How Much Public Randomness Do Modern Consensus Protocols Need?

Joseph Bonneau[13], Benedikt Bünz[1], Miranda Christ[2], and Yuval Efron[2]

[1] New York University
jcb@cs.nyu.edu,bb@nyu.edu
[2] Columbia University
mchrist@cs.columbia.edu,ye2210@columbia.edu
[3] a16z crypto

Abstract. Modern blockchain-based consensus protocols aim for efficiency (i.e., low communication and round complexity) while maintaining security against adaptive adversaries. These goals are usually achieved using a public randomness beacon to select roles for each participant. We examine to what extent this randomness is necessary. Specifically, we provide tight bounds on the amount of entropy a Byzantine Agreement protocol must consume from a beacon in order to enjoy efficiency and adaptive security. We first establish that no consensus protocol can simultaneously be efficient, be adaptively secure, and use $O(\log n)$ bits of beacon entropy. We then show this bound is tight and, in fact, a trilemma by presenting three consensus protocols that achieve any two of these three properties.

1 Introduction

Consensus is a cornerstone of distributed computing, with research spanning over four decades [33,28]. In the consensus problem, n players (or nodes) with respective inputs must engage in communication in order to reach *agreement* on a value. The challenge stems from the presence of at most f *corrupted* (eq. *Byzantine*) players, which may deviate from any given protocol in arbitrary ways.

The consensus problem has been considered under a myriad of constraints and allowances across various axes, including network synchrony, setup assumptions, the use of randomness by honest players, and the capabilities of the adversary (Garay and Kiayias [20] provide a detailed survey). With the renewed interest in this problem due to its foundational role in blockchains, two properties of particular importance from a practical point of view have been *efficiency* and *adaptive security.*

Efficiency. We consider an efficiency notion motivated by modern blockchain-based consensus protocols. We say that a protocol has *low communication* (or in other words, is *laconic*) if in every round few of the players (i.e., $o(n) << n$) send messages. A protocol has *low latency* if it requires at most $o(n)$ rounds. We refer to a protocol as *efficient* if it has both low communication and low latency. Our efficiency notion is essential for consensus protocols to scale to thousands of players and finalize transactions quickly. Indeed, most modern blockchain systems (e.g., Bitcoin and Ethereum) are efficient in this sense.

Adaptive Security. We say that a consensus protocol is *secure against an adaptive adversary* if it solves consensus even when the adversary may choose to corrupt players "on the fly," based on its observations thus far. We consider both a basic *adaptive adversary* which can corrupt up to f players at any point based on the current transcript of the protocol, but cannot un-corrupt them, and a *mobile blocking adversary* which can choose potentially new players to target in each round but can only block their messages, and not cause arbitrary behavior.

Observe that either of these properties is straightforward to achieve without the other: if efficiency is not important, classical Byzantine fault-tolerant consensus protocols [10] are adaptively secure. If adaptive security is not important, Nakamoto-style consensus [32] with round-robin leader election is efficient.

In practice, the vast majority of blockchains, including Ethereum and Bitcoin, aim to achieve both of the above properties. They do so by relying on a source of shared randomness for the players, for tasks such as leader election and committee sampling [5,32,14]. Such a primitive that provides access to fresh randomness at every point in time is referred to in the literature as a *randomness beacon* [34]. Implementing such a beacon, however, is an expensive task, with current constructions either employing delay functions [29,7], requiring many rounds or much communication [37,12,35,24], or achieving a sub-optimal version of the ideal functionality subject to manipulation [22,5].

On the other hand, it turns out, that the use of *unpredictable* randomness is essential to the design of efficient consensus protocols [17,6,1]. Dolev and Reischuk [17] show that any deterministic protocol must have $\Omega(n^2)$ communication complexity. Bar-Joseph and Ben-Or [6] show that against a computationally unbounded adaptive adversary that can view the local state of all players, any (even randomized) synchronous BA protocol requires $\tilde{\Omega}(\sqrt{n})$ rounds. Abraham et al. show that without unpredictable randomness, any consensus protocol must use $\Omega(n^2)$ communication.

1.1 Our Setting

The goal of this paper is to quantify the precise amount of randomness must be drawn from such a beacon to allow for efficient and adaptively secure consensus protocols. We make minimal auxiliary assumptions, focusing on the information-theoretic (i.e., a computationally unbounded[4] adversary and no PKI), synchronous setting. Specifically, we consider the following ideal functionality for a beacon: At each round t, a fresh, uniformly random string is revealed to all players.

Such an assumption, referred to as an *idealized common coin* or *randomness beacon* is a common assumption in the BA protocols for asynchronous networks [15,31]. We say that a protocol has *low beacon entropy* if it can be implemented using a randomness beacon that generates $O(\log n)$ bits in total during the protocol. We stress that it is important that the random string revealed to all players at round t is not only uniform, but also is *unpredictable* prior to round t. In other words, no adversary can *predict* the contents of the string of round t prior to round t. Abraham et al. [1] show that any protocol secure against an adversary that can predict the content of the beacon ahead of time, must use $\Omega(n^2)$ communication. We emphasize that our lower bound applies even when the protocol has access to our beacon with stronger unpredictability.

1.2 Our Results

We provide a complete and tight characterization of efficiency, adaptive security, and low beacon randomness:

Impossibility (Section 4, Theorem 1 and Corollary 1): We show that no consensus protocol can simultaneously achieve all three properties. That is, any efficient, adaptively secure consensus protocol must use a randomness beacon that outputs $\omega(\log n)$ bits. Our impossibility result holds against a computationally unbounded adversary in a broadcast communication model.

Possibility (Section 5, Lemmas 2, 5 and 6): We show that the above impossibility is tight by presenting three consensus protocols, each realizing exactly two of the above three properties. Our protocols work against a computationally unbounded adversary in a peer-to-peer communication model. Together, they demonstrate a *trilemma*: consensus protocols can achieve any two of efficiency, adaptive security and low-entropy, but not all three.

2 Related Work

A long line work considers a *computationally bounded* adversary [13,2,18,21]. Protocols in this setting employ cryptographic tools to enable adaptive secure consensus protocols with as few as $O(1)$ rounds (in expectation). This research culminates in the work of Ghinea, Goyal and Liu-Zhang [21], which matches the decades-old lower bound Chor, Merritt, and Shmoys [13] which states that any r round consensus protocol has error probability at least $\Omega(\frac{1}{r^r})$.

