

Fully Fluctuating Sleepy Consensus from Minimal Assumptions

Javier Nieto ✉ 

University of Illinois Urbana-Champaign, Urbana, IL, USA

Yuval Efron ✉ 

Institute for Advanced Study, Princeton, NJ, USA

Joachim Neu ✉ 

a16z Crypto Research, New York, NY, USA

Ling Ren ✉ 

University of Illinois Urbana-Champaign, Urbana, IL, USA

Abstract

Bitcoin’s proof-of-work (PoW)-based protocol is remarkable for how little it asks of its participants. Not only can miners take breaks from work whenever they please, but it is almost unique in offering a path of contrition: corrupt miners can reclaim honest status simply by resuming mining on the longest chain. The protocol only requires that honest miners hold the majority of computational power at any given time. Analogous proof-of-stake (PoS) protocols, usually formalized via the sleepy model of Pass and Shi [46], have fallen short of matching this robustness. In fact, sleepy consensus protocols in the plain PKI model *must* heavily restrict fluctuations in adversarial participation over time. The recent work of Efron et al. [21] enables fully fluctuating participation in the sleepy model by introducing the external adversary model. Their protocol, however, relies on verifiable delay functions (VDFs), a strong cryptographic primitive that somewhat resembles PoW, by assuming that the adversary cannot compute *sequential* work significantly faster than honest nodes.

In this work, we design a sleepy consensus protocol for fully fluctuating participation with an external adversary under an honest majority, from minimal assumptions: a public key infrastructure (PKI) and a verifiable random function (VRF). In particular, we make no VDF or hardware assumptions. Our key technique is *graded wakeness*, a novel primitive that allows nodes to form consistent opinions on which other nodes are awake. We further extend our protocol to handle *uncorruption*, where corrupt nodes return to honesty. This extension requires only a mild additional assumption on the unpredictability of VRF outputs for liveness.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Byzantine agreement, proof of stake, sleepy model, fluctuating participation, graded wakeness, verifiable random function

Funding This work is funded in part by the National Science Foundation award #2143058.

1 Introduction

The *Byzantine agreement* problem was first described by Lamport et al. [35] as a group of generals who must reach consensus on a battle plan, despite the presence of traitors among them. This abstraction laid the foundation for fault-tolerant distributed algorithms for decades. The classic setting appeals to our intuition via an analogy to a standing army, in which participation is mandatory.

Modern decentralized systems, such as permissionless blockchains, defy this rigid structure. Bitcoin’s proof-of-work (PoW) [43], for instance, is indifferent to how many miners are participating. Here, miners more closely resemble a group of volunteers, where the system proceeds with whoever happens to be participating at any given time. Accordingly, Bitcoin has accommodated growth from just a handful of miners to many thousands, as well as

occasional notable declines due to various factors such as outages, government regulations, block reward “halving,” or price crashes. The same flexibility naturally applies to the adversary: sustaining adversarial control is costly, so corrupt miners, too, may take a break. In summary, honest and adversarial nodes alike may fluctuate in their participation. We call this model *fully fluctuating participation*.¹

The recent work of Efron et al. [21] introduces a model of fully fluctuating participation and gives the first protocol provably secure in this model without resorting to PoW. Their protocol, however, replaces PoW with verifiable delay functions (VDFs) [8], which are proofs of sequential work assumed to require approximately the same time to compute for any node, honest or adversarial. Observe that a VDF is not only a cryptographic assumption, but also a strong and debatable hardware assumption: that the adversary cannot acquire or build any alternative hardware platform to speed up the VDF computation.

This leaves a fundamental question open: *can consensus under fully fluctuating participation be achieved without any hardware assumptions such as PoW or VDFs?*

In this work, we answer this open question affirmatively and present a consensus protocol secure under fully fluctuating participation. Our protocol assumes only a standard public key infrastructure (PKI) setup and a well-established cryptographic primitive called a verifiable random function (VRF) [41].

We also extend the fully fluctuating participation model to allow *uncorruption*, i.e., a corrupt node can return to honesty and restore the corruption budget. As with participation, Bitcoin already exhibits this. Once the adversary can no longer sustain control, the miner may simply resume work on the canonical chain as honest miners do. Our protocol cleanly extends to this more general setting with only minor changes and a mild additional assumption on the unpredictability of VRF outputs for liveness.

1.1 Technical Overview

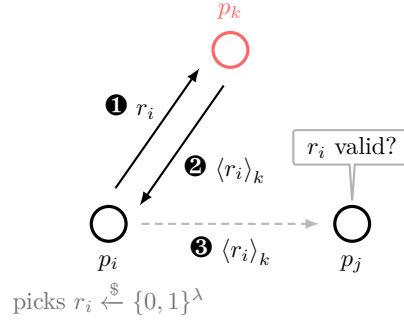
In the conventional adversarial model, a central adversary seizes the entire state of corrupt nodes, including their cryptographic secrets (private signing keys). This model immediately rules out the possibility of corrupt nodes taking a break or returning to honesty, since the adversary can indefinitely impersonate them using their signing keys. Prior works under this model must then restrict honest nodes not on break to always outnumber all nodes *ever* corrupted since the start of execution.

To enable corrupt nodes to take breaks, Efron et al. [21] introduce the *external adversary model*, which treats the node’s cryptographic secrets as external to the protocol, just like mining hardware in Bitcoin. This model is well justified because, in practice, corrupt nodes may perform actions on the adversary’s behalf but are unlikely to hand over their secret keys for fear of, e.g., being slashed or losing their cryptocurrency assets.

But making cryptographic secrets external is only the start. In Bitcoin, participation requires continuous use of hardware. Without binding a hardware resource as PoW does, participation with a PKI is *costless* and instantaneous with no restrictions on signing. Consequently, even though the external adversary cannot make corrupt nodes sign messages once they are on break, it can ask corrupt nodes to “pre-sign” messages for the future before they take a break. Honest nodes then cannot distinguish messages genuinely sent in the ongoing round from pre-signed ones.²

¹ Like most prior works in the literature, we assume a static membership set of eligible participants throughout and treat *reconfiguration* (i.e., membership changes over time) as an orthogonal problem.

² Additionally, corrupt nodes may “post-sign” messages for past rounds after returning from a break.



■ **Figure 1** Challenge-response between honest challenger p_i , adversarial prover p_k , and honest observer p_j (○ honest; ○ adversarial). ① p_i samples a fresh random nonce $r_i \in \{0, 1\}^\lambda$ (for security parameter λ) and sends it to p_k . ② p_k responds with a signature $\langle r_i \rangle_k$; since r_i was unpredictable to p_k in advance, a valid response proves that p_k was participating at the time of the challenge. ③ p_i may forward the proof $\langle r_i \rangle_k$ to p_j , but the proof is not *transferable*: p_j cannot verify that r_i was sampled freshly and at random by p_i , and so cannot conclude that p_k was participating.

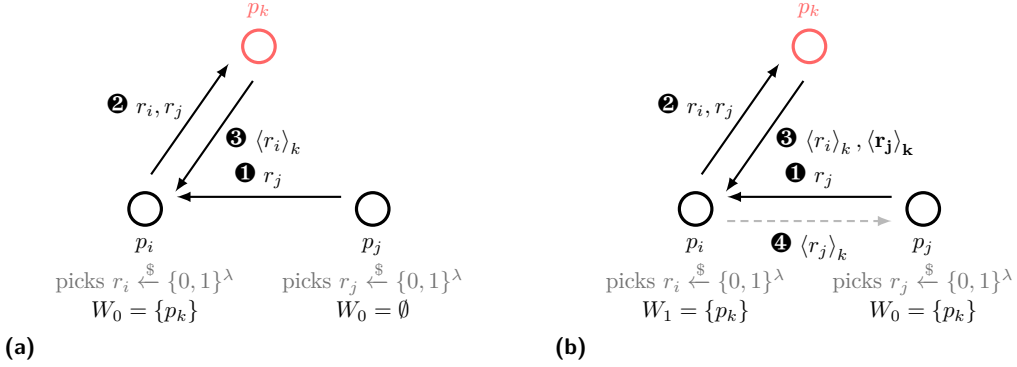
To distinguish *present* messages from pre-signed ones, consider an ideal randomness beacon that emits a random value at each round, and a protocol in which each node signs the ongoing round's beacon value alongside its message, thereby binding each message to the round. No node can predict future beacon values, so a corrupt node cannot pre-sign messages for rounds after it takes a break. Thus, honest nodes can ignore messages that do not carry a signature on the current beacon value. Unfortunately, realizing such an ideal beacon in fully fluctuating participation may be no easier than solving consensus itself, since nodes must agree on the randomness. Instead, Efron et al. [21] implement a randomness beacon using VDFs.

Our approach replaces the VDF with a simple protocol that achieves the same unpredictability *interactively*, where each node samples its own fresh randomness instead of consulting a beacon. A node p_i samples a fresh random nonce and sends it to a node p_k as a challenge; if p_k returns a signature on the nonce (Fig. 1), p_i concludes p_k is currently participating since p_k could not have pre-signed the response before it received the nonce. However, this verification is not *transferable*: p_i has no way to prove p_k 's participation to a third node p_j . So even when p_i accepts p_k 's messages, p_j may rightfully ignore them. By contrast, the freshness of an ideal randomness beacon or VDF output is verifiable by any node.

To add transferability to the verification of p_k 's participation, we additionally have node p_j send a random nonce to p_i , which p_i sends to p_k along with its own. In one scenario (Fig. 2a), p_k only signs p_i 's nonce, in which case we are in the prior example. However, p_k may now also sign p_j 's nonce (Fig. 2b), and p_i can meaningfully forward the signature to p_j . In the first scenario, only p_i knows that p_k is participating, so we say p_i considers p_k participating with *grade 0*. In the second, p_i additionally holds evidence that it can *transfer* to p_j , so p_k has *grade 1* for p_i and *grade 0* for p_j .

