of k. □

C Proofs for Recursive Threshold Signing

Lemma 1 (Recursive sampler for safety and liveness thresholds). *For the hierarchy sampled in Section 4.1, with $m_* < n$ and constant recursion depth L, sufficiently large choices of ρ, c_*, c_f, and c_{occ} make the hierarchy (q, h, t, η)-good in the sense of Definition 6, except with probability $\exp(-\lambda)$. The root liveness and robust forwarding conclusions also hold for every disjoint pair (C, A) with $|A| \geq h$, without restricting $|C|$. On the same event, every child u of every internal bin v satisfies*

$$\frac{1}{2}p|Q_v| \leq |Q_u| \leq \frac{3}{2}p|Q_v|.$$

Write $\mathcal{E}_{\mathsf{samp}}$ for this joint event. The constants depend only on s, ℓ, η, and L.

Proof. **Committee sizes.** Let $\mu_j := np^j$ be the nominal size of a level-j committee, and let $\mathcal{E}_{\mathsf{size}}$ be the event that every child u of every internal bin v satisfies

$$\frac{1}{2}p|Q_v| \leq |Q_u| \leq \frac{3}{2}p|Q_v|.$$

Put $x := (n/m_*)^{1/L} > 1$. Since

$$\rho x \le b \le \rho x + 1 \le (\rho+1)x,$$

we have

$$\frac{\rho}{\rho+1}x^{-1} \le p = \frac{\rho}{b} \le x^{-1}.$$

Consequently,

$$\left(\frac{\rho}{\rho+1}\right)^L m_* \le \mu_L = np^L \le m_*,$$

so the nominal leaf size is $\Theta(m_*)$. Moreover, $b^L = O(n/m_*)$, and hence the constant-depth hierarchy has $O(b^L) = O(n/m_*) = O(n)$ bins and edges. Reveal the hierarchy top-down and apply a multiplicative Chernoff bound to the first edge violating the displayed inequalities. Before such a violation, every child has conditional expected size $\Omega(m_*)$. Increasing c_* therefore gives $\Pr[\neg\mathcal{E}_{\mathsf{size}}] \le \exp(-\lambda)/4$, and every internal committee has size $\Omega(m_*)$ on $\mathcal{E}_{\mathsf{size}}$. For every internal bin, the size bounds along its ancestor path also give

$$\frac{|Q_v|}{b} \ge \frac{2^{-(L-1)}np^{L-1}}{b} = \frac{2^{-(L-1)}\mu_L}{\rho} = \Omega\left(\frac{m_*}{\rho}\right).$$

Choose c_* sufficiently large that this lower bound is at least one, so $b \le |Q_v|$ whenever the ancestor path satisfies the size bounds.

Safety and liveness sampling. We bound the probability that a density implication fails while all relevant size bounds hold, without conditioning on the global event $\mathcal{E}_{\mathsf{size}}$. Reveal a level-$j$ parent committee Q_v along an ancestor path satisfying the size bounds, and then condition on an arbitrary vector of child sizes in $[\frac{1}{2}p|Q_v|, \frac{3}{2}p|Q_v|]$. Under this local conditioning, the children are independent uniform subsets of Q_v of their respective prescribed sizes. Hypergeometric concentration and the union bound below give the following implications simultaneously for every internal bin v and every $S \subseteq Q_v$:

$$|S| \le d_j^-|Q_v| \Longrightarrow \left|\left\{u \in \mathrm{child}(v) : |S \cap Q_u| > d_{j+1}^-|Q_u|\right\}\right| < k_v, \tag{3}$$

$$|S| \ge d_j^+|Q_v| \Longrightarrow \left|\left\{u \in \mathrm{child}(v) : |S \cap Q_u| \ge d_{j+1}^+|Q_u|\right\}\right| \ge (\phi + 2\alpha)b. \tag{4}$$

For fixed S satisfying the relevant premise, a child that violates the required density has probability at most $\exp(-c\zeta^2 p|Q_v|)$ for an absolute constant $c > 0$. For a prescribed set of deviating children, multiply these bounds using conditional independence. Failure of the first implication requires at least ϕb deviating children, and failure of the second requires at least $(1-\phi-2\alpha)b$ deviating children. Writing $r := \min\{\phi, 1-\phi-2\alpha\}$, a union bound over at most $2^{|Q_v|}$ choices of S and 2^b choices of deviating children bounds the failure probability by

$$2^{|Q_v|+b}\exp(-c\zeta^2 r\rho|Q_v|).$$

Since $b \le |Q_v|$, this is $\exp(-\Omega(|Q_v|))$ once

$$\rho = \Omega\left(\frac{1}{\zeta^2 \min\{\phi, 1-\phi-2\alpha\}}\right)$$

with a sufficiently large multiplicative constant. The estimate is uniform over every child size vector satisfying the displayed size bounds, so averaging over those vectors bounds the probability that a density implication fails while these bounds hold. Applying the bound to every parent whose ancestor path satisfies the size bounds and union bounding over the $O(n)$ bins gives probability at most $\exp(-\lambda)/4$ that a density implication fails on $\mathcal{E}_{\mathsf{size}}$, after increasing c_*.

Root safety. If a level-j bin is S-open, then (3) and induction from the leaves imply $|S \cap Q_v| > d_j^-|Q_v|$. If the root were S-open, then $|S| > d_0^- n = q - 1$, contradicting $|S| < q$.

Root liveness. Equation (4) and induction from the leaves imply that every level-j bin satisfying $|A \cap Q_v| \ge d_j^+|Q_v|$ is A-live. The condition $|A| \ge h = d_0^+ n$ therefore makes the root live.

Robust forwarding. Condition on any fixed committee hierarchy satisfying $\mathcal{E}_{\mathsf{size}}$; the forwarding tuples are sampled independently of this hierarchy. Fix arbitrary disjoint sets $C, A \subseteq [n]$ with $|A| \ge h$. Every A-live

child u satisfies $|A \cap Q_u| \geq (\phi + 4\alpha)|Q_u|$, and $A \subseteq [n] \setminus C$. Hence $|Q_u \setminus C| \geq (\phi + 4\alpha)|Q_u|$. Every set $S \subseteq Q_u \setminus C$ satisfying $|S| \geq (1-\eta)|Q_u \setminus C|$ therefore also satisfies $|S| \geq \theta|Q_u|$. Fix an edge $u \to v$, a sender set $S \subseteq Q_u$ of size at least $\theta|Q_u|$, and a recipient set $B \subseteq Q_v$ of size at least $\beta|Q_v|$. The probability that every one of the $f|S|$ recipient samples avoids B is at most $\exp(-\beta f\theta|Q_u|)$. Union bounding over S and B contributes at most $2^{|Q_u|+|Q_v|}$. On $\mathcal{E}_{\mathsf{size}}$, $b|Q_u| \geq (\rho/2)|Q_v|$ and $|Q_u| \leq (3/2)|Q_v|$, so $f \geq c_f b$ makes this failure probability $\exp(-\Omega(|Q_v|))$ for sufficiently large constant c_f. Since every internal committee has size $\Omega(\lambda + \log n)$, increasing c_* makes one union bound over all edges at most $\exp(-\lambda)/4$. Consequently, on every edge $u \to v$, every sender set $S \subseteq Q_u$ with $|S| \geq \theta|Q_u|$ reaches all but a β-fraction of Q_v. There are at least $(\phi + 2\alpha)b$ live children, whereas reconstruction requires only $\lfloor \phi b \rfloor + 1$ child signatures. A parent party that receives fewer than $\lfloor \phi b \rfloor + 1$ child signatures receives signatures from at most $\lfloor \phi b \rfloor$ live children and therefore misses at least

$$(\phi + 2\alpha)b - \lfloor \phi b \rfloor \geq 2\alpha b \geq \alpha b$$

live children. Double counting missed pairs of a child and a recipient bounds the number of such parties by

$$\frac{\beta}{\alpha}|Q_v| = \frac{\eta(\phi + 4\alpha)}{2}|Q_v| \leq \eta|Q_v \setminus C|,$$

where the last inequality follows from $|Q_v \setminus C| \geq |A \cap Q_v| \geq (\phi + 4\alpha)|Q_v|$ for the live parent.

Committee load. To prove the bound on the number of committees containing each party in Definition 6, fix a party occurrence in a level-j committee. The number of child committees containing it is distributed as $\mathrm{Bin}(b, p)$ and has expectation ρ. A Chernoff bound shows that it exceeds $c_{\mathsf{occ}}\bar{\lambda}$ with probability at most $\exp(-\Omega(c_{\mathsf{occ}}\bar{\lambda}\log\bar{\lambda}))$. There are at most $n(1+b+\cdots+b^L) = O(n^2)$ possible pairs of a party and a committee. By the choice of c_{occ}, a union bound therefore shows, except with probability at most $\exp(-\lambda)/4$, that every party occurrence has at most $c_{\mathsf{occ}}\bar{\lambda}$ child occurrences. Induction from the root implies that every party belongs to at most $(c_{\mathsf{occ}}\bar{\lambda})^j = O(\bar{\lambda}^j)$ level-j committees. □

Lemma 2 (Access structure for safety and liveness thresholds). *For every bin v and party set $S \subseteq [n]$, the exponent x_v is a public linear combination of the scalar leaf shares held by S below v if and only if v is S-open.*

Proof. We prove the claim by induction on the height of v, together with the following stronger statement for every bin that is not S-open: for every prescribed $\Delta \in \mathbb{Z}_{\mathsf{p}}$, its secret and the sharing randomness below it can be changed so that its secret increases by Δ while every scalar leaf share held by S below it remains unchanged. At a leaf, any k_v shares interpolate $F_v^{\mathsf{leaf}}(0) = x_v$. If S holds fewer than k_v shares, then for every prescribed $\Delta \in \mathbb{Z}_{\mathsf{p}}$ the polynomial

$$G_v(z) := \Delta \prod_{P_i \in S \cap Q_v} \left(1 - \frac{z}{\xi_{v,i}}\right)$$

has degree at most $k_v - 1$, changes the constant coefficient by Δ, and leaves every share held by S unchanged. For an internal bin, let $\mathcal{O}_v$ be its S-open children. If $|\mathcal{O}_v| \geq k_v$, the induction hypothesis supplies their exponents and interpolation recovers $F_v(0) = x_v$. If $|\mathcal{O}_v| < k_v$, adding

$$G_v(z) := \Delta \prod_{u \in \mathcal{O}_v} \left(1 - \frac{z}{\xi_u}\right)$$

changes x_v by the prescribed Δ without changing an open child exponent. For each child u that is not S-open, the strengthened induction hypothesis realizes exactly the shift $G_v(\xi_u)$ in that subtree while leaving all shares held by S fixed. These changes are compatible because distinct child subtrees have separate sharing randomness, even when a party belongs to more than one subtree. This proves the stronger statement and shows that x_v is not determined by those shares when v is not S-open. □

Theorem 1 (Recursive threshold signing). *Under the setup and computational assumptions of this section, fix parameters satisfying $t < q < h = q + \lceil an \rceil \leq n - t$, a constant $L \geq 1$, and $\lambda = \Omega(\kappa + \log n)$. There exists a protocol satisfying Definition 5 against strongly adaptive corruptions with message takebacks,*

except with probability $\exp(-\lambda) + \mathrm{negl}(\kappa)$*. For input justifiers of at most* B *bits, all required outputs occur within* $L+1$ *asynchronous time units after* h *honest input contributions. On the setup event* $\mathcal{E}_{\mathsf{samp}}$*, defined in the proof and having probability at least* $1-\exp(-\lambda)$*, the total outgoing communication before corruption, in bits, is*

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right](\kappa + B)\right).$$

The expected maximum outgoing communication per party before corruption, in bits and without conditioning on $\mathcal{E}_{\mathsf{samp}}$*, is*

$$\widetilde{O}\left(\left[\lambda + \left(\frac{n}{\lambda}\right)^{1/L}\right](\kappa + B)\right).$$

Proof. Use the signing construction and parameter choices of Section 4. Let $\mathcal{E}_{\mathsf{samp}}$ be the event in Lemma 1 when $m_* < n$, and let it hold unconditionally for the direct construction. We prove each of the required properties.