Coming back to an unbounded adversary, other works [26,36,8,1,16] consider relaxed versions of the adaptive adversary, and design consensus protocols that circumvent the aforementioned round and communication lower bounds of [1,6].

[4] Under computational assumptions, e.g., using a delay function, a beacon can be constructed. The unbounded adversary setting enables us to isolate the utility of the beacon.

Adaptively secure consensus. By Abraham et al. [1], any protocol that is secure against a strongly adaptive adversary must use $\Omega(n^2)$ bits of communication. This bound is known to be tight in the setting of a computationally bounded adversary by the work of Abraham et al. [2]. The protocol of this work also achieves optimal $O(1)$ round complexity, in expectation.

Adversary Relaxations. On the road to circumventing the lower bounds of [1,6], various relaxations to the strongly rushing adversary have been considered.

- A typical relaxation is that if the adversary chooses to corrupt a player p at round r, player p still performs the honest behaviour of round r prior to the adversary taking control of p (at which point the adversary can send additional messages, still in round r). Protocols achieving $o(n^2)$ communication complexity under this assumption include the work of King and Saia [26], and the follow up work by King, Lonargan, Saia and Trehan [25] in which they show protocols with $O(n^{1.5})$ communication complexity. Additional cryptographic assumptions allow the design of protocols with nearly linear communication complexity [8,11,2,9]. Another class of protocol circumventing the communication lower bound of [1] are motivated by blockchain applications, and make use using proof-of-work or proof-of-stake assumptions [16,32].
- Another line of work considers a *late* adaptive adversary. Roughly, this is an adaptive adversary with an outdated view of the state of the protocol. At each round the adversary may choose players to corrupt based on all information available so far, however the actual corruption occurs several rounds later. Most of these works consider relaxed variants of the consensus problem, e.g., almost-everywhere consensus [36,4,3,27].
- A mobile blocking adversary has been considered previously in [36], and is generally related to the well studied model of omission failures [30,23].

Randomness beacon entropy. As mentioned, the idealized randomness beacon assumption, also known as a common coin, is extensively used in consensus protocol design in asynchronous networks (See [31] and references within). A recent work [23] studies a similar question to ours. Specifically, they consider protocols secure against an adaptive *omission failure* adversary, and characterize the trade-off between the required number of oracle calls to a random beacon and the round complexity.

3 Preliminaries

Notation. We let λ denote the security parameter. If a function $f(\lambda)$ is $O(1/\mathsf{poly}(\lambda))$ for every polynomial in λ, we say f is *negligible* in λ. We write $f = \mathsf{negl}(\lambda)$. We let $\log(x)$ denote the logarithm base 2 of x. If X is a random variable, we denote the min-entropy of X by $H_\infty(X) = \min_{x \in \mathrm{Supp}(X)} \frac{1}{\Pr[X=x]}$.

Model. We consider a network of n players (eq. nodes or participants). We focus here on the permissioned setting, in which the set of n players is fixed and known to all players in advance. At most f players can can be corrupt. We further assume that communication takes place over a synchronous network with maximum message delay of Δ, i.e. a player p at round t has received all messages sent to it up to round $t - \Delta$. For simplicity of exposition, we assume that $\Delta = 1$, but all our results extend naturally the general case of delay parameter Δ. Throughout the paper we consider two models for a communication network. In the *peer-to-peer* model, players may send different messages to different players during each round. In the *broadcast* model, players (including corrupted players) may only broadcast a message, which is received by all other players.

Specifically, the latter model does not allow corrupt players to equivocate. Our lower bound holds even in the more generous broadcast model, and our upper bounds hold even in the more restrictive peer-to-peer model of communication.

Adversary model. All of the adversaries considered in the paper are computationally unbounded, and in particular we assume no PKI (though we do assume authenticated channels between participants). Specifically, we consider three types of adversaries, *static*, *adaptive*, and *mobile blocking*. Let f be the adversary's corruption budget. It is useful for us to consider an adversary as a *pair* of algorithms $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, where $\mathcal{A}_0$ is responsible for choosing the identities of up to f corrupted players at every round t, and $\mathcal{A}_1$ is the adversary participating in the protocol. A bit more formally, at each round t, $\mathcal{A}_0$ may observe $(\mathit{\Pi}, t, \mathrm{Tr}_t)$ and output a set $\mathcal{F}_t$ of at most f players. Here, $\mathit{\Pi}$ is a description a protocol and Tr_t refers to the transcript of a protocol up to round t (see Definition 2). Each adversary type we consider imposes different restrictions on either $\mathcal{A}_0$ or $\mathcal{A}_1$.[5]

- *Static.* The static adversary chooses the identities of f corrupt players at round $t = 0$, and this choice is fixed throughout the execution of the protocol. The adversary has complete control over the behaviour of the corrupted players. In other words $\mathcal{F}_t = \mathcal{F}_0$ for all t. There are no restrictions on $\mathcal{A}_1$.
- *Adaptive.* At the beginning of each round t, the adversary chooses the identities of $f_t < f$ players to corrupt in that round, based on its observations of the transcript of the protocol and the players it has corrupted thus far. These players stay corrupted for the rest of the execution. In other words, $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for all t. There are no restrictions on $\mathcal{A}_1$.
- *Mobile blocking.* A mobile blocking adversary can, at every round, observe $(\mathit{\Pi}, t, \mathrm{Tr}_t)$, where t is a round, and Tr_t is the transcript of $\mathit{\Pi}$ up to round t, and output a set $\mathcal{F}_t$ of at most f players that are prohibited from sending messages at round t. We emphasize that unlike the adaptive adversary which is constrained by the total number of corruptions, the mobile adversary may corrupt up to f players in each round regardless of how many players it has corrupted in the past. A mobile blocking adversary is also captured by the $(\mathcal{A}_0, \mathcal{A}_1)$ notation via $\mathcal{A}_0(\mathit{\Pi}, t, \mathrm{Tr}_t) = S$, and $\mathcal{A}_1$ be an adversary behaviour in which corrupted players send no messages.