These grades resemble the hierarchy of distributed knowledge of Halpern and Moses [31]: at grade 0, a node p_i knows p_k is participating; at grade 1, p_i knows that p_j knows it too, *and*, therefore, that p_j will accept any messages p_i forwards from p_k . However, even if p_j

Prior works have addressed this issue by ignoring past messages and relying only on messages from the latest round [39, 15]. We inherit this approach.



■ **Figure 2** Two scenarios of two-grade challenge-response among honest nodes p_i, p_j and adversarial node p_k (○ honest; ○ adversarial). ① p_j samples $r_j \in \{0, 1\}^\lambda$ and sends it to p_i . ② p_i sends r_i, r_j to p_k . ③ In (a), p_k signs only r_i ; in (b), p_k signs both r_i and r_j . ④ p_i forwards what p_k signed to p_j , yielding different grade assignments in the two scenarios.

accepts a forwarded message from p_k , it cannot distinguish whether or not p_i forwarded the message to all honest nodes. At grade 2, then, p_i knows that p_j knows that all honest nodes will accept the message, allowing p_j to forward the message itself. With every additional grade, the message can be forwarded one further hop while remaining acceptable. We call this primitive *graded wakeness* (Sec. 3.1). It underpins our *2-grade graded agreement* protocol (Sec. 3.2, adapted from D’Amato et al. [16]), where each node’s input must be forwarded for two rounds, requiring graded wakeness with three grades. Chaining two instances of 2-grade graded agreement yields 3-grade graded agreement (Sec. 3.2), which we use to obtain consensus secure under fully fluctuating participation (Sec. 3.3 and Sec. 4).

Note that our focus in this work is to establish feasibility: that PoW and VDFs are not required for consensus with fully fluctuating participation. We leave improvements relevant to practical deployments, such as latency and message complexity, for future work.

2 Model Preliminaries

Nodes We now formalize the fully fluctuating participation model, which builds upon the model of Efron et al. [21]. We operate in the permissioned setting with a predefined set $\mathcal{P}$ of n nodes. We assume a public key infrastructure (PKI), so each node’s public key is known to all others. The adversary is a probabilistic polynomial-time (PPT) algorithm that can exert control over three axes—communication, sleepiness, and corruption—subject to various constraints introduced below.

Communication Time proceeds in lock-step *rounds*. We assume a *synchronous* network such that honest nodes at round t receive all messages sent to them from any prior round $t' < t$. (Note that consensus with sleepy nodes is *impossible* in partial synchrony [30, 44, 36].) For simplicity, one can assume messages sent to asleep nodes are buffered until they wake up. In practical systems, nodes do not need to buffer messages while asleep because they can retrieve relevant messages on demand from a peer-to-peer network after they wake up. The adversary sees every message sent over the network and controls the timing of message delivery, subject to the synchrony constraint.

Sleepiness The adversary can put nodes to *sleep* (i.e., make them take a break) in a *mildly adaptive* way, where it may only put nodes to sleep at the start of a round t . This reflects the model's intention that nodes consciously go to sleep at appropriate round boundaries. Sleeping mid-round without executing all protocol steps is corrupt behavior. *Asleep* nodes do not receive or send messages and do not execute the protocol. We denote the set of awake nodes in round t as $\mathcal{W}_t \subseteq \mathcal{P}$.

Corruption The adversary can adaptively corrupt nodes during a round t . We denote the set of nodes corrupt at any time in the round as $\mathcal{A}_t$. Nodes not corrupted by the adversary (called *honest*) follow the protocol when they are awake. We denote the set of nodes honest for the entire round as $\mathcal{H}_t = \mathcal{P} \setminus \mathcal{A}_t$. In Sec. 3, we consider an *immobile* adversary where corrupt nodes remain corrupt (i.e., $\mathcal{A}_t \subseteq \mathcal{A}_{t+1}$ for all t). In Sec. 4, we consider a *mobile* adversary that can uncorrupt previously corrupt nodes.

Uncorruption resets the node's protocol state to its initial state; the node is then asleep and upon waking behaves as if it woke up for the first time, receiving all messages sent to it in prior rounds. This definition is made quite natural by the external adversary model (discussed below) since signing and VRF keys are never handed to the adversary. Furthermore, the fully fluctuating sleepy model naturally resolves the question of post-uncorruption state since it already models a node waking up for the first time. Resetting the uncorrupted node's state ensures the node completely rids itself of the adversary's presence, whether that be from malware, bribery, or software bugs.

Schedules A *schedule* is defined as an infinite sequence of rounds of the adversary's choice of awake and corrupt nodes at each round. For rounds $0 \leq t_1 \leq t_2$ in a schedule, we define the following sets of nodes across the interval:

- $\mathcal{H}_{\mathcal{W}}(t_1, t_2) = \bigcap_{t_1 \leq t \leq t_2} \mathcal{H}_t \cap \mathcal{W}_t$ — nodes honest and awake at *every* round in $[t_1, t_2]$;
 - $\mathcal{A}_{\mathcal{W}}(t_1, t_2) = \bigcup_{t_1 \leq t \leq t_2} \mathcal{A}_t \cap \mathcal{W}_t$ — nodes corrupt and awake at *some* round in $[t_1, t_2]$.
- Note that $\mathcal{H}_{\mathcal{W}}(t_1, t_2)$ defines honest nodes that do *not* fluctuate in their participation over the interval.

We distinguish schedules of the adversary as *admissible* in the (T_r, T_s, ρ) -fully fluctuating sleepy model if for every round $t \geq 0$,

$$|\mathcal{H}_{\mathcal{W}}(t - T_r, t + T_s)| > \rho \cdot |\mathcal{A}_{\mathcal{W}}(t - T_r, t + T_s)|.$$

For $t < 0$, let $\mathcal{W}_t = \mathcal{W}_0$ and $\mathcal{H}_t = \mathcal{H}_0$. The two time parameters T_r and T_s are non-negative integers that adjust the interval sizes of the honest and corrupt sets; ρ is the minimum non-negative ratio allowed between these sets. The parameter T_r captures how many rounds an honest node must remain awake to *recover* from being asleep. The *stable* period parameter T_s specifies how long a node must remain honest and awake to meaningfully participate and count toward the honest set. A corrupt node counts toward the corrupt set if it is awake at any point during the stable period since it may pre-sign and post-sign messages to impersonate a stable node. In Def. 1, we define two named honest sets that will be useful.

► **Definition 1 (Stable Nodes).** For a round t , an honest node p is said to be *t -stable* if $p \in \mathcal{H}_{\mathcal{W}}(t, t + T_s)$, and *t -very-stable* if $p \in \mathcal{H}_{\mathcal{W}}(t - T_r, t + T_s)$.

Cryptographic Primitives and External Adversary Let λ be a security parameter. We make use of two cryptographic primitives: signatures and verifiable random functions. We model them in an idealized fashion using oracles $\mathcal{O}_S$ and $\mathcal{O}_V$ that are defined as follows:

1. Oracle $\mathcal{O}_S$ accepts queries of the form (p, m) from an *awake* node p with a message m . The oracle responds with signature $\langle m \rangle_p$ in the same round as the query, where $\langle m \rangle_p$ is shorthand for a signed message and is assumed to carry node p 's identity.
2. Oracle $\mathcal{O}_V$ accepts queries of the form (p, m) from an *awake* node p and responds with $(\text{VRF}_{\text{sk}}(m), \pi)$. Here, sk is the secret key of node p that is only known by $\mathcal{O}_V$.

In the external adversary model, nodes are not given secret keys to the cryptographic primitives used. Instead, each node p (honest or corrupt) can only issue queries to the oracles of the form $(p, \cdot)$ when awake. Under the PKI, any node may verify signatures and VRF outputs using the known verification algorithms.

Throughout, we will say a function $\text{NEGL}(\lambda)$ is *negligible* if for all $c > 0$, there exists a λ_0 such that $\text{NEGL}(\lambda) < \frac{1}{\lambda^c}$ for all $\lambda > \lambda_0$.

Byzantine Agreement Let $\mathcal{V}$ be some value space. We define consensus in this work as the Byzantine agreement problem in Def. 2.

► **Definition 2** (Byzantine Agreement). *Each node has an input value in $\mathcal{V}$ at round 0 and outputs a value in $\mathcal{V}$.*

- **Safety:** *If an honest node outputs x and an honest node outputs x' , then $x = x'$.*
- **Validity:** *If each honest node awake at round 0 inputs x , then no honest node outputs $x' \neq x$.*
- **Liveness:** *There exists a round t_0 such that for every $t \geq t_0$, every t -very-stable node outputs.*

3 Consensus in the Fully Fluctuating Sleepy Model

In this section, we give our base agreement protocol in the original external adversary model of Efron et al. [21] under fully fluctuating participation. As such, the adversary in this section cannot uncorrupt nodes. In Sec. 4, we extend the protocol to handle uncorruption. The protocol is largely the same; uncorruption comes with different liveness guarantees, so we begin with a complete analysis without uncorruption that captures the core ideas.

Our Byzantine agreement (BA) protocol proceeds in sequential *views* with each view calling an instance of our 3-grade graded agreement protocol GA3 (Sec. 3.2). Our 3-grade graded agreement protocol is built from two instances of our 2-grade graded agreement protocol GA2 (Sec. 3.2). The GA2 protocol is based on D'Amato et al. [16]'s construction, which we upgrade to tolerate fully fluctuating participation through the use of our graded wakeness protocol GW3 (Sec. 3.1). At the end of every view of BA, stable nodes participate in a VRF-based leader election to determine the input for the next view. In this work, we focus on single-shot agreement, but the structure closely resembles the total-order broadcast protocol of D'Amato et al. [16].