Uniqueness and liveness. The uniqueness of BLS signatures gives uniqueness. For the recursive construction, condition on $\mathcal{E}_{\mathsf{samp}}$ and let A be the set of honest contributors to the instance. If $|A| \geq h = q + \lceil an \rceil$, then Lemma 1 makes the root A-live. Moreover, every leaf committee is nonempty on $\mathcal{E}_{\mathsf{samp}}$. Hence, for every A-live leaf v,

$$|A \cap Q_v| \geq d_L^+ |Q_v| = (\phi + 4\alpha)|Q_v| > \phi |Q_v|.$$

Since $|A \cap Q_v|$ is an integer,

$$|A \cap Q_v| \geq \lfloor \phi |Q_v| \rfloor + 1 = k_v.$$

Thus the honest contributors alone send enough distinct valid leaf partial signatures for every honest member of each live leaf to reconstruct. Their partials carry justifiers π satisfying $J_{\text{IN}}(m, \pi) = \top$, so recipients can reconstruct and forward regardless of their own input. Proceed upward by induction. Let C be the final corruption set. For every A-live child u, let $S_u \subseteq Q_u \setminus C$ be the honest members that have reconstructed σ_u. At a live leaf, the argument above gives $S_u = Q_u \setminus C$. Inductively, suppose that

$$|S_u| \geq (1-\eta)|Q_u \setminus C|$$

for every $u \in \mathcal{L}_v$, where v is an A-live internal bin. The robust forwarding property of Definition 6 implies that all but an η-fraction of $Q_v \setminus C$ receive valid signatures from at least k_v distinct A-live children. Every such party interpolates σ_v and retains a justifier π for m satisfying $J_{\text{IN}}(m, \pi) = \top$, so at least $(1-\eta)|Q_v \setminus C|$ honest members of Q_v reconstruct σ_v. Applying the argument at the root $Q_r = [n]$ leaves at most $\eta|[n] \setminus C| \leq \eta n$ honest parties without an output and hence, by integrality, at most $\lfloor \eta n \rfloor$.

For the direct construction used when $m_* \geq n$, at least $h \geq q$ honest contributors send distinct valid shares to every honest party, so every honest party reconstructs directly.

Unforgeability. We prove the following stronger statement. Fix a tagged message $\mu = (\mathsf{sid}, m)$, and let T be the first execution event at which a valid root signature for μ is output by a protocol party or presented by the adversary as a purported certificate. Let $C(T)$ be the identities corrupted by time T, and let $I_\mu(T)$ be the identities uncorrupted at T that have released at least one of their own authorized leaf signature shares for μ by time T. The set $I_\mu(T)$ is defined from the execution history and is used only in the security analysis. It is not encoded in the root signature, and the statement does not require an algorithm that extracts contributing identities from a valid signature. Moreover, membership in $I_\mu(T)$ only records that the party released an authorized share for μ; it does not assert that this particular share was used to construct the root signature. An identity is counted only once in $I_\mu(T)$, even if it occurs in several leaf committees. Except with probability $\exp(-\lambda) + \mathrm{negl}(\kappa)$,

$$\left|C(T) \cup I_\mu(T)\right| \geq q. \tag{5}$$

Let $\mathsf{Bad}_{\mathsf{uf}}$ be the event that this inequality fails for some μ and T at which a valid root signature is produced.

We give the complete reduction. Let $\mathbb{F} := \mathbb{Z}_{\mathfrak{p}}$. The independent scalar randomness of the recursive sharing consists of the root exponent and the independently sampled nonconstant coefficients of every sharing polynomial at an internal bin or leaf. Enumerate these scalars as

$$\mathbf{z} = (z_0, \ldots, z_{D_0 - 1}) \in \mathbb{F}^{D_0}, \qquad z_0 = x_r.$$

Fix a deterministic polynomial upper bound $D \geq \max\{2, D_0\}$ and pad $\mathbf{z}$ with $D - D_0$ unused independent coordinates. Every bin exponent and every leaf share is a publicly known linear form in $\mathbf{z}$. Accordingly, for every bin v and every occurrence of party P_i in a leaf v, there are efficiently computable vectors $\ell_v, \ell_{v,i} \in \mathbb{F}^D$ such that

$$x_v = \langle \ell_v, \mathbf{z} \rangle, \qquad x_{v,i} = \langle \ell_{v,i}, \mathbf{z} \rangle, \qquad \ell_r = (1, 0, \ldots, 0). \tag{6}$$

For example, if $F_v(X) = x_v + \sum_{d=1}^{k_v - 1} a_{v,d} X^d$ and the coordinate vector $e_{v,d}$ corresponds to $a_{v,d}$, then a child u of v has

$$\ell_u = \ell_v + \sum_{d=1}^{k_v - 1} \xi_u^d e_{v,d}.$$

The analogous formula holds for a leaf share, using the nonconstant coefficients of F_v^{leaf}.

For an identity i, let $\mathcal{L}_i := \{\ell_{v,i} : v \text{ is a leaf and } P_i \in Q_v\}$, and, for $S \subseteq [n]$, define

$$W_S := \operatorname{span}_{\mathbb{F}} \left(\bigcup_{i \in S} \mathcal{L}_i \right).$$

In terms of coefficient vectors, Lemma 2 states that $\ell_r \in W_S$ if and only if the root is S-open. Therefore, on $\mathcal{E}_{\mathsf{samp}}$, the root safety property of Definition 6, guaranteed by Lemma 1, gives

$$|S| < q \quad \Longrightarrow \quad \ell_r \notin W_S. \tag{7}$$

Suppose that the event in which (5) fails has nonnegligible probability. We construct an AOMCDH adversary $\mathcal{B}$. Let Q_{BLS} be a deterministic polynomial upper bound on the number of distinct tagged messages evaluated under H_{BLS} during the experiment, including queries made by the adversary, evaluations by simulated honest parties, and evaluations needed to verify purported signatures. Algorithm $\mathcal{B}$ samples $J \xleftarrow{\$} [Q_{\mathsf{BLS}}]$ before the experiment and designates the J-th distinct H_{BLS} query as its target μ^*; it aborts if fewer than J distinct queries occur.

On every execution of $\mathsf{Bad}_{\mathsf{uf}}$, let μ_{bad} be the tagged message associated with the chronologically first event witnessing a failure of (5), and let J_{bad} be the index of its first H_{BLS} evaluation. Verification of the corresponding signature ensures that this evaluation occurs. Since both the AOMCDH challenge H and the programmed values g^{α_μ} are uniform elements of $\mathbb{G}$, the target designation does not alter the random oracle transcript. Consequently,

$$\Pr[J = J_{\mathsf{bad}}] = \frac{1}{Q_{\mathsf{BLS}}}.$$

Algorithm $\mathcal{B}$ invokes the AOMCDH game with $N := D - 1$ and receives $\big(\Gamma, H, Z_0, \ldots, Z_{D-1}\big)$, $Z_j = g^{z_j}$, together with access to the scalar oracle

$$\mathsf{DLog}_{\mathbf{z}}(a_0, \ldots, a_{D-1}) := \sum_{j=0}^{D-1} a_j z_j.$$

It uses the challenge exponents as the independent sharing randomness. Equation (6) lets it compute every published verification key without knowing any challenge exponent; namely,

$$X_v = \prod_{j=0}^{D-1} Z_j^{(\ell_v)_j}, \qquad X_{v,i} = \prod_{j=0}^{D-1} Z_j^{(\ell_{v,i})_j}.$$

It samples the hierarchy and all other setup randomness honestly. Thus the public setup has exactly the distribution of the real protocol. In particular, the reduction neither conditions on nor tests $\mathcal{E}_{\mathsf{samp}}$; it samples the hierarchy unconditionally and later analyzes its success on the joint event $\mathsf{Bad}_{\mathsf{uf}} \wedge \mathcal{E}_{\mathsf{samp}}$.

The reduction programs the guessed random oracle query as $H_{\mathsf{BLS}}(\mu^*) := H$. For every other fresh query μ, it samples $\alpha_\mu \leftarrow_{\$} \mathbb{F}$ and sets $H_{\mathsf{BLS}}(\mu) := g^{\alpha_\mu}$. These answers have the exact random oracle distribution. Moreover, for a nontarget message and any published key $X = g^x$, the reduction computes the required signature as $H_{\mathsf{BLS}}(\mu)^x = X^{\alpha_\mu}$, without learning x.

Beginning at setup, algorithm $\mathcal{B}$ maintains a basis $U = (u_1, \ldots, u_R)$ of all leaf share vectors whose scalar values the simulation has obtained from its oracle, together with the returned values $\langle u_j, \mathbf{z}\rangle$. Whenever a party is corrupted, whether before or after the target query is identified, the reduction processes every leaf share vector ℓ held by that party. Once μ^* has been identified, it processes in the same way every vector ℓ for which an uncorrupted party must release a leaf signature share for μ^*.

If $\ell \in \mathrm{span}(U)$, the reduction expresses ℓ as a public linear combination of the current basis vectors and applies the same combination to the stored scalar values. Otherwise, it queries $\mathsf{DLog}_{\mathbf{z}}(\ell)$ and appends ℓ and the returned scalar to U. It can therefore reveal the scalar share upon corruption or compute $H^{\langle \ell, \mathbf{z}\rangle}$ when a partial signature for μ^* is required. No scalar oracle query is needed merely to sign a nontarget message: if $H_{\mathsf{BLS}}(\mu) = g^{\alpha_\mu}$ and $X = g^x$ is a published key, the corresponding signature is X^{α_μ}.