In any adversary variant, we say that an adversary corrupting at most $f = \rho n$ players is *ρ-bounded*.

Randomness. The protocols we consider in this work have two main sources of randomness. For both of them, we assume an ideal functionality as our goal is to reason about the nature and amount of randomness required to design consensus protocols.

1. **Common Random String (CRS).** We assume that at round $t = 0$, an arbitrarily long (as per the protocol's specification) uniformly random string CRS is revealed to all players. A *static* adversary must choose the identities of corrupt players *prior* to the CRS being revealed. An *adaptive* or *mobile* adversary can choose the identities of corrupt players after CRS is revealed.
2. **Randomness Beacon.** A randomness beacon is an oracle that at every round t, produces a fresh uniformly random string seed_t of an arbitrary length (as per the protocol's specification). In particular, for each round t, an *adaptive* or a *mobile* adversary must choose the identities of corrupted players prior to the revelation of seed_t.

Clearly, in the presence of an adaptive adversary the latter source of randomness is significantly more powerful than the former. It is precisely the latter source of randomness that is the main focus of this paper.

Authenticated Channels. Similarly to prior work [26,8], we assume that communication channels in our network are *authenticated*. Intuitively, this means that each player knows the identity of the sender for any message received. This can be formalized by instantiating communication channels with a map that maps a message m to the tuple (m, p), where p is the sender of m. This mapping is fixed and can not be tampered with by the adversary.

[5] One might ask about the possibility of a mobile adversary capable of corruption, but this is tricky to reason about as it requires a notion of "un-corrupting" players.

Executions. An *execution* is a tuple $E = (\Pi, \mathcal{A}, r)$ where Π is the protocol run by honest players. $\mathcal{A}$ is a particular adversary, i.e. $\mathcal{A}$ is an algorithm that chooses corrupt players (only at $t = 0$ if static, otherwise $\mathcal{A}_0$ chooses at every round t) based on the transcript of the protocol and internal state of corrupt players, and instructions for the behaviour of the environment. r denotes the random coins of all players. Given a protocol Π we say that E is an execution of Π if the first component of E is Π.

We say that an execution is *admissible* in the adaptive model if for all rounds $t \geq 0$ it holds that $f_t < \frac{1}{3}n$. In the static model we further demand that the identities of corrupt players remain fixed throughout the execution of the protocol. At times we abuse notation and refer to an execution also as the random variable $(\Pi, \mathcal{A})$ which is a distribution over executions in which Π is the protocol run by honest players, and $\mathcal{A}$ is the adversary (corrupt players in each round, and actions taken by them). We denote this random variable by $E_{\mathcal{A}}$.

The specification of the consensus problem we consider takes the form of the well known Byzantine Agreement problem [33,28].

Byzantine Agreement (BA). In the BA task, n players must come to an agreement on a bit under some constraints. Each player P_i has an input $b_i \in \{0, 1\}$, and produces an output $o_i \in \{0, 1\}$. We say that a protocol Π solves the BA task if the following three properties are guaranteed by Π, except for with $\mathsf{negl}(\lambda)$ probability.

1. *Termination.* There exists t such that all honest players have produced an output by round t.
2. *Agreement.* For any pair of honest players P_i, P_j, the probability (over the randomness of the adversary and the protocol) that these players output different values is at most $\mathsf{negl}(\lambda)$.
3. *Validity.* If all honest players have the same input bit b, then all honest players always output b.

An additional critical notion relating to a BA protocol is *latency*, which is defined as follows. Intuitively, latency captures the expected number of rounds it takes for Π to terminate.

Definition 1 (Latency). *Let Π be a protocol solving BA, and let $T_{i,\mathcal{A}}$ denote the random variable indicating the round in which player i terminates given an adversary $\mathcal{A}$ corrupting some subset of the players. Given an execution $E = (\Pi, \mathcal{A}, r)$, let H_E denote the set of players that remain honest throughout the entire execution. Π has* expected *latency ℓ given f corruptions if for all adversaries $\mathcal{A}$ corrupting at most f players,*

$$\max_{\substack{(\Pi,\mathcal{A}) \\ \mathcal{A} \text{ corrupts at most } f \text{ players}}} \mathop{\mathbb{E}}_{E=(\Pi,\mathcal{A},r)} \left[\max_{p \in H_E} T_{p,\mathcal{A}} \right] \leq \ell.$$

Randomness beacon. In this paper, we consider an ℓ-bit randomness beacon to be an n-party protocol Π that satisfies termination and agreement and, on input 1^λ, outputs a string s of length $\ell(\lambda)$. Furthermore, the value output by the honest parties must be computationally indistinguishable from random. That is, let $\mathcal{A}$ be any (computationally unbounded) adversary corrupting f out of n players. Let $\mathcal{B}$ by any computationally bounded adversary. Denote by $v \leftarrow \Pi^{\mathcal{A}}$ the output of the honest parties from an execution of Π with the adversary $\mathcal{A}$. For any such $\mathcal{A}, \mathcal{B}$ it must hold that:

$$\left| \Pr[\mathcal{B}(1^\lambda, v) = 1 : \ v \leftarrow \Pi^{\mathcal{A}}] - \Pr[\mathcal{B}(1^\lambda, r) : \ r \leftarrow \{0,1\}^{\ell(\lambda)} \ ] \right| \leq \mathsf{negl}(\lambda).$$

We say that Π is secure against a static/adaptive/mobile blocking adversary if the above holds and $\mathcal{A}$ is a static/adaptive/mobile blocking adversary respectively. We note that the definition of a randomness beacon varies throughout the literature; here, we consider a weak notion where the adversary corrupting parties during the protocol execution is *different* from the adversary attempting to distinguish the beacon output from random. This notion is still nontrivial.

4 Impossibility Result

In the following, we define formal notions in order to rigorously discuss the *amount* of common randomness consumed by a BA protocol with adaptive security and low communication. We begin with defining the transcript of a protocol.