3.1 Graded Wakeness

Recall from Sec. 1.1 that graded wakeness allows nodes to filter for messages from awake nodes *and* ensure that other awake nodes accept such messages when forwarded. We represent this knowledge as *graded wakeness sets*, where nodes in the grade 1 set W_1^i are included in W_0^j for any stable node p_j . In the example of Fig. 2b, node p_i , with $W_1^i = \{p_k\}$, knows that p_j , with $W_0^j = \{p_k\}$, will accept messages originating from p_k . In Sec. 3.2, we show that three grades of graded wakeness suffice for the graded agreement protocol.

A node p_i executes the below steps starting from a round t_s and, at every round, forwards every received PING and ACK message to all nodes. Let t denote the rounds since t_s , where $t_s + t$ is the absolute round in the entire execution. During execution, node p_i stores R^t as the set of nonces received in PING messages by round t , including its own. A node p_j has *acknowledged* a nonce r once p_i receives $\langle \text{ACK}, r \rangle_j$. Only a node awake since round $t = 0$ executes the steps at rounds $t \geq 3$.

- $(t = 0)$: Sample nonce $R^0 \xleftarrow{\$} \{0, 1\}^\lambda$. Multicast $\langle \text{PING}, R^0 \rangle_i$.
- $(t = 3)$: Multicast $\langle \text{ACK}, r \rangle_i$ for every nonce $r \in R^3$.
- $(t = 4)$: Output W_2 as the nodes that acknowledged every nonce in R^2 .
- $(t = 5)$: Output W_1 as the nodes that acknowledged every nonce in R^1 .
- $(t = 6)$: Output W_0 as the nodes that acknowledged R^0 .

■ **Figure 3** Graded wakeness with 3 grades for a node p_i .

Def. 3 formally defines the desired properties of these sets, where the property just described is *graded delivery*. The last two properties naturally follow from what we expect from a wakeness set: that nodes asleep at the start of the protocol are excluded (*asleep exclusion*) and that stable nodes are included in all the sets (*honest inclusion*). Additionally, honest inclusion is “all-or-nothing,” such that a node honest since t_s is either in *all* of a t_s -stable node’s graded wakeness sets or in none of them.

► **Definition 3** (Graded Wakeness). *Parameterized by a start round t_s , each t_s -stable node p_i outputs $W_g^i \subseteq \mathcal{P}$ for each grade $g \in \{0, \dots, g_{\max} - 1\}$ with the following properties, except with negligible probability:*

- Graded Delivery: *For any t_s -stable node p_j and every $g \geq 1$, $W_g^i \subseteq W_{g-1}^j$.*
- Asleep Exclusion: *W_0^i includes only nodes awake after t_s .*
- Honest Inclusion: *$W_{g_{\max}-1}^i$ includes all t_s -stable nodes and every node in W_0^i that has remained honest since t_s .*

Note the unconventional notion that the protocol may start at any round $t_s \geq 0$. This captures the adversary’s ability to pre-sign messages *before* the protocol starts during long-running executions (such as during agreement).

We detail the graded wakeness protocol (GW3) for $g_{\max} = 3$ grades in Fig. 3. For simplicity, we reference the rounds of the protocol relative to its start time when it is clear from context, so round 0 of the protocol maps to round t_s in the entire execution.

Nodes start by sampling a *nonce* challenge and multicasting it alongside a signature from the oracle $\mathcal{O}_S$ in a PING message. The nonces are then forwarded for two rounds; in each round, nodes store the nonces received so far. At round 3, nodes *acknowledge* every received nonce by signing and multicasting it in an ACK message. From round 4 on, nodes begin to output the graded wakeness sets from highest to lowest. For grade 2 and a node p_i , if a node p_j has acknowledged all the nonces p_i received by round 2, then p_j is included in W_2^i . Node p_i then forwards the acknowledgments to all nodes. This continues until round 6, once nodes have output graded wakeness sets for every grade. As a result, graded delivery easily follows from the implication that nodes that acknowledge p_i ’s round 2 nonces also acknowledge all round 1 nonces of all other nodes that forward them to p_i . We now prove it precisely along with the other properties of Def. 3.

► **Lemma 4** (Graded Delivery). *For any t_s -stable nodes p_i and p_j , $W_2^i \subseteq W_1^j$ and $W_1^i \subseteq W_0^j$.*

Proof. Since $p_i, p_j \in \mathcal{H}_{\mathcal{W}}(t_s, t_s + 3)$, all nonces are forwarded from rounds 0 to 2, so $R^{0,j} \subseteq R^{1,i}$ and $R^{1,j} \subseteq R^{2,i}$. All nodes in W_2^i acknowledged all the nonces in $R^{2,i}$. Node p_i receives these acknowledgments by round 4, and they are forwarded such that node p_j receives them by round 5. Since $R^{1,j} \subseteq R^{2,i}$, all nodes in W_2^i acknowledged all the nonces in $R^{1,j}$, so $W_2^i \subseteq W_1^j$.

Similarly, all nodes in W_1^i acknowledged all the nonces in $R^{1,i}$. Node p_i receives these acknowledgments by round 5, and they are forwarded such that node p_j receives them by round 6. Since $R^{0,j} \subseteq R^{1,i}$, all nodes in W_1^i acknowledged all the nonces in $R^{0,j}$, so $W_1^i \subseteq W_0^j$. ◀

► **Lemma 5 (Asleep Exclusion).** *For any t_s -stable node p_i , W_0^i includes only nodes awake after t_s .*

Proof. Note that any node p_j that is not awake after t_s is not awake during the execution of the protocol. Such a node does not receive the nonce $R^{0,i}$. Therefore, except with probability $\text{poly}(n)/2^\lambda$, it cannot acknowledge $R^{0,i}$, so $p_j \notin W_0^i$. ◀

► **Lemma 6 (Honest Inclusion).** *For any t_s -stable node p_i , W_2^i includes all t_s -stable nodes and every node in W_0^i that has remained honest since t_s , i.e., $\mathcal{H}_{\mathcal{W}}(t_s, t_s + 6) \cup (W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)) \subseteq W_2^i$.*

Proof. Let p_j be a t_s -stable node that receives p_i 's nonce set $R^{2,i}$ by round 3. At round 3, node p_j acknowledges every nonce in $R^{2,i}$, and the acknowledgments are forwarded such that p_i receives them by round 4. Therefore, $p_j \in W_2^i$, so $\mathcal{H}_{\mathcal{W}}(t_s, t_s + 6) \subseteq W_2^i$.

For a node $p_j \in W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)$, p_j acknowledged p_i 's nonce $R^{0,i}$ at round 3. Node p_j must have also received p_i 's nonces $R^{2,i}$ and acknowledged them all, which p_i received by round 4. Thus, in round 4, p_i must have included p_j in W_2^i , and we have that $\mathcal{H}_{\mathcal{W}}(t_s, t_s + 6) \cup (W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)) \subseteq W_2^i$. ◀

► **Theorem 7.** *Fig. 3 implements graded wakeness with start round $t_s \geq 0$ and $g_{\max} = 3$ for schedules admissible in the fully fluctuating sleepy model with $T_s \geq 6$.*

Proof. With $T_s \geq 6$, every t_s -stable node p_i is in $\mathcal{H}_{\mathcal{W}}(t_s, t_s + 6)$. Therefore, every t_s -stable node outputs W_g^i for each grade g . We then have *graded delivery* by Lem. 4, *asleep exclusion* by Lem. 5, and *honest inclusion* by Lem. 6. ◀

3.2 Graded Agreement

We now outline how we adapt the graded agreement protocol from D'Amato et al. [16] with $g_{\max} = 2$ to tolerate fully fluctuating participation. Def. 8 gives our definition of graded agreement.

► **Definition 8 (Graded Agreement).** *Parameterized by a start round t_s , each node has an input value in $\mathcal{V} \cup \{\emptyset, \perp\}$, where $\perp$ indicates a default value, and $\emptyset$ indicates the node has no input. Each node outputs a single pair (x, g) where $x \in \mathcal{V} \cup \{\perp\}$ and $g \in \{0, \dots, g_{\max} - 1\}$.*

- **Graded Delivery:** *For any $g > 0$, if a t_s -stable node outputs (x, g) , then every t_s -stable node outputs (x, g') for $g - 1 \leq g' \leq g + 1$.*
- **Consistency:** *If an honest node outputs (x, g) for $g > 0$, then no honest node outputs $(x', \cdot)$ for $x' \neq x$.*
- **Validity:** *If every honest and awake node at round t_s with an input in $\mathcal{V}$ inputs the same x , then every t_s -stable node outputs $(x, g_{\max} - 1)$.*

- Integrity: If no honest and awake node at round t_s inputs $x \in \mathcal{V}$, then no honest node outputs $(x, \cdot)$.

Note the unconventional notion that honest nodes may have no input ($\emptyset$). This is used in our agreement protocol for nodes in *recovery* that have no input even though they are honest and awake at the start of the latest graded agreement. To get up to date, such nodes only observe the output of graded agreement.

First, recall the *time-shifted quorum* technique of Momose and Ren [42] that underlies the graded agreement protocol of D'Amato et al. [16]. Traditional agreement protocols rely on quorums of more than $n/2$ nodes, which any node can verify and forward, since no two such quorums can disagree. In the sleepy model, a node may never see that many nodes if a majority remain asleep, so it can only weigh a value against the inputs it actually receives.