Repeated and linearly dependent shares require no additional query. The reduction aborts if adjoining a vector would make the rank equal to D. This cannot occur on $\mathsf{Bad}_{\mathsf{uf}} \cap \mathcal{E}_{\mathsf{samp}}$ before the witnessing time T: every queried vector belongs to $W_{C(T)\cup I_{\mu^*}(T)}$, whereas Equation (7) places ℓ_r outside this space. Hence the queried span has dimension at most $D-1$.

All other protocol operations are simulated without scalar oracle queries. Pairing checks use only public information, while leaf and internal bin reconstruction consists of applying public interpolation coefficients to available group elements. Thus the simulation is perfect up to an abort and handles adaptive corruptions, scheduling, and message takebacks without guessing the corruption set.

The reduction also monitors every valid leaf, bin, or root signature for μ^* that is supplied to an honest party or produced as an output. Such a signature has a publicly known coefficient vector w, equal respectively to some $\ell_{v,i}$, ℓ_v, or ℓ_r. If its group element Y passes the corresponding pairing check and $w \notin \mathrm{span}(U)$, then $Y = H^{\langle w, \mathbf{z}\rangle}$. Indeed, the verification equation says

$$e(Y, g) = e\Big(H, g^{\langle w, \mathbf{z}\rangle}\Big) = e\Big(H^{\langle w, \mathbf{z}\rangle}, g\Big),$$

and nondegeneracy in the prime-order groups makes the map $Y \mapsto e(Y, g)$ injective.

At this point $\mathcal{B}$ completes $u_1, \ldots, u_R, w$ to a basis $u_1, \ldots, u_R, w, u_{R+2}, \ldots, u_D$ of $\mathbb{F}^D$ and queries $\mathsf{DLog}_{\mathbf{z}}$ on the $D-R-1$ new basis vectors other than w. Its total number of scalar oracle queries is exactly $R + (D-R-1) = D-1 = N$. For every queried basis vector u, the returned scalar allows it to compute $H^{\langle u, \mathbf{z}\rangle}$; the equation $Y = H^{\langle w, \mathbf{z}\rangle}$ supplies the sole unqueried basis encoding. Inverting the public change-of-basis matrix then lets $\mathcal{B}$ compute $H^{z_0}, \ldots, H^{z_{D-1}}$, and hence win the AOMCDH game. Concretely, denote the displayed basis $u_1, \ldots, u_R, w, u_{R+2}, \ldots, u_D$ by $b_1, \ldots, b_D$, so that $b_{R+1} = w$. For every $j \in \{0, \ldots, D-1\}$, let e_j be the standard basis vector corresponding to the coordinate z_j. Since $b_1, \ldots, b_D$ form a public basis of $\mathbb{F}^D$, the reduction computes coefficients $\gamma_{j,1}, \ldots, \gamma_{j,D} \in \mathbb{F}$ satisfying

$$e_j = \sum_{k=1}^{D} \gamma_{j,k} b_k.$$

It then computes

$$H^{z_j} = H^{\langle e_j, \mathbf{z}\rangle} = \prod_{k=1}^{D} \left(H^{\langle b_k, \mathbf{z}\rangle}\right)^{\gamma_{j,k}}.$$

For $k \neq R+1$, the corresponding factor is obtained from a discrete logarithm oracle answer; the remaining factor is $Y = H^{\langle w, \mathbf{z}\rangle}$ from $Y = H^{\langle w, \mathbf{z}\rangle}$.

It remains to show that a violation for the guessed target gives the reduction a signature whose coefficient vector lies outside $\mathrm{span}(U)$. Immediately before the target signature is produced at time T, every vector queried by the reduction belongs to $W_{C(T)\cup I_{\mu^*}(T)}$; hence $\mathrm{span}(U) \subseteq W_{C(T)\cup I_{\mu^*}(T)}$. Indeed, every identity whose scalar leaf shares were revealed belongs to $C(T)$. Every identity for which the reduction computed a leaf signature for μ^* before corruption either remains uncorrupted at time T and therefore belongs to $I_{\mu^*}(T)$, or was subsequently corrupted and therefore belongs to $C(T)$. If $|C(T) \cup I_{\mu^*}(T)| < q$, then Equation (7) gives $\ell_r \notin W_{C(T)\cup I_{\mu^*}(T)}$ and hence $\ell_r \notin \mathrm{span}(U)$. Consequently, either the reduction has already stopped at an earlier valid signature for μ^* whose coefficient vector lies outside $\mathrm{span}(U)$, or the valid root signature itself supplies the missing encoding for $w = \ell_r$. In either case $\mathcal{B}$ solves AOMCDH. The target guess loses only the polynomial factor Q_{BLS}, and therefore

$$\Pr[\mathsf{Bad}_{\mathsf{uf}} \wedge \mathcal{E}_{\mathsf{samp}}] \leq Q_{\mathsf{BLS}} \cdot \mathrm{Adv}^{\mathsf{AOMCDH}}_{\mathcal{B}, D-1}(\kappa) = \mathrm{negl}(\kappa).$$

Since $\Pr[\neg\mathcal{E}_{\mathsf{samp}}] \le \exp(-\lambda)$, this proves (5) with the claimed failure probability.

Finally, suppose a valid signature is produced at time T. Write $c := |C(T)|$. Since $C(T) \cap I_\mu(T) = \emptyset$, Equation (5) gives $|I_\mu(T)| \ge q - c$. At most $t - c$ of these currently uncorrupted parties can be corrupted later, so at least $(q-c) - (t-c) = q - t$ of them remain honest throughout the execution. By the protocol rule, every one of these parties released at least one of its own authorized leaf signature shares only after contributing input (m, π) satisfying $J_{\text{IN}}(m, \pi) = \top$ to the instance. This proves the unforgeability requirement in Definition 5.

For the direct construction used when $m_* \ge n$, the same reduction applies with the coefficient vectors of an ordinary q-out-of-n Shamir sharing. Its access property gives $\ell_r \notin W_S$ deterministically whenever $|S| < q$, so the rest of the argument is unchanged.

Define

$$d_{\mathsf{load}} := \max\left\{2, \left\lceil \frac{\log n}{\log\log(n+2)} \right\rceil\right\}.$$

Let $X(\mathcal{G})$ be the maximum, over all parties, of the total bit length of all partial signature sends and all non-root forwarding sends prescribed in one instance. Since $X(\mathcal{G})$ depends only on the public transmission structure $\mathcal{G}$, it bounds every instance regardless of the set of contributors or the message schedule.

The following communication bounds assume input justifiers of $O(\kappa)$ bits; for justifiers of length at most B, multiply the bounds by $1 + B/\kappa$. Conditioned on $\mathcal{E}_{\mathsf{samp}}$, if $\lambda = \Omega(\log n)$, the total online outgoing communication is $O\big(\big[n\lambda + n(n/\lambda)^{1/L}\big]\,\kappa\big)$, where the hidden constant depends on the normalized safety and liveness thresholds, their gap, η, and L. Moreover, if $m_* < n$, every party sends at most

$$O\left((\lambda + \log n)^L \left[\lambda + \log n + \left(\frac{n}{\lambda + \log n}\right)^{1/L}\right]\kappa\right)$$

bits.

Total communication. Let B bound the bit length of an accepted input justifier. Each prescribed transmission carries one signature and one input justifier, for $O(\kappa + B)$ bits; we first give the bounds for $B = O(\kappa)$. In the direct construction, at most n contributors each send to all n parties, for total communication $O(n^2\kappa)$ and the deterministic bound $X(\mathcal{G}) = O(n\kappa)$. Since $n \le m_* = O(\lambda + \log n)$ and $\lambda = \Omega(\log n)$, these bounds are $O(n\lambda\kappa)$ and $O(\lambda\kappa)$, respectively, satisfying the claimed total communication and both moment bounds. When $m_* < n$, on $\mathcal{E}_{\mathsf{samp}}$, the total number of party occurrences at recursive level j is $O(n\rho^j)$ and every leaf has size $O(m_*)$. Leaf transmission therefore costs $O(n\rho^L m_*\kappa) = O(n\bar{\lambda}\kappa)$ bits. Forwarding costs

$$O\left(nf\sum_{j=1}^{L}\rho^j\kappa\right) = O\left(n\left(\frac{n}{\bar{\lambda}}\right)^{1/L}\kappa\right)$$

bits because $f = \Theta(b)$ and L and ρ are constant.

Maximum outgoing communication. For every possible party occurrence in an internal bin, let Y be the number of child occurrences that it generates. The random variable Y is binomial with mean $bp = \rho$, and hence

$$\Pr[Y \ge x] \le \left(\frac{e\rho}{x}\right)^x$$

for every $x \ge e\rho$. There are at most $O(n^2)$ possible party occurrences in the constant-depth hierarchy. Let Δ_{ch} be the maximum number of children generated by any possible party occurrence. A union bound and summation of the displayed tail give, for every fixed positive integer r, $\mathbb{E}[\Delta_{\mathsf{ch}}^r] = O(d_{\mathsf{load}}^r)$. Every party belongs to at most Δ_{ch}^j level-j committees. On $\mathcal{E}_{\mathsf{samp}}$, each leaf occurrence sends its member partial to $O(m_*)$ recipients and its reconstructed leaf bin signature to at most f recipients, for $O(m_* + f)$ recipient transmissions in total. Each internal non-root occurrence sends to at most $f = O((n/m_*)^{1/L})$ recipients. Thus the maximum prescribed outgoing communication satisfies $X(\mathcal{G}) = O(\Delta_{\mathsf{ch}}^L(m_* + f)\kappa)$ on this event. There are $O(n)$ bins even outside $\mathcal{E}_{\mathsf{samp}}$, and executing each indexed send once bounds each party's outgoing communication by $O(n^2\kappa)$ on every setup. Consequently, the contribution of $\neg\mathcal{E}_{\mathsf{samp}}$ to each of the first two

moments is negligible because n is polynomially bounded in κ and $\lambda = \Omega(\kappa + \log n)$. Applying this moment bound with $r = L$ and $r = 2L$ therefore gives

$$\mathbb{E}[X(\mathcal{G})] \le \left(\mathbb{E}[X(\mathcal{G})^2]\right)^{1/2} = O\left(d_{\mathsf{load}}^L \left[m_* + \left(\frac{n}{m_*}\right)^{1/L}\right] \kappa\right),$$

which gives the claimed moment bounds because $m_* = \Theta(\lambda + \log n)$. On $\mathcal{E}_{\mathsf{samp}}$, Lemma 1 guarantees the bound in Definition 6: each party belongs to at most $(c_{\mathsf{occ}}\bar{\lambda})^j$ committees at level j, hence to $O(\bar{\lambda}^L)$ committees over all levels. At a leaf occurrence it sends to $O(m_* + f) = O(\bar{\lambda} + (n/\bar{\lambda})^{1/L})$ recipients, and at every internal non-root occurrence it sends to at most $f = O((n/\bar{\lambda})^{1/L})$ recipients. Multiplying these quantities by the signature length κ gives the claimed uniform per-party bound.