Definition 2 (Transcript). *Consider a* BA *protocol* Π *for a network of* n *players, and let* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *denote some adversary. Denote by* $E_{\mathcal{A}}$ *the random variable of executions under* $(\Pi, \mathcal{A})$*. We define the transcript* $Tr_{E_{\mathcal{A}}}$ *to be the random variable that indicates the identities and contents of the players speaking in each round. Formally, we have*

- $\mathrm{Tr}_{E_{\mathcal{A}}} = \left\{\mathrm{Tr}^t_{E_{\mathcal{A}}}\right\}_{t\in\mathbb{N}}$
- $\mathrm{Tr}^t_{E_{\mathcal{A}}} = (I^t_{E_{\mathcal{A}}}, C^t_{E_{\mathcal{A}_1}}, A^t_{\mathcal{A}}, \mathsf{seed}_t, \mathsf{CRS})$*, where* $I^t_{E_{\mathcal{A}_0}} \subseteq [n]$ *is a subset of honest players at round* t*, and* $C^t_{E_{\mathcal{A}}} \in \left(\{0,1\}^*\right)^{I^t_{E_{\mathcal{A}}}}$ *denotes the messages of those players. Furthermore, an honest player* p_i *speaks in round* t *iff* $i \in I^t_{E_{\mathcal{A}}}$*, and the message it sends is consistent with* $C^t_{E_{\mathcal{A}}}$*.* $A^t_{\mathcal{A}}$ *contains the identities of the adversarial parties speaking at round* t *and the contents of their messages*

We consider the following notion of unpredictability, that intuitively says that the probability that an adaptive adversary can guess correctly the set of speaking parties in every round, given the transcript of the protocol up to that round is negligible.

Definition 3 (Adversary's guess of speaking parties). *Let* Π *be a consensus protocol, and let* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *be an adversary. We define the adversary's* guess *to be a series* $(\hat{I^0_{\mathcal{A}}}, \hat{I^1_{\mathcal{A}}}, \hat{I^2_{\mathcal{A}}}, ...)$ *given by an algorithm* $\mathcal{A}_2$ *of the adversary's choice where for each* $t \in \mathbb{N}$*,*

$$\hat{I}^t_{\mathcal{A},\mathcal{A}_2} \leftarrow \mathcal{A}_2(\Pi, \left\{\mathrm{Tr}^i_{E_{\mathcal{A}}} \mid 0 \le i \le t-1\right\}, t)$$

Given the above definition, we can now define global leader unpredictability:

Definition 4 (Global leader unpredictability). *Let* Π *be a* BA *protocol. We say that* Π *has global leader unpredictability if for any adversaries* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *and* $\mathcal{A}_2$*,*

$$\Pr\left[\forall t, \hat{I}^t_{\mathcal{A},\mathcal{A}_2} = I^t_{E_{\mathcal{A}_0}} \mid \mathsf{CRS}\right] = \prod_{t=0}^{\infty} \Pr\left[\hat{I}^t_{\mathcal{A},\mathcal{A}_2} = I^t_{E_{\mathcal{A}_0}} \mid \left\{I^i_{E_{\mathcal{A}_0}} \mid 0 \le i < t\right\}, \mathsf{CRS}\right] \le \mathsf{negl}(\lambda)$$

We now prove that if a protocol Π satisfies global leader unpredictability, then its transcript contains a significant amount of min-entropy.

Lemma 1. *Let* Π *be a* BA *protocol that satisfies global leader unpredictability. Then for any constant* $c > 0$ *and any adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$*,*

$$\sum_{t=0}^{\infty} H_\infty(\mathrm{Tr}^t_{E_{\mathcal{A}}} \mid \left\{\mathrm{Tr}^i_{E_{\mathcal{A}}} \mid 0 \le i < t\right\}, \mathsf{CRS}) \ge c \log n.$$

Proof. Assume towards a contradiction that there exist a constant $c > 0$ and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ such that

$$\sum_{t=0}^{\infty} H_\infty(\mathrm{Tr}^t_{E_{\mathcal{A}}} \mid \left\{\mathrm{Tr}^i_{E_{\mathcal{A}}} \mid 0 \le i < t\right\}, \mathsf{CRS}) < c \log n.$$

Now for each $t \in \mathbb{N}$, denote by k_t the value $H_\infty(\mathrm{Tr}^t_{E_{\mathcal{A}}} \mid \left\{\mathrm{Tr}^i_{E_{\mathcal{A}}} \mid 0 \le i < t\right\}, \mathsf{CRS})$. Applying $f(x) = \frac{1}{2^x}$ to both sides of the inequality above, we get that

$$\frac{1}{2^{\sum_{t=0}^{\infty} k_t}} > \frac{1}{n^c}$$

Thus, by definition of min-entropy, we have that there exist strings $S_t, t \in \mathbb{N}$ s.t.

$$\frac{1}{2^{k_t}} = \Pr[\mathrm{Tr}^t_{E_\mathcal{A}} = S_t \mid \left\{\mathrm{Tr}^i_{E_\mathcal{A}} \mid 0 \leq i < t\right\}, \mathsf{CRS}]$$

We thus have that

$$\Pr[\forall t \in \mathbb{N}, \mathrm{Tr}^t_{E_\mathcal{A}} = S_t \mid \mathsf{CRS}] = \prod_{t=0}^{\infty} \Pr[\mathrm{Tr}^t_{E_\mathcal{A}} = S_t \mid \left\{\mathrm{Tr}^i_{E_\mathcal{A}} \mid 0 \leq i < t\right\}, \mathsf{CRS}] > \frac{1}{n^c}$$

Where the first transition is by definition of the probability of event intersection. Since $I^t_{E_\mathcal{A}}$ is a part of $\mathrm{Tr}^t_{E_\mathcal{A}}$, We thus get that there exists set $\hat{I}_t, t \in \mathbb{N}$ s.t.

$$\Pr[\forall t \in \mathbb{N}, I^t_{E_\mathcal{A}} = \hat{I}_t \mid \mathsf{CRS}] > \frac{1}{n^c}$$

Now note that an adversary $\mathcal{A}_2$ that predicts at round t the set $\hat{I}_t$ can realize this success probability, thus violating the global leader unpredictability condition, as required. Such an adversary is realizable both in the adaptive and mobile blocking models since the adversary at round t has definitive knowledge of the transcript up to and including round $t-1$, including CRS. □

We would now like to prove that the lack of the global unpredictability condition, combined with low communication, implies susceptibility to attacks by adaptive adversaries. With the notion of a transcript in hand, we can also now formally define relevant notion for this paper of *low* communication.