Suppose, then, that a node accepts a value if more than half of the inputs it has received are for that value. Consider three disjoint sets of awake nodes P_1, P_2, P_3 of equal size at round t , where the number of awake nodes $n_t = |\mathcal{W}_t|$ is far less than n . If the nodes in P_3 are corrupt and send their inputs only to P_2 , then P_1 receives $n_t/3$ inputs while P_2 receives $2n_t/3$. A value carried by just over $n_t/6$ inputs then suffices for P_1 but not for P_2 , so these *subjective* quorums do not transfer. As with the challenge-response behind graded wakeness, the time-shifted quorum technique restores transferability.

After nodes send their input at the start of the protocol ($t = 0$), the time-shifted quorum technique proceeds as follows for a node p_i :

- For rounds $t = 1, 2, 3$, obtain the *inputters* $I^{t,i}$ as the set of nodes from which p_i receives inputs by round t .
- For rounds $t = 2, 3$, obtain the *equivocators* $E^{t,i}$ as the set of nodes from which p_i receives two distinct inputs by round t .

Throughout, nodes forward any received inputs to all nodes: for another node p_j , any inputs received by p_i by round t are guaranteed to be received by p_j by the next round. Therefore, p_j 's set of non-equivocating inputters at round 2 includes all of p_i 's non-equivocating inputters from rounds 1 to 3, i.e., $I^{1,i} \setminus E^{3,i} \subseteq I^{2,j} \setminus E^{2,j}$. This follows because p_i forwards its inputs at round 1, and p_j forwards any evidence of equivocation among those inputters in round 2. Let $I_x^{t,i}$ be the set of inputters in $I^{t,i}$ that sent input x ; then if p_i has a value x such that $|I_x^{1,i} \setminus E^{3,i}| > |I^{3,i}|/2$, it outputs x with grade 1, and p_j then outputs x with at least grade 0 since $|I_x^{2,j} \setminus E^{2,j}| > |I^{2,j}|/2$.

Upgrading the protocol to tolerate fully fluctuating participation then amounts to obtaining graded wakeness sets for every round of forwarding, which would be 3 graded wakeness sets for the above time-shifted quorum. The above output invariants now become: p_i outputs x with grade 1 if $|(I_x^{1,i} \setminus E^{3,i}) \cap W_2^i| > |I^{3,i} \cap W_0^i|/2$, and p_j then outputs x with grade 0 if $|(I_x^{2,j} \setminus E^{2,j}) \cap W_1^j| > |I^{2,j} \cap W_1^j|/2$. We can check that since $W_2^i \subseteq W_1^j \subseteq W_0^i$, we still have $(I_x^{1,i} \setminus E^{3,i}) \cap W_2^i \subseteq (I_x^{2,j} \setminus E^{2,j}) \cap W_1^j$ and $I^{2,j} \cap W_1^j \subseteq I^{3,i} \cap W_0^i$.

We give our 2-grade graded agreement protocol in Fig. 4. As in Sec. 3.1, we reference the rounds of the protocol relative to its start time t_s . Note that the time-shifted quorum just described maps exactly to rounds 4 to 6. We now prove that the protocol satisfies the properties of Def. 8 for $T_s \geq 6$ and $\rho \geq 1$ in the fully fluctuating sleepy model. Furthermore, we assume *very-stable input* as defined in Def. 9.

► **Definition 9** (Very-Stable Input). For a start round t_s , a graded agreement protocol is said to have *very-stable input* if every t_s -very-stable node has input $x \neq \emptyset$.

► **Lemma 10** (Graded Delivery). If a t_s -stable node p_i outputs $(x, 1)$, then every t_s -stable node p_j outputs (x, g') for $g' \leq 1$.

A node p_i with input x executes the below steps starting from a round t_s and, at every round, forwards every received IN message to all nodes. Let t denote the rounds since t_s , where $t_s + t$ is the absolute round in the entire execution. During execution, node p_i stores the sets I^t , I_y^t , and E^t at every round t where, for nodes $p_j \in \mathcal{P}$:

- *Inputters*: $p_j \in I^t$ if p_i has received $\langle \text{IN}, x \rangle_j$ for any $x \in \mathcal{V}$ by round t .
- *y-Inputters*: $p_j \in I_y^t$ if p_i has received $\langle \text{IN}, y \rangle_j$ by round t .
- *Equivocators*: $p_j \in E^t$ if p_i has received $\langle \text{IN}, x \rangle_j$ and $\langle \text{IN}, y \rangle_j$ for $x \neq y$ by round t .

$(t = 0)$: If $x \neq \emptyset$: multicast $\langle \text{IN}, x \rangle_i$.

$(t = 0, 1, 2, 3)$: Execute GW3.

$(t = 4)$: Store I^4 .

$(t = 5)$: Store I^5 and E^5 .

$(t = 6)$: If awake since round 0:

- If $\exists y \in \mathcal{V}$ such that $|(I_y^4 \setminus E^6) \cap W_2| > |I^6 \cap W_0|/2$: output $(y, 1)$.
- Else if $\exists y \in \mathcal{V}$ such that $|(I_y^5 \setminus E^5) \cap W_1| > |I^5 \cap W_1|/2$: output $(y, 0)$.
- Otherwise, output $(\perp, 0)$.

■ **Figure 4** Graded agreement with 2 grades for a node p_i .

Proof. Since p_i outputs $(x, 1)$, we have $|(I_x^{4,i} \setminus E^{6,i}) \cap W_2^i| > |I^{6,i} \cap W_0^i|/2$, and p_i has been awake since round 0. Thus, p_i forwards all IN messages in $I^{4,i}$ at round 4, so p_j receives them by round 5, giving $I^{4,i} \subseteq I^{5,j}$. For any sender in $I^{4,i}$ that p_j observes equivocating by round 5, p_j forwards the equivocation evidence by round 5, and p_i receives it by round 6, so $E^{5,j} \subseteq E^{6,i}$. Therefore, $I_x^{4,i} \setminus E^{6,i} \subseteq I_x^{5,j} \setminus E^{5,j}$. By graded delivery of GW3, we have $W_2^i \subseteq W_1^j$. Combined, we get

$$(I_x^{4,i} \setminus E^{6,i}) \cap W_2^i \subseteq (I_x^{5,j} \setminus E^{5,j}) \cap W_1^j. \quad (1)$$

Similarly, p_j forwards all values in $I^{5,j}$ such that p_i receives them by round 6, i.e., $I^{5,j} \subseteq I^{6,i}$, and, by graded delivery of GW3, we have $W_1^j \subseteq W_0^i$. Combined, we get

$$I^{5,j} \cap W_1^j \subseteq I^{6,i} \cap W_0^i. \quad (2)$$

Therefore, by Equations (1) and (2), $|(I_x^{4,i} \setminus E^{6,i}) \cap W_2^i| > |I^{6,i} \cap W_0^i|/2 \geq |I^{5,j} \cap W_1^j|/2$, so the grade 0 condition holds at p_j for x . Applying the same argument as above with p_j in place of p_i shows that if p_j outputs with grade 1, then its value meets p_j 's grade 0 condition as well. Since the grade 0 condition requires more than half of $|I^{5,j} \cap W_1^j|$, only one value meets it, so node p_j outputs (x, g') for $g' \leq 1$. ◀

► **Lemma 11 (Consistency).** *If an honest node p_i outputs $(x, 1)$, then no honest node p_j outputs $(x', \cdot)$ for $x' \neq x$.*

Proof. Nodes p_i and p_j will only output if they have been awake since the start of the protocol, so both lie in $\mathcal{H}_W(t_s, t_s + 6)$. Thus, by graded delivery (Lem. 10), p_j will not output $(x', \cdot)$ for $x' \neq x$. ◀

For the following proofs of validity and integrity, we use Lem. 12. Let $\mathcal{I}$ be the set of honest and awake nodes at round t_s whose input is not $\emptyset$.

► **Lemma 12** ($\mathcal{I}$ -Majority). *Assuming very-stable input (Def. 9), every t_s -stable node p_i has*

$$|(\mathcal{I} \cap W_2^i) \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)| > |I^{6,i} \cap W_0^i|/2.$$

Proof. We first bound the number of inputs p_i receives in $I^{6,i} \cap W_0^i$. Nodes in $\mathcal{I}$ are honest and awake at round 0, so they send an input, which p_i receives by round 6 in $I^{6,i}$. By asleep exclusion of GW3, every node in W_0^i is awake after t_s , so any other node in W_0^i that could send an input is corrupt and awake during the protocol; so $(\mathcal{I} \cap W_0^i) \cup \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)$ includes all nodes in W_0^i that could send an input by round 6, giving us:

$$|I^{6,i} \cap W_0^i| \leq |(\mathcal{I} \cap W_0^i) \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)| + |\mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)|, \quad (3)$$

where the left summand indicates the set of honest inputs and the right indicates the set of corrupt inputs.