Rounds and adaptive corruptions. Within one asynchronous time unit after the last of the required h honest parties contributes input, their leaf partials reach their honest recipients and allow every honest member of a live leaf to reconstruct. The accompanying input justifiers enable these reconstruction and forwarding rules without a local input or an additional message exchange. Each of the L edges on a leaf-to-root path then adds at most one asynchronous time unit, since the liveness argument uses only messages between honest parties and reconstruction is local computation. The total is at most $L + 1$ asynchronous time units, or one for the direct construction. Strong adaptivity with takebacks is handled by fixing the final corruption set and using only messages of honest senders in the liveness proof. □

Corollary 1 (Existential contributor set). *Under the assumptions of Theorem 1, except with probability* $\exp(-\lambda) + \mathrm{negl}(\kappa)$*, the following holds simultaneously for every tagged message* $\mu = (\mathsf{sid}, m)$*. Let* T *be the first execution event at which a valid root signature for* μ *is output by a protocol party or presented by the adversary. Let* $C(T) \subseteq [n]$ *be the set of parties corrupted by time* T*, and let* $I_\mu(T) \subseteq [n] \setminus C(T)$ *be the set of parties uncorrupted at* T *that have released at least one of their own authorized leaf signature shares for* μ *by that time. Then there exists a set* $S_\mu(T) \subseteq C(T) \cup I_\mu(T)$ *with* $|S_\mu(T)| = q$*. We call* $S_\mu(T)$ *an* existential contributor set*. It is only an analytical witness and need not be recoverable from the signature. Members of* $S_\mu(T) \cap C(T)$ *need not have contributed a signature through the protocol, since the adversary has their secret shares; every member of* $S_\mu(T) \setminus C(T)$ *has released an authorized leaf partial signature. If* C *is the final corruption set and* $c := |C|$*, then at least* $q - c$ *members of* $S_\mu(T)$ *remain honest throughout the execution and released an authorized share for* μ *by time* T*.*

Proof. Equation (5) gives $|C(T) \cup I_\mu(T)| \ge q$. Choose any q-element subset $S_\mu(T) \subseteq C(T) \cup I_\mu(T)$. Since $C(T) \subseteq C$, every member of $S_\mu(T) \setminus C$ belongs to $I_\mu(T)$ and remains honest throughout the execution. Therefore, $|S_\mu(T) \setminus C| = q - |S_\mu(T) \cap C| \ge q - c$. Every such party released an authorized share only after contributing its own J_{IN}-justified input. □

Corollary 2 (Exclusion of conflicting signatures). *Suppose that* $q > \frac{n+t}{2}$*, and consider two differently tagged value-specific signing instances at the same logical step. If every honest party contributes a signature for at most one of these instances, then valid root signatures cannot be produced for both instances, except with probability* $\exp(-\lambda) + \mathrm{negl}(\kappa)$*.*

Proof. Suppose that valid signatures for the two tagged messages μ_0 and μ_1 are first produced at times T_0 and T_1. For each $b \in \{0, 1\}$, let S_b be an existential contributor set of size q guaranteed by Corollary 1. These sets are chosen only for the analysis and are not extracted from the signatures. Since $S_0, S_1 \subseteq [n]$, $|S_0 \cap S_1| \ge |S_0| + |S_1| - n = 2q - n > t$. Let C be the final corruption set. Since $|C| \le t$, there exists $i \in (S_0 \cap S_1) \setminus C$. This party was never corrupted. Its membership in $S_b \subseteq C(T_b) \cup I_{\mu_b}(T_b)$ therefore implies $i \in I_{\mu_b}(T_b)$ for both $b \in \{0, 1\}$. Hence P_i contributed a signature for both conflicting signing instances, contradicting the assumption that each honest party contributes a signature for at most one of the two instances. □

D Proofs for Byzantine Agreement Components

Lemma 3 (Sampled disperser). *The graph output by* $\mathsf{Setup}_{\mathsf{dissem}}(1^\lambda, n, a, \eta)$ *is an* (a, η)*-dissemination disperser except with probability at most* $\exp(-\lambda)$*. Its maximum left degree satisfies*

$$d_{\mathsf{D}} \le \frac{\ln 4 + \lambda/n}{a\eta} + 1.$$

Since a and η are fixed constants, this is $d_{\mathsf{D}} = O(1 + \lambda/n)$.

Proof. If $d_{\mathsf{D}} = n$, the claim holds deterministically. Suppose that $d_{\mathsf{D}} < n$. If D is not an (a, η)-dissemination disperser, Definition 8 gives a set $A \subseteq [n]$ with $|A| \geq \lceil an \rceil$ and $|[n] \setminus N_D(A)| > \lfloor \eta n \rfloor$. By restricting A to $\lceil an \rceil$ elements and choosing $B \subseteq [n] \setminus N_D(A)$ of size $\lfloor \eta n \rfloor + 1$, we obtain sets of these exact sizes with no edge from A to B. For fixed A and B, this event has probability at most $(1 - |B|/n)^{d_{\mathsf{D}}|A|} \leq e^{-d_{\mathsf{D}}|A||B|/n} \leq e^{-d_{\mathsf{D}} a \eta n}$. There are at most 4^n choices for the pair (A, B). By the definition of d_{D}, the union bound gives a failure probability of at most $4^n \exp(-d_{\mathsf{D}} a \eta n) \leq \exp(-\lambda)$. □

Theorem 2 (Certificate dissemination). *For constants $0 < a, \eta < 1$ with $\lceil an \rceil \leq n - t$, there exists a protocol satisfying Definition 7 against strongly adaptive corruptions with message takebacks, except with probability $\exp(-\lambda)$ over the setup. All required outputs occur within one asynchronous time unit after $\lceil an \rceil$ honest input contributions. For input certificates of at most B bits, the total outgoing communication before corruption is $O((n + \lambda)(B + \kappa + \log n))$ bits.*

Proof. Use $\Pi_{\mathsf{dissem}}(a, \eta, J_{\text{IN}})$ with the setup of Figure 3. Condition on D being an (a, η)-dissemination disperser. Let C be the final corruption set and let $A \subseteq [n] \setminus C$ contain at least $\lceil an \rceil$ honest parties that contribute the J_{IN}-justified value m and an associated justifier as input to the instance. Every party in A executes its prescribed sends exactly once when contributing that justified input, including if it already accepted an output before contributing input, and its messages cannot be suppressed by a later corruption because it is honest. These messages are eventually delivered to every honest out-neighbor. Definition 8 gives $|[n] \setminus N_D(A)| \leq \lfloor \eta n \rfloor$.

Consequently, all but $\lfloor \eta n \rfloor$ honest parties receive and output the J_{IN}-justified value m together with an associated justifier. Every honest output value m is J_{IN}-justified because its accompanying justifier π satisfies $J_{\text{IN}}(m, \pi) = \top$, independently of any messages sent by corrupted parties. By the definition of asynchronous time in Section 3, every party in A completes its prescribed sends as part of its input contribution. Hence every honest recipient in $N_D(A)$ has either already output the J_{IN}-justified value m together with an associated justifier or does so upon delivery of one of these messages, within one asynchronous time unit after the last required input contribution. Each party executes the send rule once, bounding its messages before corruption by its out-degree. At most $n \left(\frac{\ln 4 + \lambda/n}{a\eta} + 1 \right)$ messages of $O(B + \kappa + \log n)$ bits are sent by parties before corruption. Since a and η are fixed constants, this is $O(n + \lambda)$ messages. □

Lemma 4 (Binary sign-or-disseminate). *Let m_0 and m_1 be two distinct candidate values, and let h_0 and h_1 count the honest parties with J_{IN}-justified inputs that contribute them to both primitives for these respective values, with each counted honest party contributing to at most one candidate. A candidate to which no honest party contributes input has count zero. Suppose $h_0 + h_1 \geq H$. Let the threshold signing instances have parameters $(q, a_{\mathsf{C}}, \eta_{\mathsf{C}})$, let the dissemination instances implement $(a_{\mathsf{D}}, \eta_{\mathsf{D}})$-certificate dissemination, and define $h_{\mathsf{C}} := q + \lceil a_{\mathsf{C}} n \rceil$, $h_{\mathsf{D}} := \lceil a_{\mathsf{D}} n \rceil$. If $H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}$, then at least one of the following holds for an index $b \in \{0, 1\}$ maximizing h_b.*

1. *All but at most $\lfloor \eta_{\mathsf{C}} n \rfloor$ honest parties output a valid q-threshold signature for m_b.*
2. *All but at most $2\lfloor \eta_{\mathsf{D}} n \rfloor$ honest parties output both J_{IN}-justified values m_b and m_{1-b} together with associated justifiers.*

Proof. Let b maximize h_b. If $h_{1-b} \geq h_{\mathsf{D}}$, then $h_b \geq h_{1-b} \geq h_{\mathsf{D}}$, so both values meet the liveness threshold h_{D} for certificate dissemination. Applying the liveness property in Definition 7 to both dissemination instances and taking the intersection of their output sets gives the second conclusion. Otherwise, integrality gives $h_{1-b} \leq h_{\mathsf{D}} - 1$, and therefore $h_b \geq H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}$. Applying the liveness property in Definition 5 to the threshold signing instance for m_b gives the first conclusion. □

Lemma 5 (Input validation). *Suppose that all but ϵn honest parties have a binary input. Then all but $\eta_{\mathsf{C}} n$ honest parties output a $J^{\mathsf{valid}}_{\text{OUT}}$-justified bit together with its justifier. Moreover, every $J^{\mathsf{valid}}_{\text{OUT}}$-justified bit was the input of at least one honest party.*

Proof. At least $n - t - \epsilon n$ honest parties have an input. One bit is therefore input by at least $\left\lceil \frac{n - t - \epsilon n}{2} \right\rceil$ honest parties. Using $h_{\mathsf{valid}} \leq t + \epsilon n + 2$, $t \leq (1 - \delta)n/3$, and $6\epsilon < \delta$ gives $\frac{n - t - \epsilon n}{2} - h_{\mathsf{valid}} \geq \frac{\delta - 3\epsilon}{2} n - 2 > 0$. Thus

the more common bit reaches h_{valid}, and this also verifies the feasibility condition $h_{\mathsf{valid}} \le n - t$. The liveness property in Definition 5 gives the claimed outputs. By Corollary 1, the existence of a valid q_{valid}-threshold signature implies that at least $q_{\mathsf{valid}} - t = (t+1) - t = 1$ party that remains honest throughout the execution released an authorized share for that instance. Such a party contributes a signature only for its own input b. Hence b was the input of an honest party. □

Lemma 6 (Recursive proxcensus). *Suppose that $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$ satisfies justified graded agreement and justified validity, and supplies an output to all but ϵn honest parties whenever all but ϵn honest parties have a J_{IN}-justified input. Assume also that all value-specific threshold signing and certificate dissemination instances used in Figure 5 satisfy, respectively, the liveness and unforgeability properties in Definition 5 and the liveness and justified output properties in Definition 7, with the parameters fixed in Section 5.2. Then $\Pi_{\mathsf{prox}(2s-1)}(J_{\text{IN}})$ implements justified almost-everywhere $(2s-1)$-proxcensus as in Definition 9.*

Proof. Let Z be the set of $J^{(s)}_{\text{OUT}}$-justified values. By justified graded agreement of $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$, either Z is a singleton or $Z \subseteq \{z-1, z\}$ for some $z \in \{1, \dots, s-1\}$.