Definition 5. *Let Π be a consensus protocol, and let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be an adversary. We define the communication complexity of Π to be the random variable $C = \sum_{t \in \mathbb{N}} |I^t_{E_\mathcal{A}}|$. We say that the communication is uniform if there exists T such that except for with $\mathsf{negl}(\lambda)$ probability it holds that $|I^t_{E_\mathcal{A}}| = O(\frac{C}{T})$ for all $t \leq T$ and $|I^t_{E_\mathcal{A}}| = 0$ for all $t > T$. We refer to $\frac{C}{T}$ as the uniform communication complexity of Π. We say that a protocol has* low communication *if it has uniform communication of $o(n)$.*

The goal of defining uniform communication, as opposed to just considering the general number of bits exchanged between players in the protocol, is to speak rigorously about protocols in which only a few players speak in each round. We now aim to prove the following theorem, which says that any BA protocol in which few players speak in every round, uses low beacon entropy, and is adaptively secure, requires many rounds. In other words, no BA protocol can simultaneously be efficient, adaptively secure, and use low beacon entropy.

Theorem 1. *Let Π be a protocol that has the following properties:*

- *Π does not satisfy global leader unpredictability.*
- *Except for $O(1)$ initial rounds, Π has uniform communication complexity k*

Then Π w.p. $\frac{1}{p(n)}$ for some polynomial n, Π requires $\Omega(\frac{\rho n}{k})$ rounds in the presence of a ρ-bounded adaptive adversary, for any $\rho \geq \frac{k}{n}$. Furthermore, if a mobile blocking adversary is considered, then Π does not exist for any $\rho \geq \frac{k}{n}$.

Proof. By the assumption that Π does not satisfy global leader unpredictability, we deduce that there exist a polynomial $p(n)$ and adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and $\mathcal{A}_2$ such that

$$\Pr[\forall t, \hat{I}^t_{\mathcal{A},\mathcal{A}_2} = I^t_{E_{\mathcal{A}_0}} \mid \mathsf{CRS}] \geq \frac{1}{p(n)}.$$

Now consider the following adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$, acting as follows. Let $r = O(1)$ be the number of initial rounds which may have high communication complexity. In the following, we employ a well-known fact [13], that any BA protocol with r rounds has error probability of at least $\Omega(\frac{1}{n^r})$, with the adversary needing to corrupt at most r players [13]. In our case, $\Omega(\frac{1}{n^r}) = 1/n^{O(1)}$.

1. For the first r rounds, employ the adversary strategy [13], corrupting at most r players in the process.
2. $\mathcal{B}_0$ acts as follows: At the beginning of any round $t > r$, use $\mathcal{A}_2(\mathit{\Pi}, \left\{\mathrm{Tr}^i_{E_\mathcal{A}} \mid 0 \leq i < t\right\})$ to obtain $\hat{I}^t_{\mathcal{A},\mathcal{A}_2}$. In the case of the adaptive adversary, corrupt of all the players in $\hat{I}^t_{\mathcal{A},\mathcal{A}_2}$ until the total number of corrupted parties has reached the corruption budget ρn. In the case of a mobile blocking adversary, corrupt them for round t.
3. $\mathcal{B}_1$ sends no messages by the corrupted players; that is, the players in $\hat{I}^t_{\mathcal{A},\mathcal{A}_2}$. Note that this is compatible behaviour with the mobile blocking adversary, in addition to the adaptive adversary.

Condition on the event that $\mathcal{A}_2$ has guessed correctly the identity of speakers at all rounds, which occurs w.p. at least $\frac{1}{p(n)}$, by assumption. Due to the strategy of [13], we have that with at least inverse polynomial probability, $\mathit{\Pi}$ does not solve BA up to round r of the execution. For the following $\mathit{\Omega}(\frac{\rho n}{k})$ rounds, no progress is made since the adversary corrupts all parties participating in the protocol for those rounds. Thus at least with inverse polynomial probability, $\mathit{\Pi}$ requires $\mathit{\Omega}(\frac{\rho n}{k})$ rounds, as required. In the case of a mobile blocking adversary, no progress is made at all, indefinitely, and so in particular the protocol either has no termination, or has error probability at least $\mathit{\Omega}(\frac{1}{n^r \cdot p(n)})$, which is a contradiction. □

We now show that global leader unpredictability implies a randomness beacon. This randomness beacon simply runs $\mathit{\Pi}$ and computes the hash of the identities that spoke in each round. Since these identities are unpredictable, given enough parties there is sufficient randomness to construct a beacon.

Corollary 1. *Let $\mathit{\Pi}$ be an n-party protocol satisfying global leader unpredictability. If $n \geq \lambda$, there exists a randomness beacon $\mathit{\Pi}'$ such that:*

- *$\mathit{\Pi}'$ has the same communication complexity and latency as $\mathit{\Pi}$.*
- *$\mathit{\Pi}'$ outputs λ bits that are indistinguishable from random by any efficient adversary in the random oracle model.*

Proof. First, observe that $\mathit{\Pi}'$ trivially has the same communication complexity and latency as $\mathit{\Pi}$, since it involves only running $\mathit{\Pi}$ and applying a function to its transcript. Let $H : \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function which we model as a random oracle. Let $\mathit{\Pi}'$ be the protocol obtained by running $\mathit{\Pi}$ and outputting the hash of the identities of the parties that speak in each round of $\mathit{\Pi}$. Since we operate in the broadcast setting with authenticated channels, all parties know these identities of speaking parties. Recall that global leader unpredictability states that this tuple of speaking parties has at least $c \log n$ min-entropy for any constant c. Therefore, for any fixed string s, the probability that this tuple of identities equals s is at most $1/\lambda^c$ for any constant c. Let $\mathcal{B}$ be a computationally bounded adversary making a polynomial number of queries to the random oracle $\mathcal{O}$. The probability that $\mathcal{B}$ queried the tuple of speaking parties to $\mathcal{O}$ is smaller than the inverse of any polynomial, which is negligible. Therefore, except with negligible probability the hash of the tuple of speaking parties is a freshly random value from the perspective of $\mathcal{B}$. □