We now lower bound the number of inputs in $(\mathcal{I} \cap W_2^i) \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)$. Under the assumption of very-stable input (Def. 9), we have that $\mathcal{I}$ includes all t_s -very-stable nodes, and by honest inclusion of GW3, we have that W_2^i includes all t_s -stable nodes. Further, by graded delivery of GW3, we have $W_2^i \subseteq W_0^i$, and by honest inclusion, we have $W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6) \subseteq W_2^i$; therefore, it follows that W_2^i and W_0^i include the same set of honest nodes, i.e., $W_2^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6) = W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)$. Therefore, by the assumption that we are in a schedule admissible in the fully fluctuating sleepy model, we have

$$|\mathcal{I} \cap (W_0^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6))| \geq |\mathcal{H}_{\mathcal{W}}(t_s - T_r, t_s + T_s)| > |\mathcal{A}_{\mathcal{W}}(t_s, t_s + 6)|,$$

since $\rho \geq 1$ and $T_s \geq 6$. Finally, by Equation (3), it follows that $|\mathcal{I} \cap (W_2^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6))| > |I^{6,i} \cap W_0^i|/2$. ◀

► **Lemma 13** (Validity). *If every node in $\mathcal{I}$ inputs x , then every t_s -stable node p_i outputs $(x, 1)$.*

Proof. Nodes in $\mathcal{I}$ are honest and awake at round 0, so they send x , which p_i receives by round 4 in $I_x^{4,i}$. For the nodes in W_2^i that remain honest until round 6, they do not equivocate; by Lem. 12 ($\mathcal{I}$ -majority), we get $|(I_x^{4,i} \setminus E^{6,i}) \cap W_2^i| \geq |\mathcal{I} \cap (W_2^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6))| > |I^{6,i} \cap W_0^i|/2$, and it follows that p_i outputs $(x, 1)$. ◀

► **Lemma 14** (Integrity). *If no node in $\mathcal{I}$ inputs $x \in \mathcal{V}$, then no honest node p_i outputs (x, g) for every $g \in \{0, 1\}$.*

Proof (of Lem. 14). Note that for p_i to output, it must be a stable node. For the sake of contradiction, assume node p_i outputs $(x, 1)$, implying it received more than $|I^{6,i} \cap W_0^i|/2$ inputs of x , i.e., $|(I_x^{4,i} \setminus E^{6,i}) \cap W_2^i| > |I^{6,i} \cap W_0^i|/2$. By Lem. 12 ($\mathcal{I}$ -majority), we have $|\mathcal{I} \cap (W_2^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6))| > |I^{6,i} \cap W_0^i|/2$. So, for p_i to output $(x, 1)$, it must have received x from a node in $\mathcal{I} \cap (W_2^i \setminus \mathcal{A}_{\mathcal{W}}(t_s, t_s + 6))$. However, no node in $\mathcal{I}$ inputs x , so such a node would have to equivocate to send x if it is corrupt after round 0, contradicting that the input is in $I_x^{4,i} \setminus E^{6,i}$. Thus, no honest node outputs $(x, 1)$. The proof is identical for grade 0, so we omit it. ◀

► **Theorem 15.** *Let $t_s \geq 0$ be the start round. If every t_s -very-stable node has input $x \neq \emptyset$ (very-stable input, Def. 9), then the protocol in Fig. 4 implements graded agreement with start round t_s and $g_{\max} = 2$ for schedules admissible in the fully fluctuating sleepy model with $T_s \geq 6$ and $\rho \geq 1$.*

Proof. Since $T_s \geq 6$, each t_s -stable node $p_i \in \mathcal{H}_{\mathcal{W}}(t_s, t_s + 6)$, so we have *graded delivery* by Lem. 10, *consistency* by Lem. 11, *validity* by Lem. 13, and *integrity* by Lem. 14. ◀

A node p_i with input x executes the below steps starting from a round t_s . Let t denote the rounds since t_s , where $t_s + t$ is the absolute round in the entire execution.

- $(t = 0 \dots 6)$: Execute GA2_0 with input x .
- $(t = 6)$: Let $(y_0, g_0) \leftarrow$ the output of GA2_0 if it exists; otherwise, let $(y_0, g_0) \leftarrow (\emptyset, 0)$.
- $(t = 6 \dots 12)$: Execute GA2_1 with input y_0 .
- $(t = 12)$: If awake since round 0:
 - Let $(y_1, g_1) \leftarrow$ the output of GA2_1 .
 - If $g_0 = 1$: output $(y_0, 2)$.
 - Otherwise, output (y_1, g_1) .

■ **Figure 5** Graded agreement with 3 grades for a node p_i .

3.2.1 Graded Agreement with 3 Grades

We build 3-grade graded agreement from two sequential instances of 2-grade graded agreement, GA2_0 and GA2_1 , as shown in Fig. 5. A node without an output from GA2_0 inputs $\emptyset$ into GA2_1 . We prove that the protocol satisfies the properties of Def. 8 with $g_{\max} = 3$ for $T_r \geq 6$, $T_s \geq 12$, and $\rho \geq 1$ in the fully fluctuating sleepy model, again assuming very-stable input (Def. 9). Since $T_r \geq 6$, every t_s -very-stable node for GA2_1 's start round 6 has been awake since round 0; it therefore has an output $y_0 \neq \emptyset$ from GA2_0 that it inputs into GA2_1 , so GA2_1 also has very-stable input.

► **Lemma 16** (Graded Delivery). *For any $g > 0$, if a t_s -stable node p_i outputs (x, g) , then every t_s -stable node p_j outputs (x, g') for $g - 1 \leq g' \leq g + 1$.*

Proof. Note that a node only outputs if it has been awake since round 0, so it is t_s -stable for both GA2_0 and GA2_1 . First, suppose $g = 2$, so that p_i outputs $(x, 1)$ from GA2_0 . By graded delivery of GA2_0 , every t_s -stable node p_j outputs (x, g_0) from GA2_0 for $g_0 \leq 1$, and, by consistency of GA2_0 , no honest node outputs a value other than x from GA2_0 . Every honest and awake node at round 6 with an input $\neq \emptyset$ therefore inputs x into GA2_1 , so by validity of GA2_1 , node p_j outputs $(x, 1)$ from GA2_1 . Node p_j then outputs $(x, 2)$ if $g_0 = 1$ and $(x, 1)$ otherwise, both within $g - 1 \leq g' \leq g + 1$.

Now suppose $g = 1$, so that p_i outputs $(x, 1)$ from GA2_1 . By graded delivery of GA2_1 , every t_s -stable node p_j outputs (x, g_1) from GA2_1 for $g_1 \leq 1$, so p_j outputs $(x, 1)$ or $(x, 0)$ unless it obtains grade 1 from GA2_0 . In the latter case, p_j outputs $(y_0, 2)$ where $(y_0, 1)$ is its output from GA2_0 ; by consistency of GA2_0 , every honest and awake node at round 6 with an input $\neq \emptyset$ inputs y_0 into GA2_1 , so by integrity of GA2_1 , no honest node outputs a value other than y_0 from GA2_1 , giving $y_0 = x$. Every t_s -stable node p_j thus outputs (x, g') for $g' \leq 2$, again within $g - 1 \leq g' \leq g + 1$. ◀

► **Lemma 17** (Consistency). *If an honest node p_i outputs (x, g) for $g > 0$, then no honest node p_j outputs $(x', \cdot)$ for $x' \neq x$.*

Proof. Nodes output the value of their GA2_0 output at grade 2 and the value of their GA2_1 output at grades 1 and 0. First, suppose some honest node outputs $(y, 1)$ from GA2_0 . By consistency of GA2_0 , no honest node outputs a value other than y from GA2_0 , so every honest and awake node at round 6 with an input $\neq \emptyset$ inputs y into GA2_1 . By validity of GA2_1 , every t_s -stable node outputs $(y, 1)$ from GA2_1 , so both branches only produce the value y , and $x = x' = y$. Otherwise, no honest node outputs grade 1 from GA2_0 , so no honest node

outputs grade 2; since p_i and p_j only output because they have been awake since round 0, both are t_s -stable for both GA2_0 and GA2_1 and therefore obtain outputs from GA2_1 , with p_i obtaining x from grade 1 of GA2_1 and p_j obtaining x' from GA2_1 ; by consistency of GA2_1 , we have $x = x'$. $\blacktriangleleft$

► **Lemma 18** (Validity). *If every honest and awake node at round t_s with an input $\neq \emptyset$ inputs x , then every t_s -stable node p_i outputs $(x, 2)$.*

Proof. By validity of GA2_0 , every t_s -stable node p_i outputs $(x, 1)$ from GA2_0 . Therefore, node p_i has $g_0 = 1$ and outputs $(x, 2)$. $\blacktriangleleft$

► **Lemma 19** (Integrity). *If no honest and awake node at round t_s inputs x , then no honest node p_i outputs $(x, \cdot)$.*

Proof. Node p_i outputs $(x, 2)$ only with an output $(x, 1)$ from GA2_0 ; by integrity of GA2_0 , no honest node outputs x from GA2_0 , so no honest node outputs $(x, 2)$. Node p_i outputs $(x, 1)$ or $(x, 0)$ only with an output $(x, \cdot)$ from GA2_1 . Recall that every honest node with an input $\neq \emptyset$ into GA2_1 inputs its output from GA2_0 ; since no honest node outputs x from GA2_0 , no honest and awake node at round 6 inputs x into GA2_1 . By integrity of GA2_1 , no honest node outputs x from GA2_1 , so no honest node outputs $(x, 1)$ or $(x, 0)$. $\blacktriangleleft$

► **Theorem 20.** *Let $t_s \geq 0$ be the start round. If every t_s -very-stable node has input $x \neq \emptyset$ (very-stable input, Def. 9), then the protocol in Fig. 5 implements graded agreement (Def. 8) with start round t_s and $g_{\max} = 3$ for schedules admissible in the fully fluctuating sleepy model with $T_r \geq 6$, $T_s \geq 12$, and $\rho \geq 1$.*

Proof (of Thm. 20). Since $T_r \geq 6$ and $T_s \geq 12$, both GA2_0 and GA2_1 have very-stable input and implement graded agreement by Thm. 15. We then have *graded delivery* by Lem. 16, *consistency* by Lem. 17, *validity* by Lem. 18, and *integrity* by Lem. 19. $\blacktriangleleft$

3.3 Byzantine Agreement

Our agreement protocol in Fig. 6 follows the phase-king approach [6]. The protocol proceeds in a series of views, each calling an instance of GA3 from Sec. 3.2. For a view $v \geq 0$, we denote GA3_v as the 3-grade graded agreement instance of view v .