For termination, all but ϵn honest parties obtain an output from $\Pi_{\mathsf{prox}(s)}(J_{\text{IN}})$ and contribute that output as input to threshold signing and certificate dissemination. Apply the binary sign-or-disseminate guarantee of Lemma 4 to these at most two adjacent candidates, assigning count zero to an absent candidate when Z is a singleton. If one value reaches the signing liveness threshold, then all but $\eta_{\mathsf{C}} n \le \eta_{\mathsf{P}} n$ honest parties have already output or can output the corresponding even grade. Otherwise, both values in Z reach the dissemination threshold, and all but $2\eta_{\mathsf{D}} n \le \eta_{\mathsf{P}} n$ honest parties have already output or can output the intervening odd grade.

For justified graded agreement, first observe that every $J^{(2s-1)}_{\text{OUT}}$-justified even output $2x$ implies $x \in Z$. Its justifier contains a valid q_{uniq}-threshold certificate for Sign_x. Since $q_{\mathsf{uniq}} > t$, Corollary 1 implies that a party remaining honest throughout the execution released an authorized share for this certificate. That party contributed a signature to Sign_x only after obtaining the $J^{(s)}_{\text{OUT}}$-justified value x together with an associated justifier. Hence $x \in Z$.

By the soundness of the compact relation proved in Lemma 7, every $J^{(2s-1)}_{\text{OUT}}$-justified odd output $2z-1$ implies that both $z-1$ and z are $J^{(s)}_{\text{OUT}}$-justified, and hence $z-1, z \in Z$. Corollary 2 excludes two different $J^{(2s-1)}_{\text{OUT}}$-justified even output values. Two different $J^{(2s-1)}_{\text{OUT}}$-justified odd output values would require more than two $J^{(s)}_{\text{OUT}}$-justified values, contradicting lower-level justified graded agreement. Finally, if an even output $2x$ and an odd output $2z-1$ are both $J^{(2s-1)}_{\text{OUT}}$-justified, then $x, z-1, z \in Z$, so $x \in \{z-1, z\}$ and their output grades differ by exactly one. Thus any two $J^{(2s-1)}_{\text{OUT}}$-justified outputs differ by at most one.

For justified validity, suppose that b is the only J_{IN}-justified bit. Lower-level justified validity gives $Z \subseteq \{b(s-1)\}$. A $J^{(2s-1)}_{\text{OUT}}$-justified odd output would imply that two distinct adjacent values belong to Z, which is impossible. Every $J^{(2s-1)}_{\text{OUT}}$-justified even output $2x$ satisfies $x \in Z$ by the preceding argument, and therefore $2x = 2b(s-1) = b((2s-1)-1)$. □

Corollary 3 (Common coin from threshold signing). *Fix parameters (q, a, η) satisfying the hypotheses of Theorem 1. For each coin instance, set $m := \texttt{coin}$ and let sid encode the BA session and iteration. Run threshold signing on (sid, m), with signed group element $M := H_{\mathsf{BLS}}(\mathsf{sid}, m)$ as in Section 4.4. Then $c := H_{\mathsf{coin}}(\mathsf{sid}, \sigma_m) \in \{0,1\}$ is an almost-everywhere common coin as in Definition 10, with liveness threshold $h = q + \lceil an \rceil$ and at most ηn missing honest outputs. Its value remains computationally hidden until at least $q - t > 0$ honest parties have contributed a signature. The signature σ_m is a transferable justifier for c. The protocol takes at most $L+1$ asynchronous time units and has the communication complexity stated in Theorem 1.*

Proof. Agreement and almost-everywhere termination follow from uniqueness and the liveness proof of Theorem 1. Consider the first query $H_{\mathsf{coin}}(\mathsf{sid}, \sigma_m)$ in which σ_m is a valid root signature for the coin instance. If σ_m was previously produced in the protocol, apply Corollary 1 at its first production time. Otherwise, if the adversary makes this query before any protocol party produces σ_m, a wrapper adversary outputs σ_m in the threshold signature unforgeability experiment upon observing the query and applies the same corollary at that point.

In either case, there exists an existential contributor set of size q for σ_m. Since at most t parties are ever corrupted, at least $q - t$ members remain honest throughout the execution. Every such party contributed a signature for the coin instance before the coin was exposed. Thus the first valid query at the root signature occurs only after at least $q - t$ honest parties have contributed a signature. At this first query, the corresponding entry of the independent bit-valued random oracle H_{coin} is fresh, so $H_{\mathsf{coin}}(\mathsf{sid}, \sigma_m)$ is a uniform bit independent of the preceding execution prefix. □

E Proofs for Byzantine Agreement

Lemma 8 (Liveness with decision dissemination). *Use the signing and dissemination parameters of Section 5.2 and the decision rule of Figure 6, and suppose every recursive signing hierarchy satisfies Lemma 1 and the dissemination graph satisfies Lemma 3. Suppose also that justified graded agreement holds at every recursive level. If all but ϵn honest parties have either a J_{IN}-justified input to a recursive proxcensus instance or a decision certificate, all but $\eta_{\mathsf{P}} n$ honest parties obtain either its output or a decision certificate. Each recursive level incurs at most one additional asynchronous time unit for decision dissemination.*

Proof. Condition on the stated guarantees for the signing hierarchies and the dissemination graph. Once $h_{\mathsf{D}} = \lceil \epsilon n \rceil$ honest parties hold decision certificates, their contributions to the existing dissemination graph give all but $\eta_{\mathsf{D}} n$ honest parties a decision certificate within one asynchronous time unit. It therefore suffices to prove progress while fewer than h_{D} honest parties have terminated.

For one recursive level, consider the interval of $L + 1$ asynchronous time units after the required number of lower-level outputs or decision certificates is available, where L is the signing hierarchy depth. Assume the dissemination threshold has not been reached by the end of this interval, and let $\mathcal{T}$ be the set of honest parties that have terminated by then, with $d := |\mathcal{T}| < h_{\mathsf{D}}$. Let C be the final corruption set. For signing, apply the root liveness and robust forwarding conclusions of Lemma 1 with $C \cup \mathcal{T}$ in place of C. For every set A of honest contributors outside $\mathcal{T}$ with $|A| \geq h_{\mathsf{C}}$, the sets $C \cup \mathcal{T}$ and A are disjoint, as required by that lemma. The liveness induction in Appendix C therefore applies to honest contributors outside $\mathcal{T}$ and supplies a signature to all but $\eta_{\mathsf{C}} n$ honest parties outside $\mathcal{T}$. For the direct signing construction, $h_{\mathsf{C}} \geq q_{\mathsf{uniq}}$ contributors outside $\mathcal{T}$ send enough shares to every party, so the same conclusion holds within one asynchronous time unit. Members of $\mathcal{T}$ already have decision certificates. The contributor sets used below consist of parties with the required inputs at the start of the interval that remain outside $\mathcal{T}$ until the end of this interval. Every signature contribution counted below therefore comes from a party that has not obtained a decision certificate. If the dissemination threshold is reached during the interval, the first paragraph gives the conclusion instead.

For recursive proxcensus, use induction over its levels. Suppose at least $H := n - t - \epsilon n$ honest parties have either a lower-level output or a decision certificate at the start of the interval. Let h_0, h_1 count these honest parties outside $\mathcal{T}$ with each of the at most two adjacent $J^{(s)}_{\text{OUT}}$-justified grades, assigning zero to an absent candidate. Then $d + h_0 + h_1 \geq H$. Every member of $\mathcal{T}$ sends its decision certificate along the same dissemination graph used by each grade instance. Consequently, if $h_b + d \geq h_{\mathsf{D}}$, the disperser property ensures that all but $\eta_{\mathsf{D}} n$ honest parties obtain either that grade and its justifier or a decision certificate. The graph property depends only on the source set, so it applies to this union of sources even though their messages carry different types of certificates. All these sends have occurred by the end of the interval, so delivery takes at most one additional asynchronous time unit. If both $h_0 + d$ and $h_1 + d$ meet h_{D}, intersecting the two recipient sets gives all but $2\eta_{\mathsf{D}} n$ honest parties either both grade justifiers or a decision certificate. Otherwise, for an index b with $h_{1-b} + d < h_{\mathsf{D}}$,

$$h_b \geq H - d - h_{1-b} \geq H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}.$$

Signing for this grade supplies its threshold certificate to all but $\eta_{\mathsf{C}} n$ honest parties outside $\mathcal{T}$. Each such party has already output from this recursive level or can output the corresponding even grade, and every party in $\mathcal{T}$ holds a decision certificate. These are the same thresholds and the same support inequality as in Lemma 4. Since $\eta_{\mathsf{P}} = \max\{\eta_{\mathsf{C}}, 2\eta_{\mathsf{D}}\} \leq \epsilon$, each recursive level preserves the required number of outputs or decisions. If no $J^{(s)}_{\text{OUT}}$-justified grade exists, all H parties already hold decision certificates and the dissemination threshold is met.

The set $\mathcal{T}$ may depend on the execution prefix, since the graph properties hold simultaneously for all such sets. Thus each recursive level finishes within its component time bound and at most one additional asynchronous time unit, either by producing its output or by delivering a decision certificate. □

Lemma 9 (Bound on justified iterations). *Suppose that Corollary 1, the soundness guarantee of Lemma 7, the justified graded agreement and justified validity properties of proxcensus in Section 5.4, and the justified output property in Definition 7 hold for all instances. Suppose a bit is $J^{\mathsf{BA}}_{\text{OUT}}$-justified by a grade from iteration r of the protocol in Figure 6. Then, for every instance in an iteration numbered $r+2$ or higher with input justification predicate $\mathcal{J}$, there is no $\mathcal{J}$-justified input value.*

Proof. Let b be the bit justified by an extreme grade in iteration r. Justified graded agreement restricts every $J^{(G,r)}_{\text{OUT}}$-justified grade to $\{0,1\}$ if $b=0$ and to $\{G-2,G-1\}$ if $b=1$. Each justifier accepted by $J^{(r+1)}_{\text{IN}}$ contains a grade in $\{1,\dots,G-2\}$, so every $J^{(r+1)}_{\text{IN}}$-justified bit equals b. If such a bit exists, justified validity makes every $J^{(G,r+1)}_{\text{OUT}}$-justified grade equal to $b(G-1)$. Since this is an extreme grade, it cannot justify an input to iteration $r+2$.