5 Possibility Results

Having established the above trade-off, we turn to showing that it exactly captures the role of randomness in efficient and adaptively secure BA. Specifically, we present three protocols solving BA, all simplified versions of known protocols from the literature, and prove that each of them satisfies two of the three properties we described above. To make our results as strong as possible, throughout this section, we consider a model where players are deterministic and all players are given access to an ideal randomness beacon and a CRS. Recall (see Section 3) that an adaptive adversary must make corruption choices for round r *prior* to seeing the beacon output of round r. While for the lower bound result we assumed a broadcast model of communication, we assume a peer-to-peer network for the upper bounds, to make our results as strong as possible. In particular, corrupt players can equivocate. We formally describe our framework and general BA protocol in the following section, and then showcase how this framework is instantiated in three ways to obtain protocols:

1. **Efficiency and adaptive security.** We show that when entropy from a beacon is not limited, there exists a protocol that is adaptively secure and efficient. The details, along with our general framework, are in Section 5.1.
2. **Low beacon entropy and adaptive security.** If one is willing to forgo low communication, we show that there is an adaptively secure, low beacon entropy protocol that also has $O(1)$ expected round complexity. The details are in Section 5.2.
3. **Low beacon entropy and efficiency.** If one wishes to forgo adaptive security, then there exists an efficient protocol that uses low beacon entropy that is secure against a static adversary. The details are in Section 5.3.

Mobile blocking adversary. All the proofs relating to security against an adaptive adversary in this section can easily be modified to work for a mobile blocking adversary, with the *same* protocols. The main observation is that against an adversary that can only silence players, none of our proofs use the property that *the same* parties are corrupted between rounds, and also that in our protocols, the actions of an honest player depend only on the messages received from the previous round, and not any other round in the past.

5.1 Efficiency and Adaptive Security

In this section we describe our general framework for BA protocol design, along with one instantiation of it to obtain a BA protocol that has low communication(i.e. $O(\lambda)$ parties talk in each round) and is secure against an adaptive adversary, as long as $3f_t + 1 < (1-\epsilon)n$ for all $t \in \mathbb{N}$ and constants ϵ. Formally, we prove the following in this section:

Lemma 2. *For all $\epsilon > 0$, assuming a randomness beacon, there exists a protocol Π that except for with $\mathsf{negl}(\lambda)$ probability, solves the BA task in the presence of an adaptive adversary that satisfies $3f_t+1 < (1-\epsilon)n$ for all $t \in \mathbb{N}$.*
Furthermore, the protocol has $O(1)$ expected latency, and the protocol has uniform $O(\lambda)$ communication complexity, in expectation.

As mentioned, we assume access to a randomness beacon. I.e. at each round $t \in \mathbb{N}$, a uniformly random string of length α is given to all players, completely independent of all other strings provided to the players. This protocol makes no use of the given CRS. Denote the string distributed at round t to the players by the randomness beacon by seed_t. We choose $\alpha = O(\lambda \log n)$, where λ is the security parameter, and we treat seed_t as $\mathsf{seed}_t = (\ell_t, com_t)$ where $\ell_t \in [n]$ is the identity of a player, which is referred to as the *leader* of round t, and $com_t \subseteq [n]$ is a subset of size $O(\lambda)$ of players, indicating the *committee* of round t.

Our general framework, with which all three of our protocols are designed, resembles that of Gafni and Losa [19] of interlacing executions of the commit-adopt task and leader election. The main differences are the use of the randomness beacon to ensure both low communication and security against an adaptive adversary, and the consideration of a stronger adversary (Losa and Gafni assume a non equivocating adversary). With this in mind, we describe the general framework and the its concrete implementation to obtain Lemma 2. The following sections explain how to modify the implementation to obtain the other two protocols. Our framework is comprised from the interlacing of two components.

– Commit-Adopt (CA). The CA protocol consists of 2-rounds. In each of these rounds, only the *committee* com_t speaks. If the CA step fails to achieve consensus, players proceed to the second phase of the protocol.
– Conciliator (CO). Intuitively, the goal of the Conciliator task is to bring back the honest players into a consistent view after the previous CA failed. This is done by running an additional CA and a leader election, and then outputting a value to continue with for the subsequent CA.

We now formally define the two procedures CA, and CO.

Definition 6 (Commit-Adopt). *In the commit-adopt task (CA), each player receives an input value z, and must produce an output of the form commit(z') or adopt(z') for some value z', with the following guarantees.*

1. Agreement. *If an honest player outputs commit*(z) *for some value* z*, then all honest players must output either commit*(z) *or adopt*(z).
2. Validity. *If all honest players input the same value* z*, then all honest players must output commit*(z).
3. Termination. *There exists a round* $r \in \mathbb{N}$ *such that by round* r*, all awake honest players have submitted an output.*

Definition 7 (Conciliator). *In the conciliator task* (CO)*, each player has an input value* z *and must produce an output with the following guarantees.*
1. Validity. *If all honest players input the same value* z*, then all honest players output* z.
2. Termination. *There is a round* $r \in \mathbb{N}$ *such that all honest players output by round* r.
3. Probabilistic Agreement. *With probability at least* $\frac{2}{3}$*, all players output the same value* z *inputted by some honest player.*

Our generic protocol alternates between executions of CO and CA until BA is solved. We denote the i-th execution of CA and CO by $\mathsf{CA}[i]$ and $\mathsf{CO}[i]$, respectively. We now provide formal descriptions of the protocols for CA and CO using ℓ_t, com_t as explained above to obtain Lemma 2. The following sections explain how these implementations are modified to obtain our other protocols.

Algorithm 1 (CA) *We employ the following adopt-commit protocol executed by all honest players* p *with input* z_p.