For the starting view 0, awake nodes pass their initial input (called the *candidate* input) to GA3_0 . For a node that remains awake, it updates its candidate input to the output value of GA3_0 . If the grade is at least 1, it also locks on that value; otherwise, it releases its lock. If the grade is 2, it outputs the value in the agreement protocol (but continues to participate to help other nodes). It then multicasts and signs its candidate input in $\mathcal{V}$, in a **PROPOSE** message, along with its VRF output and proof over the next view number. Such a proposal always exists, since at least one t_s -stable node exists by admissibility with $\rho \geq 1$ and proposes a candidate in $\mathcal{V}$.

At the end of the view, nodes without a lock update their candidate input to the proposal with the highest VRF output for the view. This input is then passed on to the next GA3 instance. The adversary may pre-compute VRF outputs for many future rounds and then cherry-pick the corrupt node with the highest VRF output to be awake for the corresponding view. This hurts the latency of the agreement protocol but does not affect safety, liveness, or validity.

We prove that the agreement protocol satisfies the properties of Def. 2 for schedules admissible in the $(T_r = 13, T_s = 13, \rho = 1)$ -fully fluctuating sleepy model. First, we show in Lem. 21 that each GA3_v satisfies the very-stable input condition (Def. 9) for Thm. 20.

A node p_i with input x starts execution at round 0 and executes the below steps at the start of the round $13v$ for every view $v \geq 0$. Beyond the outputs of GW and GA, p_i maintains the following state variables with the following initial values:

- *Candidate:* $x_C \leftarrow x$.
- *Lock:* $x_L \leftarrow \perp$.

$(t = 0 \dots 12)$: Execute GA3_v with input x_C .

$(t = 12)$: If awake since the beginning of view v :

- Let $(y, g) \leftarrow$ the output of GA3_v ; if $y \neq \perp$, set $x_C \leftarrow y$.
- If $g \geq 1$: set $x_L \leftarrow y$; otherwise, set $x_L \leftarrow \perp$.
- If $g = 2$ and not yet output: output y .
- Multicast $\langle \text{PROPOSE}, (x_C, \text{VRF}_{\text{sk}}(v+1), \pi) \rangle_i$.

$(t = 13)$:

- If $x_L = \perp$: set $x_C \leftarrow$ the proposed value in $\mathcal{V}$ with the highest value VRF output for view $v+1$.
- If not awake since the beginning of the view: set $x_C \leftarrow \emptyset$.

Note that $t = 13$ of view v occurs alongside $t = 0$ of view $v+1$.

■ **Figure 6** Byzantine agreement in the fully fluctuating sleepy model.

► **Lemma 21** (Very-Stable Input). *For every view $v \geq 0$, each $13v$ -very-stable node has an input $x_C \in \mathcal{V}$.*

Proof (of Lem. 21). For the very first view $v = 0$, each 0-very-stable node has input $x_C \in \mathcal{V}$ for GA3_0 . For views $v > 0$, each $13v$ -very-stable node p_i has been awake since round $13v - 13$, making it $13(v-1)$ -stable at the start of GA3_{v-1} . By definition, GA3_{v-1} outputs value $y \in \mathcal{V} \cup \{\perp\}$, and p_i only sets its candidate x_C to y if $y \neq \perp$, so $x_C \in \mathcal{V}$. If p_i holds no lock, it replaces its candidate with the proposal of the highest VRF output; at least one such proposal exists, since p_i is itself $13(v-1)$ -stable and multicasts its own proposal at the end of view $v-1$. Every proposal carries a value in $\mathcal{V}$, so each $13v$ -very-stable node, then, has an input $x_C \in \mathcal{V}$. ◀

► **Lemma 22** (Totality). *If an honest node p_i outputs y during view v , then for every view $v' > v$, every node in $\mathcal{H}_{\mathcal{W}}(13v', 13v' + 12)$ outputs y if it has not output already.*

Proof (of Lem. 22). Since p_i outputs y during view v , it obtains $(y, 2)$ from GA3_v . We first claim that, for any view $v' > v$, if some honest node obtains $(y, 2)$ from $\text{GA3}_{v'-1}$, then every node in $\mathcal{H}_{\mathcal{W}}(13v', 13v' + 12)$ obtains $(y, 2)$ from $\text{GA3}_{v'}$. By graded delivery of $\text{GA3}_{v'-1}$, every node in $\mathcal{H}_{\mathcal{W}}(13(v'-1), 13(v'-1) + 12)$ obtains (y, g) from $\text{GA3}_{v'-1}$ for $g \geq 1$. Every such node therefore sets its candidate to $x_C = y$ and locks on $x_L = y$, so it keeps $x_C = y$ at round $13v'$, since a node holding a lock does not adopt a proposal. Any node honest and awake at round $13v'$ that was not awake throughout view $v'-1$ resets its candidate to $\emptyset$. Hence every honest and awake node at round $13v'$ with an input $\neq \emptyset$ inputs y into $\text{GA3}_{v'}$, and by validity of $\text{GA3}_{v'}$, whose very-stable input condition holds by Lem. 21, every $\mathcal{H}_{\mathcal{W}}(13v', 13v' + 12)$ node obtains $(y, 2)$. The lemma then follows by induction on $v' > v$: for $v' = v+1$, the claim applies with p_i as the honest node, and for $v' > v+1$, it applies with any node in $\mathcal{H}_{\mathcal{W}}(13v', 13v' + 12)$, of which at least one exists since the schedule is admissible with $\rho \geq 1$, and which obtains $(y, 2)$ by the induction hypothesis. ◀

► **Lemma 23** (Validity). *If each honest node awake at round 0 inputs x , then any node that outputs also outputs x .*

Proof. Since all honest nodes awake at round 0 have input x , all 0-very-stable nodes input $x_C = x$ into GA3_0 . By validity of GA3 , every 0-stable node obtains $(x, 2)$ from GA3_0 . Therefore, all 0-stable nodes output x , and by Lem. 24 (safety), any other honest node that outputs also outputs x . ◀

► **Lemma 24** (Safety). *For two honest nodes p_i and p_j that output x and x' at rounds t and t' , respectively, $x = x'$.*

Proof. Without loss of generality, assume $t \leq t'$. Then we have two cases: they output in the same view ($t = t'$), or p_j outputs in a later view ($t < t'$). For $t = t'$, we directly get agreement since no two honest nodes can output two different values with grade 2 by consistency of GA3 . For $t < t'$, let views v and v' be the views in which p_i and p_j output, respectively. Since p_j outputs in view v' , it is in $\mathcal{H}_W(13v', 13v' + 12)$, as only such nodes obtain a grade 2 output. Lem. 22 then gives that p_j outputs x , so $x' = x$. ◀

► **Lemma 25** (Liveness). *There exists a round t_0 such that for every $t \geq t_0$, every t -very-stable node outputs.*

Proof. Any t -very-stable node with t in view v satisfies $[t - T_r, t + T_s] \supseteq [13v, 13v + 12]$, so it is $13v$ -stable for view v . We call a view v an *honest-leader view* if the proposal with the highest VRF output for view $v + 1$ is from a node p_ℓ that is $13v$ -stable. We first show that if v is an honest-leader view, then every $13(v + 1)$ -stable node outputs in view $v + 1$.

Let x be the candidate that p_ℓ proposes, which is its output value from GA3_v unless that value is $\perp$, in which case it is the candidate p_ℓ already held; either way $x \in \mathcal{V}$. A $13v$ -stable node that holds no lock adopts p_ℓ 's proposal, so its candidate is x . A $13v$ -stable node holds a lock only if it obtains grade at least 1 from GA3_v , and then graded delivery of GA3_v gives every $13v$ -stable node the same output value; since p_ℓ is one of them, that value is x , which the node keeps as its candidate. Either way every $13v$ -stable node has candidate x at round $13(v + 1)$, while a node honest and awake at that round but not awake throughout view v sets its candidate to $\emptyset$. Every honest and awake node at round $13(v + 1)$ with an input $\neq \emptyset$ therefore inputs x into GA3_{v+1} , so by validity of GA3_{v+1} , every $13(v + 1)$ -stable node obtains $(x, 2)$ and outputs x in view $v + 1$.

We now show that an honest-leader view v_0 occurs within $n\lambda$ views except with negligible probability. Fix a view v with $t_s = 13v$. Since the schedule is admissible with $\rho \geq 1$, at least one t_s -stable node exists in view v . The adversary may cherry-pick any corrupt nodes since the beginning of the execution for the highest VRF output. Therefore, there are $|\mathcal{H}_W(t_s, t_s + 12) \cup \mathcal{A}_W(0, t_s + 13)|$ possible proposals in the view, making the probability that an honest node's proposal has the highest VRF output be

$$\frac{|\mathcal{H}_W(t_s, t_s + 12)|}{|\mathcal{H}_W(t_s, t_s + 12)| + |\mathcal{A}_W(0, t_s + 13)|}.$$

In the worst case, only one node is honest and awake, and the remaining nodes are corrupt and asleep. Since VRF outputs are independent and uniform, the highest output is equally likely to belong to any node in $\mathcal{P}$. The probability that view v is an honest-leader view is then at least $1/n$. Since the VRF outputs for each view are sampled anew, with the view number as the oracle input, this bound holds for a view regardless of the adversary's choices in the preceding views. The probability that none of $n\lambda$ consecutive views is an honest-leader view is therefore at most $(1 - 1/n)^{n\lambda}$, which is negligible in λ .