If no $J^{(r+1)}_{\text{IN}}$-justified bit exists, no $J^{(G,r+1)}_{\text{OUT}}$-justified grade exists either. Indeed, at the two-grade base level the output predicate is the input predicate. At every recursive level, a $J^{(2s-1)}_{\text{OUT}}$-justified even grade implies a $J^{(s)}_{\text{OUT}}$-justified grade by Corollary 1, and a $J^{(2s-1)}_{\text{OUT}}$-justified odd grade implies two such grades by Lemma 7. Induction on the recursion depth therefore excludes every $J^{(G,j)}_{\text{OUT}}$-justified grade whenever no $J^{(j)}_{\text{IN}}$-justified input bit exists.

In either case, no $J^{(r+2)}_{\text{IN}}$-justified input bit exists. Repeating the same argument excludes every $J^{(j)}_{\text{IN}}$-justified input bit and every $J^{(G,j)}_{\text{OUT}}$-justified grade for $j \geq r+2$. Inputs to signing and dissemination at each recursive level require a justifier from the preceding level, and inputs to the common coin require a proxcensus grade from the same iteration. Thus, for every instance in an iteration numbered $r+2$ or higher with input justification predicate $\mathcal{J}$, there is no $\mathcal{J}$-justified input value. □

Lemma 10 (Binary agreement from justified inputs). *Under the setup and computational assumptions of Section 4, there is an instantiation of Π_{BA} using the components of Section 5 with the following guarantees. For every initial justification predicate $J^{(0)}_{\text{IN}}$, $\Pi_{\mathsf{BA}}(J^{(0)}_{\text{IN}})$ satisfies justified agreement and justified validity of Definition 4, except with negligible probability. If all but ϵn honest parties hold a $J^{(0)}_{\text{IN}}$-justified input, all but $\eta_{\mathsf{P}} n$ honest parties terminate in an expected $O(1)$ iterations and asynchronous rounds. These expectations are conditioned on the event $\mathcal{E}_{\mathsf{BA}}$ defined in its proof, which has probability $1-\text{negl}(\kappa)$.*

Proof. Use the recursive proxcensus and common coin constructions of Section 5 with $G=9$ and $\eta_{\mathsf{P}}+\eta_{\mathsf{coin}} \leq \epsilon$. For the safety and liveness arguments, condition on the guarantees of Definitions 9 and 10, Corollary 1, and Lemmas 7, 1, and 3 for the relevant instances. The aggregate failure probability is included in p_{bad} below. We first prove justified validity. Suppose that b is the only $J^{(0)}_{\text{IN}}$-justified bit. Justified validity of proxcensus implies that every $J^{(G,0)}_{\text{OUT}}$-justified grade equals $b(G-1)$. Extreme grades do not justify inputs to the next iteration, so no input bit or output grade is justified in a later iteration, by the recursive argument in the proof of Lemma 9. Every $J^{\mathsf{BA}}_{\text{OUT}}$-justified decision bit therefore equals b, including decisions received through decision dissemination.

We next prove justified agreement. Suppose, toward a contradiction, that b and $1-b$ are $J^{\mathsf{BA}}_{\text{OUT}}$-justified by grades in iterations r and r', respectively, where, without loss of generality, $r \leq r'$. If $r=r'$, then grades 0 and $G-1$ are both $J^{(G,r)}_{\text{OUT}}$-justified, contradicting justified graded agreement. Hence $r<r'$. Since the decision for b is justified in iteration r, justified graded agreement restricts every $J^{(G,r)}_{\text{OUT}}$-justified grade to $\{0,1\}$ when $b=0$, and to $\{G-2,G-1\}$ when $b=1$. Consequently, every $J^{(r+1)}_{\text{IN}}$-justified bit equals b. If such a bit exists, justified validity of proxcensus forces every $J^{(G,r+1)}_{\text{OUT}}$-justified grade to equal $b(G-1)$. If no such bit exists, no output grade is justified in that iteration, by the recursive argument in the proof of Lemma 9. In either case, Lemma 9 excludes justified inputs and grades in iteration $r+2$ and later. Thus iteration r' cannot justify a decision for $1-b$, a contradiction. Decision dissemination preserves this agreement because a recipient checks the same output justification predicate.

It remains to prove almost-everywhere termination and the expected iteration bound. By hypothesis, all but ϵn honest parties initially hold a $J^{(0)}_{\text{IN}}$-justified input. Until an honest party first obtains a decision

certificate, no honest party terminates. The component termination properties therefore give the following induction: within the proxcensus round bound in iteration r, either an honest party obtains a decision certificate, or all but $\eta_{\mathsf{P}} n$ honest parties obtain a nonextreme proxcensus grade and contribute input to the common coin. In the latter case, either a decision certificate is obtained during the coin instance, or all but $\eta_{\mathsf{coin}} n$ honest parties obtain its output. Intersecting these sets supplies all but $(\eta_{\mathsf{P}} + \eta_{\mathsf{coin}})n \leq \epsilon n$ honest parties with a $J_{\text{IN}}^{(r+1)}$-justified input.

Now suppose an honest party first obtains a decision certificate for bit b in iteration r. Write $G = 2s - 1$ and $z = b(s-1)$, so the certificate is a threshold signature for Sign_z at the final recursive level of iteration r. Corollary 1 implies that at least $q_{\mathsf{uniq}} - t \geq h_{\mathsf{D}}$ parties remaining honest throughout the execution contributed signatures for this instance before the certificate was obtained. Each had its own lower-level output (z, π_z) and contributed it to Dissem_z before contributing a signature, as prescribed in Figure 5. Their dissemination therefore supplies (z, π_z) to all but $\eta_{\mathsf{D}} n$ honest parties within one asynchronous time unit. Since $G = 2^k + 1 \geq 6$, the lower-level extreme grade z is even, and π_z is itself a threshold certificate for the corresponding signing instance of $\Pi_{\mathsf{prox}(s)}$. Every recipient that has neither terminated nor already output from $\Pi_{\mathsf{prox}(s)}$ can therefore output (z, π_z), since Figure 5 prescribes this output upon obtaining that threshold certificate. Thus all but $\eta_{\mathsf{D}} n \leq \epsilon n$ honest parties have either their own lower-level output or a decision certificate, irrespective of any unfinished earlier coin instances. The final recursive step of Lemma 8 now gives all but $\eta_{\mathsf{P}} n$ honest parties either an iteration-r proxcensus output or a decision certificate.

Justified graded agreement restricts every $J_{\text{OUT}}^{(G,r)}$-justified grade to the deciding grade and its adjacent grade. No such grade requires a coin output: extreme grades cause decision dissemination, and adjacent grades supply the next input directly. Among at least $H := n - t - \epsilon n$ honest parties, each therefore either holds a decision certificate or contributes b to iteration $r + 1$. If at least h_{D} honest parties hold decision certificates, decision dissemination gives all but $\eta_{\mathsf{D}} n$ honest parties a decision certificate. Otherwise, at least $H - h_{\mathsf{D}} + 1 \geq h_{\mathsf{C}}$ honest parties contribute the same bit b to iteration $r+1$. Together, these contributors and the decision certificate holders meet the premise of Lemma 8 for iteration $r+1$. Justified validity makes every $J_{\text{OUT}}^{(G,r+1)}$-justified grade extreme, giving all but $\eta_{\mathsf{P}} n$ honest parties either that grade or a decision certificate. Either outcome makes them decide and disseminate. This uses the same signing and dissemination thresholds as Lemma 4 and takes a constant number of asynchronous rounds after the first decision certificate is obtained.

If the set of $J_{\text{OUT}}^{(G,r)}$-justified grades is nonempty and contains no extreme grade, it is a singleton or consists of two adjacent nonextreme grades. When all these grades lie in $\{2, \ldots, G-3\}$, every $J_{\text{IN}}^{(r+1)}$-justified bit equals c_r. When the only grade is 1 or $G-2$, every $J_{\text{IN}}^{(r+1)}$-justified bit equals 0 or 1, respectively.

To justify the probability of obtaining a common input for the next iteration, consider the execution prefix immediately before the coin is first exposed. An honest party has already contributed input to the coin instance and therefore recorded a $J_{\text{OUT}}^{(G,r)}$-justified proxcensus grade before this exposure. Choose such a recorded grade g by a fixed rule depending only on this prefix. Justified graded agreement confines every $J_{\text{OUT}}^{(G,r)}$-justified grade to $\{g-1, g, g+1\} \cap \{0, \ldots, G-1\}$. If $g \leq 2$, coin value 0 makes all $J_{\text{IN}}^{(r+1)}$-justified input bits equal 0; if $g \geq G-3$, coin value 1 makes them all equal 1. For $3 \leq g \leq G-4$, every $J_{\text{OUT}}^{(G,r)}$-justified grade lies in $\{2, \ldots, G-3\}$, and either coin value makes the input for the next iteration unique. The conditions $g \leq 2$ and $g \geq G-3$ are disjoint because $G \geq 6$. In the experiment defined in Section 5.5, the bit revealed at this exposure is uniform and independent of the preceding execution prefix, giving probability at least $1/2$ for the favorable value fixed by g. The real execution differs from this experiment only with negligible probability over every polynomial execution prefix. If a $J_{\text{IN}}^{(r+1)}$-justified bit exists, justified validity then restricts every $J_{\text{OUT}}^{(G,r+1)}$-justified grade to the corresponding extreme grade. The resulting geometric bound applies to every polynomial execution prefix, up to the aggregate negligible failure probability of the primitives in that prefix. For the analysis only, set $K := \kappa$ and let $p_{\mathsf{bad}}(\kappa)$ bound the probability of any setup or primitive failure in iterations $0, \ldots, K+2$, together with every invalid certificate or input justifier accepted from the polynomial adversarial transcript. Let T be the number of iterations up to the first iteration r for which at most one bit is $J_{\text{IN}}^{(r+1)}$-justified, and define $\mathcal{E}_{\mathsf{BA}}$ to be the event that these primitive guarantees hold and $T \leq K$. For $T > j$, each of the first j iterations must admit both input bits for the next iteration. This requires a coin exposure in each such iteration, with a value other than the favorable value fixed by the recorded grade. The independent uniform coins in this experiment therefore give

$$\Pr[T > j] \leq 2^{-j} + p_{\mathsf{bad}}(\kappa), \qquad 0 \leq j \leq K.$$

If an iteration has no $J_{\text{IN}}^{(r+1)}$-justified bit, it already satisfies the condition defining T. Consequently, $\Pr[\neg\mathcal{E}_{\mathsf{BA}}] \le 2^{-K} + p_{\mathsf{bad}}(\kappa)$ after enlarging p_{bad} to cover all primitive failures. For every fixed positive integer d, summing this tail up to K gives

$$\mathbb{E}[T^d \mid \mathcal{E}_{\mathsf{BA}}] \le \frac{O(1) + O(K^d p_{\mathsf{bad}}(\kappa))}{\Pr[\mathcal{E}_{\mathsf{BA}}]} = O(1).$$