1. *At round* $t = 0$*, if* $p \in com_0$*,* p *broadcasts its input* z_p*. Otherwise, do nothing.*
2. *At round* $t = 1$*, if* $p \notin com_1$*, do nothing. Otherwise, if there exists a value* z *that* p *has received* z *from more than* $\frac{2|com_0|}{3}$ *of the players in* com_0[6]*,* p *broadcasts vote*(z)*. Otherwise, do nothing.*
3. *At round* $t \geq 2$*,* p *decides on its output as follows.*
   (a) *If there exists value* z *such that* p *has received vote*(z) *from at least* $\frac{2|com_1|}{3}$ *of the players in* com_1*, output commit*(z)*.*
   (b) *Else if there exists a value* z *for which* p *received more vote*(z) *from players in* com_1 *than for any other value, output adopt*(z)*.*
   (c) *Else, output adopt*(z_p)*.*

We now prove that the above procedure solves the CA problem in the presence of an adaptive adversary, as long as there is a constant $\epsilon > 0$ s.t. $3f_t + 1 < (1-\epsilon)n$ for all $t \in \mathbb{N}$. We begin with the following observation, which is used liberally throughout the section. We denote the above protocol by $\mathit{\Pi}$.

Observation 2 *Let* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *be an adaptive adversary. Then for all* t *it holds that*

$$\Pr\left[|\mathcal{A}_0(\mathit{\Pi}, t, \mathrm{Tr}_t) \cap com_t| \geq \frac{|com_t|}{3}\right] = \mathsf{negl}(\lambda)$$

Proof. The proof follows from standard concentration bounds. Specifically we get that the expected number of corrupted players in com_t is upper bounded by $\frac{|com_t|}{3}(1-\epsilon)$, and thus by a Hoeffding bound we get the probability that the number of corrupted players in com_t exceeds $\frac{|com_t|}{3}$ is at most $e^{-\Omega(\epsilon^2|com_t|)} = e^{-\Omega(\epsilon^2\lambda)}$. For this we use the facts that $|com_t| = \Omega(\lambda)$, ϵ is a constant, the choice of each player in com_t being independent, and the independence of $\mathcal{A}_0(\mathit{\Pi}, t, \mathrm{Tr}_t)$ on com_t. □

We thus assume from here on in for the remainder of the analysis that an honest super-majority assumption holds at all rounds amongst the committee members, i.e. $|\mathcal{A}_0(\mathit{\Pi}, t, \mathrm{Tr}_t) \cap com_t| < \frac{|com_t|}{3}$ holds for all t. With that in mind all that is left is to prove that the above protocol solves the CA problem.

Lemma 3. *For any constant* $\epsilon > 0$*, except for with* $\mathsf{negl}(\lambda)$ *probability, Algorithm 1 solves the* CA *task whenever* $3f_t + 1 < (1-\epsilon)n$.

[6] Here we use our authenticated channels assumption

Proof. Termination is clear from the behavior of the protocol. We move on to Validity, assume that all honest parties start the protocol with input z, and let com_0, com_1 be as in the protocol. Then in particular, all honest players in com_0 broadcast z. The above observation implies that any honest player in com_1 observes the value z from more than $\frac{2}{3}$ of the players in com_0. Thus all honest players in com_1 broadcast $vote(z)$ message at round 1. Which in turn causes all honest players to output $commit(z)$ at $t = 2$, as required. For Agreement, let $commit(z)$ be the value output by some honest player p. Which means that p observed $vote(z)$ from more than $\frac{2}{3}$ of the members of com_1. Note first that no other honest member of com_1 sends $vote(z')$ for a different value, as this implies that that two honest players q_1, q_2 in com_1, respectively received multicasts of z, z' from more than $\frac{2}{3}$ of the members of com_0, and a quorum intersection argument implies the existence of an honest player in com_0 that broadcast two different inputs, which can not occur. Thus for any other value z' an honest player can only observe strictly less than $\frac{|com_1|}{3}$ $vote(z')$ messages from players in com_1. On the other hand, if p observed $vote(z)$ from more than $\frac{2}{3}$ of the members of com_1, this implies, together with Theorem 2, that more than $\frac{1}{3}$ of the honest members of com_1 broadcast $vote(z)$, which means *all honest players* have seen $vote(z)$ messages from more than $\frac{1}{3}$ of the members of com_1. Combined, this means that all honest players have observed more $vote(z)$ messages from players in com_1 than for any other value, and thus output either $adopt(z)$ or $commit(z)$, as required.

□

We now proceed to designing a protocol for the CO task.

Algorithm 2 (CO) *We employ the following conciliator protocol executed by all honest players p with input z_p.*

```
1. If $t \in [0, 2]$, run according to CA protocol with input value $z$.
2. If $r = 3$, if $p = \ell_3$ or $p \in com_3$, broadcast CA output.
3. if $r \geq 4$, then:
   (a) If $p$ received $commit(z)$ from more than $\frac{1}{3}$ of the players in $com_3$ for some value $z$, then $p$ outputs $z$.
   (b) Else, if $p$ received $adopt(z)$ or $commit(z)$ for some value $z$ from $\ell_3$, then output $z$
   (c) Else, output $z_p$.
```

We now proceed to prove the correctness of the above procedure.

Lemma 4. *For any constant $\epsilon > 0$, except for with $\mathsf{negl}(\lambda)$ probability, Algorithm 2 solves the CO task whenever $3f_t + 1 < (1 - \epsilon)n$.*

Proof. Termination is clear from the behavior of the protocol. For Validity, if z is the input of all honest players, then by the Validity of CA, we get that by round 3, all honest players output $commit(z)$. For probabilistic agreement, note that if no honest player outputs according to item (a), then the property holds as the leader is honest w.p. at least $\frac{2}{3}$. Otherwise, Let p be an honest player that output according to item (a), which implies that p observed more than $\frac{1}{3}$ fraction of $commit(z)$ messages for the same value z from the players in com_3. In particular this implies that at least one of those players is honest. Thus by the Agreement property CA, we have that all honest players output either $commit(z)$ or $adopt(z)$. Thus no player in com_3 sends $commit(z')$ for any value $z \neq z'$. In particular this means that no honest player views more than a $\frac{1}{3}$ fraction of $commit(z')$ for any $z' = z$. Thus, all honest players that output according to item (a) agree. Denote that value by z. Now note that w.p. at least $\frac{2}{3}$, ℓ_3 is honest, and if this is the case, then ℓ_3 also output either $commit(z)$ or $adopt(z)$, by the agreement property of CA, and all other honest players output z according to item b in that case. Thus, w.p. at least $\frac{2}{3}$, all honest players agree on the output, as required. □

We denote by L_c, L_{ca} the number of rounds required to run the CO, CA tasks, respectively. We can now describe the generic BA protocol we employ. The following protocol is executed by every honest player p with input z.