Let v_0 be the first honest-leader view and set $t_0 = 13(v_0 + 1)$. Since the schedule is admissible with $\rho \geq 1$, at least one $13(v_0 + 1)$ -stable node exists, so honest nodes indeed output in view $v_0 + 1$. Any t -very-stable node with $t \geq t_0$ lies in some view $v \geq v_0 + 1$ and is $13v$ -stable. If $v = v_0 + 1$, it outputs by the implication above for the honest-leader view v_0 . If $v > v_0 + 1$, it outputs by Lem. 22, since honest nodes output in view $v_0 + 1$. ◀

► **Theorem 26.** *The protocol in Fig. 6 implements Byzantine agreement (Def. 2) for schedules admissible in the $(T_r = 13, T_s = 13, \rho = 1)$ -fully fluctuating sleepy model.*

Proof. Since each view $v \geq 0$ has very-stable input (Def. 9) by Lem. 21, $T_r \geq 6$, $T_s \geq 12$, and $\rho \geq 1$, each GA3_v instance implements graded agreement by Thm. 20. Thus, we have *safety* by Lem. 24, *validity* by Lem. 23, and *liveness* by Lem. 25. ◀

4 Consensus in the Fully Fluctuating Sleepy Model with Uncorruption

This section presents a discussion of our agreement protocol in Sec. 3.3 under an adversary that can uncorrupt corrupt nodes. Uncorruption raises two challenges. The first concerns both safety and liveness of the agreement protocol but is resolved with a minor change to GW3 and GA2. The second addresses how the adversary may bias the VRF-based leader election, which requires a mild additional assumption to guarantee liveness.

4.1 Challenge 1: Pre-Signed Equivocation

In the previous setting of Sec. 3, honest nodes are assumed to always have a “clean slate,” since a node made corrupt in a prior round remains corrupt. With uncorruption, no node is guaranteed a clean slate. In graded agreement, therefore, an honest node can appear as an equivocator in a view when its past corrupt-self pre-signed input messages for the view. Honest and awake nodes are then unable to distinguish honest inputs from the pre-signed ones. The graded agreement protocol loses validity since non-equivocating inputs may not include all honest inputs, so consequently, the agreement protocol loses safety and liveness.

The liveness violation follows since a set of non-equivocating inputters may never form a majority for a single input in GA2 and subsequently in GA3, so no node may ever output from BA. Safety is subtler: suppose an honest node decides x in some view, causing the other honest and awake nodes to lock on x . Then, in the next view, every honest inputter may be considered an equivocator, so GA3 outputs $(\perp, 0)$ and honest nodes release their locks on x . The adversary may then win the next leader election and propose $y \neq x$, which the honest nodes adopt and decide, violating agreement.

Graded wakeness already solves the problem of distinguishing the latest messages from pre-signed ones. We simply have GW3 take in a value that nodes attach alongside their nonce acknowledgments. Fig. 7 shows the changes in red. The graded agreement protocol then passes its input to GW3 and is otherwise unchanged, except that a node counts as an inputter or an equivocator only for a value it acknowledged in GW3, at the grade already used for that round in Fig. 4. A pre-signed input has no fresh nonce acknowledgment, so it counts neither as an input nor as equivocation evidence. With these changes, the proofs of validity and integrity (Lems. 13 and 14) remain unchanged since, except with negligible probability, no corrupt node can pre-sign nonce-value pairs. Equivocation evidence remains transferable due to graded delivery of graded wakeness, so the proofs of graded delivery and consistency (Lems. 10 and 11) also remain unchanged.

A node p_i with value x executes the below steps starting from a round t_s . A node p_j has *acknowledged* a nonce r with value x' once p_i receives $\langle \text{ACK}, r, x' \rangle_j$.

- $(t = 0)$: Sample nonce $R^0 \xleftarrow{\$} \{0, 1\}^\lambda$. Multicast $\langle \text{PING}, R^0 \rangle_i$.
- $(t = 3)$: Multicast $\langle \text{ACK}, r, x \rangle_i$ for every nonce $r \in R^3$.
- $(t = 4)$: Output W_2 as the **node-value pairs** that acknowledged every nonce in R^2 .
- $(t = 5)$: Output W_1 as the **node-value pairs** that acknowledged every nonce in R^1 .
- $(t = 6)$: Output W_0 as the **node-value pairs** that acknowledged R^0 .

■ **Figure 7** Graded wakeness with 3 grades for a node p_i with value x . Text in red indicates the changes to Fig. 3 that attach a node's value to each nonce acknowledgment.

4.2 Challenge 2: Biased Leader Election

Uncorruption introduces a significant source of bias in the VRF-based leader election. In the previous setting, the adversary could pull from its set of corrupt nodes since the beginning of execution as possible candidates for the leader election. With uncorruption, the adversary could theoretically pre-compute VRF outputs for every node in $\mathcal{P}$ by corrupting and uncorrupting each node. Therefore, the adversary could predict the outcome of every leader election.

In this extreme case, the protocol cannot guarantee liveness, so some additional assumption is needed. In essence, any assumption that guarantees an eventual honest leader suffices. We formalize this using the notion of an *unpredictable* node in Def. 27.

► **Definition 27** (Unpredictable Nodes). *A node p_i is said to be unpredictable for a view $v > 0$ if it is $13(v - 1)$ -stable and has never queried the VRF oracle $\mathcal{O}_V$ for view v before round $13(v - 1)$.*

The adversary does not know whether an unpredictable node in a view will win the leader election or not. Unpredictability is a mild assumption in practice, since it is extremely unlikely that the adversary has obtained every node's VRF output before the election. Some nodes might simply have never been corrupted, so they remain unpredictable. Other nodes may be more reluctant to share VRF outputs with the adversary than signatures, since VRF outputs decide leader election, so the adversary obtains fewer of them. Assuming such an unpredictable node occurs often enough, liveness is guaranteed. Thm. 28 makes this precise.

► **Theorem 28.** *The protocol in Fig. 6, with GW3 and GA2 modified as in Sec. 4.1, implements Byzantine agreement (Def. 2) for schedules that:*

- *are admissible in the $(T_r = 13, T_s = 13, \rho = 1)$ -fully fluctuating sleepy model; and*
- *have at least $n\lambda$ views $v > 0$ for which some node is unpredictable (Def. 27).*

The proof of liveness follows as outlined in Lem. 25 because, except with negligible probability, eventually some unpredictable node will have the highest VRF output.

5 Related Work

Pass and Shi [46] formalize the sleepy model. Their original sleepy model assumes a fixed set of corrupt nodes that do not fluctuate. The model has attracted significant attention, with many variants [14, 33, 18, 4] of the longest-chain approach of Nakamoto [43], as well as works improving latency [42, 25, 22] and providing censorship resistance [23].

The original sleepy model was extended in subsequent works [39, 15, 16] to allow for the corrupt set to grow, but not shrink, so these protocols are not fully fluctuating. Recent work [49] gives tight upper and lower bounds on communication complexity in this model. Efron et al. [21] introduce the external adversary model, and, like PoSAT [19], utilize VDFs to achieve consensus under fully fluctuating participation. Farahbakhsh et al. [24] improve on the latency and security of fully permissionless PoW consensus.

Reconfigurable consensus [34, 32, 38, 48, 7, 20, 45] also deals with participation changes over time but is not to be confused with the sleepy model. Reconfiguration changes the set of nodes that *can* participate in the protocol, which is orthogonal to a majority of the nodes being asleep at any given time. The sleepy model allows for fluctuations in terms of which nodes *do* participate. Only a few works consider both reconfigurations and sleepy nodes [4, 11, 14]. Lewis-Pye and Roughgarden [37] show that any protocol in the reconfiguration setting must make use of non-time-malleable cryptography (e.g., VDFs). Note that this result does not apply to our protocol since our model does not include reconfiguration.

Algorand [12] achieves consensus with *player replaceability*, where each step of the protocol is executed by a different committee. Algorand’s committees are subsampled from the membership set via VRF-based sortition [41]. Algorand relies on a fixed threshold of honest and awake nodes, so it does not tolerate adversarially controlled sleepiness. It requires a super-majority of honest and awake nodes, which more closely resembles traditional models. Subsequent works [29, 13, 17] extend player replaceability to MPC and also require a super-majority of honest and awake nodes.

Works on mixed fault models [28, 40, 1, 26, 2] consider a combination of crash and Byzantine faults simultaneously, but do not consider scenarios in which a majority of nodes may crash. Attiya et al. [3] implement storage primitives under *churn*, where nodes continually enter and leave an asynchronous system. They require a known threshold on the churn rate, so they also do not allow an arbitrary number of nodes to leave or crash at once.

The mobile Byzantine fault model, formalized by Garay [27], captures an adversary that can reallocate its corruption budget across nodes, potentially *curing* (i.e., uncorrupting) previously corrupt nodes. The literature divides into constrained mobility, where the adversary moves with messages [10], and unconstrained mobility, where the adversary moves between rounds [27]. Another key axis is whether the cured node can detect that it was corrupt [27, 5] or may inadvertently continue sending corrupt messages [47, 9]. Our work more closely resembles the setting of unconstrained mobility with corruption detection. Intuitively, the fully fluctuating sleepy model is unique in that nodes are already assumed to detect that they were asleep, so it is reasonable to assume they also detect that they were corrupt.