On $\mathcal{E}_{\mathsf{BA}}$, at most one bit is $J_{\text{IN}}^{(T)}$-justified. The induction before the first decision certificate ensures that either an honest party obtains a decision certificate within these $T+1$ iterations, or all but $\eta_{\mathsf{P}} n$ honest parties obtain an output in iteration T. If no $J_{\text{IN}}^{(T)}$-justified bit exists, no $J_{\text{OUT}}^{(G,T)}$-justified grade exists, so the latter case is impossible. Otherwise, justified validity makes every such output grade extreme, and an honest party obtains a decision certificate in this case as well. The dissemination and support argument above then gives all but $\eta_{\mathsf{P}} n$ honest parties a decision certificate within a constant number of further asynchronous rounds. Let r be the least iteration index admitting a $J_{\text{OUT}}^{(G,r)}$-justified extreme grade. If the unique $J_{\text{IN}}^{(T)}$-justified bit exists, no input is justified after iteration T, since every $J_{\text{OUT}}^{(G,T)}$-justified grade is extreme; if no such bit exists, no grade is justified in iteration T or later. Hence $r \le T$, and Lemma 9 excludes justified inputs in iteration $r+2$ and later. Each iteration before the first decision certificate uses one proxcensus instance and one common coin instance, each with a constant asynchronous round bound. Lemma 8 adds at most one asynchronous time unit per recursive level afterwards, so the moment bounds for T give constant expected asynchronous round complexity for almost-everywhere termination. Each party contributes to decision dissemination at most once and performs no further protocol work after completing those sends. □

Lemma 11 (Two-value anchoring across BA iterations). *Let $G = 2^k + 1 \ge 5$ be obtained from the recursive proxcensus construction. If $J_{\text{IN}}^{(0)} = J_{\text{OUT}}^{\mathsf{valid}}$, then $J_{\text{IN}}^{(r)}$ has the two-value anchoring property of Definition 11 in every BA iteration $r \ge 0$.*

Proof. For $r = 0$, every $J_{\text{IN}}^{(0)}$-justified bit has only q_{valid}-threshold certificates produced by input validation as its associated justifiers. Fix an iteration r and suppose that both bits are $J_{\text{IN}}^{(r+1)}$-justified. Justified graded agreement restricts all $J_{\text{OUT}}^{(G,r)}$-justified grades in iteration r to one value or two adjacent values. Inspection of the update rule shows that two different bits for the next iteration are possible only for the boundary pairs $\{1,2\}$ and $\{G-3, G-2\}$. For $\{1,2\}$, the $J_{\text{IN}}^{(r+1)}$-justified bit 1 has only associated justifiers containing a justifier for the even grade 2 and a coin certificate, and this grade justifier is a single q_{uniq}-threshold certificate. For $\{G-3, G-2\}$, the $J_{\text{IN}}^{(r+1)}$-justified bit 0 has only associated justifiers containing a justifier for the even grade $G-3$ and a coin certificate, where $G-3$ is even because $G = 2^k + 1$. Thus, one of the two $J_{\text{IN}}^{(r+1)}$-justified bits has only threshold-anchored associated justifiers of $O(\kappa)$ bits. □

Theorem 3 (Binary agreement with almost-everywhere termination). *Fix constants $\delta > 0$, $0 < \nu < 1$, and an integer $c \ge 1$. There is a constant $\epsilon > 0$ such that, for all sufficiently large $n \le \kappa^c$ and $t \le (1-\delta)n/3$, there exists a protocol satisfying Definition 2 with parameters (ϵ, ν) against strongly adaptive corruptions with message takebacks, except with negligible probability. The protocol assumes authenticated reliable asynchronous channels, the setup and computational assumptions of Section 4, and the independent random oracle of Section 5.5. If all but ϵn honest parties receive an input, its expected complexity is $O(1)$ asynchronous rounds, $O(n\kappa^2)$ total bits, and $\widetilde{O}(\kappa^2)$ maximum outgoing bits per party. Communication is measured in bits and counted before each party's corruption, and expectations are conditioned on $\mathcal{E}_{\mathsf{BA}}$ defined in the proof of Lemma 10, which has probability $1 - \mathrm{negl}(\kappa)$.*

Proof. Choose $0 < \epsilon < \min\{\delta/6, \nu\}$, set $G = 9$, and use a fixed signing hierarchy depth $L \ge c$ and $\lambda = \Theta(\kappa)$. Then $\lambda = \Omega(\kappa + \log n)$ because $n \le \kappa^c$. Set $\eta_{\mathsf{C}} := \epsilon/4$, $\eta_{\mathsf{D}} := \epsilon/8$, $\eta_{\mathsf{P}} := \max\{\eta_{\mathsf{C}}, 2\eta_{\mathsf{D}}\} = \epsilon/4$, and $\eta_{\mathsf{coin}} := \epsilon/4$. Use the input validation, proxcensus, and common coin implementations in Section 5. Moreover, $\eta_{\mathsf{P}} + \eta_{\mathsf{coin}} = \epsilon/2 < \epsilon$.

The dealer independently generates a committee hierarchy, forwarding maps, and recursively shared signing key for each of the two certificate thresholds q_{valid} and q_{uniq} and for the common coin, using the corresponding values of (q, h, ϕ, k_v). These three setups and the dissemination graph are sampled once and reused for all domain-separated instances of the BA session. Input validation starts with the BA session,

and each party contributes input to iteration 0 upon obtaining its validated input. Each signing instance uses its prescribed input justification predicate, and parties verify the accompanying input justifier before reconstructing or forwarding. For the common coin in iteration r, an input justifier is a pair (g, ζ) satisfying $J_{\text{OUT}}^{(G,r)}(g, \zeta) = \top$; an honest party contributes a signature only upon obtaining its own such grade with $1 \le g \le G - 2$ and while it has no decision certificate.

Theorem 1 implements the two threshold signing instances required by input validation and recursive proxcensus, with their claimed safety and liveness thresholds, against strongly adaptive corruptions with message takebacks. Theorem 2 implements every certificate dissemination instance within one asynchronous time unit after the required input contributions and in linear outgoing communication. Lemma 4 supplies the liveness alternative for recursive proxcensus. Suppose that all but ϵn honest parties receive a binary input. Until an honest party first obtains a decision certificate, every honest party continues the component protocols. Thus Lemma 5 either supplies all but $\eta_{\mathsf{C}} n \le \epsilon n$ honest parties with a $J_{\text{IN}}^{(0)}$-justified input, or an honest party obtains a decision certificate while input validation is in progress. In the latter case, the dissemination and support argument in the proof of Lemma 10 gives almost-everywhere termination directly. The choices $\eta_{\mathsf{C}} = \epsilon/4$ and $\eta_{\mathsf{D}} = \epsilon/8$ give $\eta_{\mathsf{P}} = \max\{\eta_{\mathsf{C}}, 2\eta_{\mathsf{D}}\} = \epsilon/4$.

Corollary 3 gives a common coin that supplies outputs to all but $\eta_{\mathsf{coin}} n$ honest parties whenever all but $\eta_{\mathsf{P}} n$ honest parties contribute input, and remains unpredictable until at least one honest signature contribution. The inequality $\eta_{\mathsf{P}} + \eta_{\mathsf{coin}} < \epsilon$ preserves enough inputs between iterations until the first decision certificate is obtained. Lemma 10 therefore gives agreement, almost-everywhere termination, and expected constant round complexity. At most $\eta_{\mathsf{P}} n$ honest parties fail to output, which is at most a ν fraction because $\eta_{\mathsf{P}}/(1 - t/n) \le 3\epsilon/8 < \nu$.

For validity, Corollary 1 implies that the existence of a $q_{\mathsf{valid}} = t + 1$ input validation certificate entails that at least $q_{\mathsf{valid}} - t = 1$ honest party released an authorized signature share after contributing that initial bit as input to the input validation instance. If only one bit obtains such a certificate, justified validity of Π_{BA} (Lemma 10) forces that bit; if both bits obtain certificates, either possible decision is the input of an honest party.

We prove the following more general bounds for every fixed $L \ge 1$ and $\lambda = \Omega(\kappa + \log n)$, assuming that all but ϵn honest parties receive an input. Use d_{load} as defined in the proof of Theorem 1. All expectations are conditioned on $\mathcal{E}_{\mathsf{BA}}$, defined in the proof of Lemma 10, whose complement has negligible probability under this input premise. For every fixed L, the expected communication is

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right]\kappa + (n + \lambda)(\kappa + \log n)\right). \tag{8}$$

If C_i^{BA} is the communication sent by P_i before corruption, then

$$\mathbb{E}\left[\max_{i \in [n]} C_i^{\mathsf{BA}}\right] = O\left(d_{\mathsf{load}}^{L}\left[\lambda + \log n + \left(\frac{n}{\lambda + \log n}\right)^{1/L}\right]\kappa + \left(1 + \frac{\lambda}{n}\right)(\kappa + \log n)\right). \tag{9}$$

For constant $L \ge 2$, its dependence on n is $O(n^{1/L})$, up to security parameter and logarithmic factors, so the protocol is balanced.

If $n \le \kappa^c$ for a fixed integer $c \ge 1$ and $\lambda = \Theta(\kappa)$, then (8) becomes

$$O\left(n\kappa^2 + n\kappa^{1+(c-1)/L} + \kappa^2\right). \tag{10}$$

Hence any fixed integer $L \ge c$ gives expected communication $O(n\kappa^2)$, expected maximum communication $\widetilde{O}(\kappa^2)$ before corruption, and expected constant asynchronous round complexity.

The construction of 9-proxcensus uses three recursive applications, from 2 to 3, from 3 to 5, and from 5 to 9. Each application uses a constant number of threshold signing and dissemination instances, and each party contributes to one value-specific instance of each type. Let B_r be the maximum length of a canonical justifier π accompanying an iteration-r input bit b used by any party before corruption, with $J_{\text{IN}}^{(r)}(b, \pi) = \top$. The input validation certificate gives $B_0 = O(\kappa)$. Lemma 11 establishes the two-value anchoring property required by Lemma 7 in every iteration. Lemma 7 bounds a grade justifier in iteration r by $B_r + O(\kappa)$, and the next iteration rule adds at most one $O(\kappa)$-bit coin justifier. Consequently,

$$B_{r+1} \le B_r + O(\kappa) \qquad \text{and hence} \qquad B_r = O((r + 1)\kappa).$$

The one-time input validation phase can activate at most two value-specific q_{valid}-threshold signing instances, one for each bit. Their raw input justifiers are canonically $\perp$ and are omitted from messages. By Theorem 1, their combined communication is

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right]\kappa\right).$$

This one-time term is absorbed into the first term of Equation (8). Its maximum outgoing cost, which depends on the setup, is likewise included in the variable X used below and is therefore absorbed into Equation (9). The threshold signing protocol and the common coin each use