6 Conclusion

This work presents a consensus protocol in the fully fluctuating sleepy model without any hardware assumptions such as PoW or VDFs. We introduce a new primitive, *graded wakeness*, which we implement using several rounds of a simple interactive challenge-response protocol. Graded wakeness allows us to upgrade the graded agreement of D’Amato et al. [16] to tolerate fully fluctuating participation. We then give agreement protocols in the fully fluctuating sleepy model. Without uncorruption, no additional assumption is required. With uncorruption, we make a minor change to the graded wakeness and graded agreement protocols and require an additional mild assumption. We leave it for future work to optimize aspects of the protocols such as round and message complexity.

AI Disclosure

We used Claude Code to improve the clarity, grammar, and readability of this work.

References

- 1 Ittai Abraham, Danny Dolev, Alon Kagan, and Gilad Stern. Brief announcement: Authenticated consensus in synchronous systems with mixed faults. In *DISC*, volume 246 of *LIPICs*, pages 38:1–38:3. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- 2 Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. Sync HotStuff: Simple and practical synchronous state machine replication. In *SP*, pages 106–118. IEEE, 2020.
- 3 Hagit Attiya, Sweta Kumari, Archit Somani, and Jennifer L. Welch. Store-collect in the presence of continuous churn with application to snapshots and lattice agreement. *Inf. Comput.*, 285(Part):104869, 2022.
- 4 Christian Badertscher, Peter Gazi, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Ouroboros Genesis: Composable proof-of-stake blockchains with dynamic availability. In *CCS*, pages 913–930. ACM, 2018.
- 5 Nazreen Banu, Samia Souissi, Taisuke Izumi, and Koichi Wada. An improved Byzantine agreement algorithm for synchronous systems with mobile faults. *International Journal of Computer Applications*, 43(22):1–7, 2012. doi:10.5120/6400-8878.
- 6 Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Towards optimal distributed consensus (extended abstract). In *FOCS*, pages 410–415. IEEE Computer Society, 1989.
- 7 Alysso Neves Bessani, João Sousa, and Eduardo Adílio Pelinson Alchieri. State machine replication for the masses with BFT-SMART. In *DSN*, pages 355–362. IEEE Computer Society, 2014.
- 8 Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *CRYPTO (1)*, volume 10991 of *Lecture Notes in Computer Science*, pages 757–788. Springer, 2018.
- 9 François Bonnet, Xavier Défago, Thanh Dang Nguyen, and Maria Potop-Butucaru. Tight bound on mobile byzantine agreement. In *DISC*, volume 8784 of *Lecture Notes in Computer Science*, pages 76–90. Springer, 2014.
- 10 Harry Buhrman, Juan A. Garay, and Jaap-Henk Hoepman. Optimal resiliency against mobile faults. In *FTCS*, pages 83–88. IEEE Computer Society, 1995.
- 11 Vitalik Buterin, Diego Hernandez, Thor Kamphofner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X. Zhang. Combining GHOST and casper. *CoRR*, abs/2003.03052, 2020.
- 12 Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.*, 777:155–183, 2019.
- 13 Arka Rai Choudhuri, Aarushi Goel, Matthew Green, Abhishek Jain, and Gabriel Kaptchuk. Fluid MPC: secure multiparty computation with dynamic participants. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 94–123. Springer, 2021.
- 14 Phil Daian, Rafael Pass, and Elaine Shi. Snow White: Robustly reconfigurable consensus and applications to provably secure proof of stake. In *Financial Cryptography*, volume 11598 of *Lecture Notes in Computer Science*, pages 23–41. Springer, 2019.
- 15 Francesco D’Amato, Joachim Neu, Ertem Nusret Tas, and David Tse. Goldfish: No more attacks on Ethereum?! In *FC (1)*, volume 14744 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2024.
- 16 Francesco D’Amato, Roberto Saltini, Thanh-Hai Tran, and Luca Zanolini. TOB-SVD: Total-order broadcast with single-vote decisions in the sleepy model. In *ICDCS*, pages 1033–1043. IEEE, 2025.

- 17 Bernardo David, Giovanni Deligios, Aarushi Goel, Yuval Ishai, Anders Konring, Eyal Kushilevitz, Chen-Da Liu-Zhang, and Varun Narayanan. Perfect MPC over layered graphs. In *CRYPTO (1)*, volume 14081 of *Lecture Notes in Computer Science*, pages 360–392. Springer, 2023.
- 18 Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *EUROCRYPT (2)*, volume 10821 of *Lecture Notes in Computer Science*, pages 66–98. Springer, 2018.
- 19 Soubhik Deb, Sreeram Kannan, and David Tse. PoSAT: Proof-of-work availability and unpredictability, without the work. In *Financial Cryptography (2)*, volume 12675 of *Lecture Notes in Computer Science*, pages 104–128. Springer, 2021.
- 20 Sisi Duan and Haibin Zhang. Foundations of dynamic BFT. In *SP*, pages 1317–1334. IEEE, 2022.
- 21 Yuval Efron, Joachim Neu, and Toniann Pitassi. Fully-fluctuating participation in sleepy consensus. In *AFT*, volume 354 of *LIPIcs*, pages 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.
- 22 Yuval Efron, Joachim Neu, Ling Ren, and Ertem Nusret Tas. Optimal good-case latency for sleepy consensus. In *EUROCRYPT*, volume 16543 of *Lecture Notes in Computer Science*, pages 397–426. Springer, 2026.
- 23 Yuval Efron and Ertem Nusret Tas. Dynamically available common subset. *IACR Cryptol. ePrint Arch.*, 2025:16, 2025.
- 24 Ali Farahbakhsh, Giuliano Losa, Youer Pu, and Lorenzo Alvisi. Fast deterministically safe proof-of-work consensus. In *SP*, pages 2133–2151. IEEE, 2026.
- 25 Eli Gafni and Giuliano Losa. Brief announcement: Byzantine consensus under dynamic participation with a well-behaved majority. In *DISC*, volume 281 of *LIPIcs*, pages 41:1–41:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- 26 Mingyuan Gao, Hung Dang, Ee-Chien Chang, and Jialin Li. Mixed fault tolerance protocols with trusted execution environment. *CoRR*, abs/2208.01946, 2022.
- 27 Juan A. Garay. Reaching (and maintaining) agreement in the presence of mobile faults (extended abstract). In *WDAG*, volume 857 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 1994.
- 28 Juan A. Garay and Kenneth J. Perry. A continuum of failure models for distributed computing. In *WDAG*, volume 647 of *Lecture Notes in Computer Science*, pages 153–165. Springer, 1992.
- 29 Craig Gentry, Shai Halevi, Hugo Krawczyk, Bernardo Magri, Jesper Buus Nielsen, Tal Rabin, and Sophia Yakoubov. YOSO: you only speak once - secure MPC with stateless ephemeral roles. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2021.
- 30 Seth Gilbert and Nancy A. Lynch. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.
- 31 Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *J. ACM*, 37(3):549–587, 1990.
- 32 Leander Jehl and Hein Meling. Asynchronous reconfiguration for Paxos state machines. In *ICDCN*, volume 8314 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 2014.
- 33 Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *CRYPTO (1)*, volume 10401 of *Lecture Notes in Computer Science*, pages 357–388. Springer, 2017.
- 34 Leslie Lamport, Dahlia Malkhi, and Lidong Zhou. Vertical Paxos and primary-backup replication. In *PODC*, pages 312–313. ACM, 2009.
- 35 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
- 36 Andrew Lewis-Pye and Tim Roughgarden. Byzantine generals in the permissionless setting. In *FC (1)*, volume 13950 of *Lecture Notes in Computer Science*, pages 21–37. Springer, 2023.

- 37 Andrew Lewis-Pye and Tim Roughgarden. Permissionless consensus. *CoRR*, abs/2304.14701, 2023.
- 38 Jacob R. Lorch, Atul Adya, William J. Bolosky, Ronnie Chaiken, John R. Douceur, and Jon Howell. The SMART way to migrate replicated stateful services. In *EuroSys*, pages 103–115. ACM, 2006.
- 39 Dahlia Malkhi, Atsuki Momose, and Ling Ren. Towards practical sleepy BFT. In *CCS*, pages 490–503. ACM, 2023.
- 40 Dahlia Malkhi, Kartik Nayak, and Ling Ren. Flexible byzantine fault tolerance. In *CCS*, pages 1041–1053. ACM, 2019.
- 41 Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS*, pages 120–130. IEEE Computer Society, 1999.
- 42 Atsuki Momose and Ling Ren. Constant latency in sleepy consensus. In *CCS*, pages 2295–2308. ACM, 2022.
- 43 Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>, 2009. White paper.
- 44 Joachim Neu, Ertem Nusret Tas, and David Tse. Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In *SP*, pages 446–465. IEEE, 2021.
- 45 Javier Nieto, Joachim Neu, and Ling Ren. On the limits of consensus under dynamic availability and reconfiguration. *IACR Cryptol. ePrint Arch.*, 2025:1829, 2025.
- 46 Rafael Pass and Elaine Shi. The sleepy model of consensus. In *ASIACRYPT (2)*, volume 10625 of *Lecture Notes in Computer Science*, pages 380–409. Springer, 2017.
- 47 Toru Sasaki, Yukiko Yamauchi, Shuji Kijima, and Masafumi Yamashita. Mobile byzantine agreement on arbitrary network. In *OPODIS*, volume 8304 of *Lecture Notes in Computer Science*, pages 236–250. Springer, 2013.
- 48 Alexander Shraer, Benjamin C. Reed, Dahlia Malkhi, and Flavio Paiva Junqueira. Dynamic reconfiguration of primary/backup clusters. In *USENIX ATC*, pages 425–437. USENIX Association, 2012.
- 49 Qiang Tang and Yuchen Ye. On the communication complexity of sleepy consensus. *IACR Cryptol. ePrint Arch.*, 2026:960, 2026.