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right](B_r + \kappa)\right)$$

communication per BA iteration. Certificate dissemination in iteration r uses $O((n+\lambda)(B_r+\kappa+\log n))$ bits, and one iteration has $O(L)$ asynchronous rounds. Its left degree is $O(1+\lambda/n)$, so every party sends $O((1+\lambda/n)(B_r+\kappa+\log n))$ bits in that iteration. Let R be one plus the largest iteration index in which any party performs prescribed work before its corruption. Decision dissemination is counted separately, and all execution expectations in this proof are conditioned on $\mathcal{E}_{\mathsf{BA}}$. Let T be the number of iterations defined in the proof of Lemma 10. Lemma 9 bounds R by $T+O(1)$, so the bound on $\Pr[T>j]$ proved there gives $\mathbb{E}[R^d] = O(1)$ for every fixed positive integer d. Each party contributes to decision dissemination at most once, with an $O(\kappa)$-bit certificate. The resulting total communication is $O((n+\lambda)(\kappa+\log n))$, and each party sends $O((1+\lambda/n)(\kappa+\log n))$ bits, so this cost is absorbed by the dissemination terms below. The three configurations of recursive sharing and the dissemination graph are each sampled once, so their probabilities of setup failure are added only a constant number of times, not once per iteration. Computational failures are bounded jointly over the polynomial execution prefix and all adversarial attempts to forge certificates or input justifiers, with $K=\kappa$ used only in the proof. Therefore,

$$\begin{aligned}\mathbb{E}\left[\sum_{r=0}^{R-1}(n+\lambda)(B_r+\kappa+\log n)\right] &= O\big((n+\lambda)\big(\kappa\mathbb{E}[R^2] + (\kappa+\log n)\mathbb{E}[R]\big)\big)\\ &= O((n+\lambda)(\kappa+\log n))\,.\end{aligned}$$

Since $\sum_{r=0}^{R-1}(B_r+\kappa) = O(\kappa R^2)$ and $\mathbb{E}[R^2] = O(1)$, summing the per-iteration threshold signing and common coin costs and adding the one-time input validation cost gives

$$O\left(\left[n\lambda + n\left(\frac{n}{\lambda}\right)^{1/L}\right]\kappa\right)$$

bits. This proves (8). The same degree bound gives the cumulative maximum outgoing dissemination cost

$$\max_{i\in[n]} C^{\mathsf{BA}}_{i,\mathsf{dissem}} = O\left(\left(1+\frac{\lambda}{n}\right)\left(\kappa R^2 + R\log n\right)\right),$$

because $B_r = O((r+1)\kappa)$. Therefore,

$$\mathbb{E}\left[\max_{i\in[n]} C^{\mathsf{BA}}_{i,\mathsf{dissem}} \,\middle|\, \mathcal{E}_{\mathsf{BA}}\right] = O\left(\left(1+\frac{\lambda}{n}\right)(\kappa+\log n)\right).$$

Let X be the sum of the maximum outgoing communication bounds $X(\mathcal{G})$ from the proof of Theorem 1 for one instance with $O(\kappa)$-bit justifiers under each of the three setups, multiplied by the constant number of instances per iteration. The second-moment bound in that proof and Cauchy–Schwarz for this constant sum give, without conditioning on an execution,

$$\left(\mathbb{E}[X^2]\right)^{1/2} = O\left(d_{\mathsf{load}}^L\left[\lambda + \log n + \left(\frac{n}{\lambda+\log n}\right)^{1/L}\right]\kappa\right).$$

Since $\Pr[\mathcal{E}_{\mathsf{BA}}] = 1 - \mathrm{negl}(\kappa)$, conditioning increases this second moment by at most a constant factor. In iteration r, including its input justifiers multiplies this bound by $O(r+1)$, so the maximum cumulative communication of the threshold signing and coin protocols is $O(XR^2)$, including reconstruction and forwarding. No independence between hierarchy load and the execution length is required, because

$$\mathbb{E}[XR^2 \mid \mathcal{E}_{\mathsf{BA}}] \le \sqrt{\mathbb{E}[X^2 \mid \mathcal{E}_{\mathsf{BA}}]\,\mathbb{E}[R^4 \mid \mathcal{E}_{\mathsf{BA}}]}.$$

Combining this bound with the dissemination calculation proves (9).

Finally, $n \le \kappa^c$ and $\lambda = \Theta(\kappa)$ imply $\log n = O(\log \kappa)$ and

$$n\left(\frac{n}{\lambda}\right)^{1/L}\kappa = O\Big(n\kappa^{1+(c-1)/L}\Big),$$

while

$$(n+\lambda)(\kappa + \log n) = O(n\kappa + \kappa^2).$$

This gives (10). If $L \ge c$, then $(c-1)/L \le 1$, so every term is $O(n\kappa^2)$.

For the maximum per-party bound, $d^L_{\mathsf{load}} = \mathrm{polylog}(\kappa)$ because L is fixed, and

$$\lambda + \log n + \left(\frac{n}{\lambda + \log n}\right)^{1/L} = O\Big(\kappa + \kappa^{(c-1)/L}\Big) = O(\kappa).$$

Moreover,

$$\left(1 + \frac{\lambda}{n}\right)(\kappa + \log n) = O(\kappa^2).$$

Equation (9) is therefore $\widetilde{O}(\kappa^2)$. Finally, one BA iteration takes $O(L)$ asynchronous rounds. Since L is fixed and the expected number of iterations is constant, the expected asynchronous round complexity remains $O(1)$. □

Corollary 4 (Linear full binary agreement). *Under the assumptions of Theorem 3, there exists a protocol satisfying Definition 3 against adaptive corruptions without message takebacks, except with negligible probability. When every honest party receives an input, it has expected $O(1)$ asynchronous rounds, $O(n\kappa^2)$ total bits, and $\widetilde{O}(\kappa^2)$ maximum outgoing bits per party before corruption. These expectations are conditioned on $\mathcal{E}_{\mathsf{full}}$ defined in the proof of Corollary 4, which has probability $1 - \mathrm{negl}(\kappa)$.*

Proof. Use the construction in the proof of Theorem 3, with $\lambda = \Theta(\kappa)$ and fixed $L \ge c$, and apply the dissemination rule of Section 6.2. Let $p_{\mathsf{AE}} := \Pr[\neg\mathcal{E}_{\mathsf{BA}}]$. Let $\mathcal{E}_{\mathsf{full}}$ be the event that $\mathcal{E}_{\mathsf{BA}}$ holds and the first $\lceil \alpha_{\mathsf{out}} n \rceil d$ prescribed sends completed by uncorrupted senders collectively select every honest recipient. For termination and the expected complexity bounds, suppose that every honest party receives an input, as in Definition 3. This satisfies the requirement that all but ϵn honest parties receive an input in Theorem 3. The proof of Theorem 3 remains valid when additional messages deliver accepted decision certificates, since every recipient follows the same decision dissemination rule. On $\mathcal{E}_{\mathsf{BA}}$, at least $\lceil n - t - \eta_{\mathsf{P}} n \rceil = \lceil \alpha_{\mathsf{out}} n \rceil$ honest parties therefore obtain a $J^{\mathsf{BA}}_{\mathrm{OUT}}$-justified decision bit and its transferable justifier. Each performs the d additional sends before terminating, whether it obtained its certificate from proxcensus, decision dissemination, or another party's additional send. Consequently, on $\mathcal{E}_{\mathsf{BA}}$, at least $M := \lceil \alpha_{\mathsf{out}} n \rceil d$ prescribed sends are completed while their senders are uncorrupted.

Let $S_1, S_2, \ldots$ be the chronological sequence of prescribed point-to-point sends completed while their senders are uncorrupted; a send remains in this sequence even if its sender is corrupted later. On $\mathcal{E}_{\mathsf{BA}}$, the first M such sends exist.

Fix an identity $j \in [n]$. For every ℓ, consider any execution history immediately before the action S_ℓ that samples the recipient and sends the message. Conditional on this history, the recipient is a fresh uniform identity in $[n]$. Because recipient sampling and sending form one atomic action, the adversary cannot use the sampled recipient to decide whether the send enters the sequence. Therefore, iterating the conditional bound gives

$$\Pr[S_1, \ldots, S_M \text{ exist and all miss } P_j] \le \left(1 - \frac{1}{n}\right)^M \le \exp(-M/n) \le \exp(-\alpha_{\mathsf{out}} d) \le \frac{\exp(-\lambda)}{n}.$$

This calculation does not condition on whether P_j remains honest. A union bound over all n identities therefore gives

$$\Pr[\mathcal{E}_{\mathsf{BA}} \wedge \neg \mathcal{E}_{\mathsf{full}}] \leq \exp(-\lambda).$$

On $\mathcal{E}_{\mathsf{full}}$, every honest party is selected by a send committing the $J^{\mathsf{BA}}_{\mathrm{OUT}}$-justified decision bit and its associated justifier for eventual delivery, and no subsequent corruption of its sender can retract that message. Adding p_{AE} gives $\Pr[\neg \mathcal{E}_{\mathsf{full}}] \leq p_{\mathsf{AE}} + \exp(-\lambda) = \mathrm{negl}(\kappa)$.

Agreement follows because justified agreement of Π_{BA} makes all $J^{\mathsf{BA}}_{\mathrm{OUT}}$-justified decision bits equal. Validity is inherited from Theorem 3 because dissemination introduces no new justified value. The top-level decision grades 0 and $G-1$ are even, so their justifiers are single $O(\kappa)$-bit threshold certificates. Each party performs the additional sends only for its first decision certificate and therefore sends at most d such messages before corruption, for a total of at most $nd = O(n\lambda)$ prescribed messages of $O(\kappa + \log n)$ bits. Once the required honest parties have obtained their decision certificates, each performs its finite sequence of send actions without waiting for another message. By the definition of asynchronous time in Section 3, the first M committed sends therefore occur without waiting for another message delivery. On $\mathcal{E}_{\mathsf{full}}$, every honest recipient is selected by at least one such send. That message was committed while its sender was uncorrupted and cannot be retracted. Since asynchronous time without takebacks is measured using the maximum delay of messages sent before corruption to honest recipients (Section 3), every honest party receives the $J^{\mathsf{BA}}_{\mathrm{OUT}}$-justified decision bit together with an associated justifier within one additional asynchronous time unit.

The joint failure bound gives

$$\Pr[\mathcal{E}_{\mathsf{full}} \mid \mathcal{E}_{\mathsf{BA}}] \geq 1 - \frac{\exp(-\lambda)}{1 - p_{\mathsf{AE}}} = 1 - \mathrm{negl}(\kappa).$$

For every nonnegative cost X of the base protocol, conditioning further on successful delivery increases its expectation by at most the reciprocal of this probability, since

$$\mathbb{E}[X \mid \mathcal{E}_{\mathsf{full}}] \leq \frac{\mathbb{E}[X \mid \mathcal{E}_{\mathsf{BA}}]}{\Pr[\mathcal{E}_{\mathsf{full}} \mid \mathcal{E}_{\mathsf{BA}}]}.$$

This factor is bounded by a constant for all sufficiently large security parameters. The maximum message delay without takebacks is at least the maximum delay between honest parties used in the analysis of the base protocol, so it cannot increase the normalized base running time. Applying the inequality to base outgoing communication, maximum outgoing communication and running time, and adding the deterministic dissemination overheads, proves the conditional expected bounds. □