Algorithm 3

```
1. For $i = 0, ..., D$:
   (a) If $t \in [(L_c + L_{ca})i, (L_c + L_{ca})i + L_c]$, run as in CO[$i$] with input being the output of CA[$i - 1$] or $z$ if $i = 0$ .
```

(b) If $r \in [(L_c + L_{ca})i + (L_c + 1), (L_c + L_{ca} + 1)i]$ run as in $\mathsf{CA}[i]$ with input being p's output in $\mathsf{CO}[i]$.

(c) let o be the output of $\mathsf{CA}[i]$.

i. If $o = commit(v)$ for some value, then output v as BA decision. Participate in the protocol for one more iteration with inputs to all subroutines being fixed to v.

ii. Else, move on to $i + 1$.

With the above instantiations of CA and CO in mind, we now prove that the above protocol proves Lemma 2.

Proof. For Termination, consider an iteration i of the protocol above. By probabilistic agreement of CO, we have that w.p. at least $\frac{2}{3}$, all honest players agree on the output of $\mathsf{CO}[i]$. Denote it by z. Conditioned on agreement, Validity of CA guarantees that all honest players output $commit(z)$, and thus they all terminate at the end of $\mathsf{CA}[i]$. Thus for every iteration i, w.p. at least $\frac{2}{3}$, all honest players terminate at the end of iteration $i + 1$. Thus, w.h.p. all honest players terminate after $O(\log n)$ iterations. For Validity, consider the case where al players have the same input z. The Validity properties of both CO and CA imply that at end of iteration 1, all honest players output $commit(z)$ from $\mathsf{CA}[1]$ and output z, as required. For Agreement, Let p be the first honest player to output, with output z. I.e. there exists an i such that p output $commit(z)$ from $\mathsf{CA}[z]$, and no honest player has output $commit(z')$ for any z at any iteration $i^* < i$. In particular, this implies, by the consistency of CA, that all other honest players output either $commit(z)$ or $adopt(z)$ from $\mathsf{CA}[i]$. Which implies that all honest players enter iteration $i + 1$ with input z. The Validity of an iteration, proven above, implies thus that by the end of iteration $i + 1$, all honest players output z.

Combining the fact that for every iteration i, w.p. at least $\frac{2}{3}$, all honest players terminate at the end of iteration $i + 1$ with the small size of com_t for all t, we obtain that the protocol halts in $O(1)$ rounds in expectation, and has $O(\lambda)$ uniform communication complexity, in expectation, as required. □

Note that we have essentially proved that Algorithm 3 solves BA whenever the number of honest participants in each round exceeds a $\frac{2}{3}$ fraction. Furthermore, it has expected latency of $O(1)$.

Nakamoto Consensus Note that Nakamoto consensus [32], also gives an efficient protocol with adaptive security. In each round, a leader is elected from a beacon. The leader adds a block to a chain, and consensus is reached on a prefix of the current longest chain. The prefix discards $k = O(\lambda)$ blocks. This implies that the beacon needs to emit $k \cdot \log(n)$ random bits to reach consensus. We note that the classic implementation of Nakamoto does not satisfy the validity condition of BA. This is easily remedied with a single invocation of CA prior to the initiation of the Nakamoto protocol. While CA requires $\Omega(n^2)$ communication, note that out lower bound (Theorem 1) applies even when the protocol has $O(1)$ initial rounds of high communication.

5.2 Adaptive security and low beacon entropy

Next, we showcase a protocol that is secure against an adaptive adversary and does not satisfy Definition 4 (i.e., has low beacon entropy). This, of course, as per Theorem 1, implies that this protocol has to have high communication. Specifically, all n parties send messages in every round. Specifically, we prove the following lemma.

Lemma 5. *For every $\rho < \frac{1}{3}$, and assuming a randomness beacon there exists a ρ-secure consensus protocol against an adaptive adversary, with $O(1)$ expected latency. Furthermore, the protocol does not satisfy Definition 4.*

We once again in this protocol make no use CRS, as we aim to be secure against an adaptive adversary. The protocol Π is going to follow the same structure of Algorithm 3, with the following modifications.

- All players participate in every round of Algorithm 3. In particular, the randomness beacon is not used for committee sampling.
- The random beacon is still being used to sample a uniformly random leader during the CO task. That is the only use of the randomness beacon

Correctness of the protocol and its security against an adaptive adversary are immediate from the proof of correctness for *Algorithm* 3, and the adversary being ρ-bounded for $\rho < \frac{1}{3}$. The last item to prove is the following.

Claim. Algorithm 3 when implemented without committees, satisfies that there exists a constant $c > 0$ such that

$$\sum_{t=0}^{\infty} H_\infty(\mathrm{Tr}^t_{E_\mathcal{A}} \mid \{\mathrm{Tr}^i_{E_\mathcal{A}} \mid 0 \leq i < t\}) \leq c \log n$$

Proof. Note that the only source of entropy in the modified protocol is the leader election in the CO subroutine. Besides that, all content of all messages is a deterministic function of the inputs of the players. Furthermore, we have that for each iteration i, w.p. at least $\frac{2}{3}$, all honest players terminate by the end of iteration $i+1$. Thus, We get that For any adversary $\mathcal{A}$ and any execution E, and for every iteration $i > 0$ w.p. at least $1 - \frac{1}{3^{i-1}}$, $I^t_{E_\mathcal{A}} = \emptyset$ where t is any round during iteration i. In particular we then get that the total min entropy of the transcript of the protocol $\mathit{\Pi}$ during iteration i, for all $i > 1$ is upper bounded by $\log(\frac{3^{i-1}}{3^{i-1}-1})$. For $i = 0, 1$, the transcript is determined by the identity of the random leader, hence both of these iterations provide $\log n$ min entropy each. In total, we get that

$$\sum_{t=0}^{\infty} H_\infty(I^t_{E_\mathcal{A}} | \{Tr^i_{E_\mathcal{A}} \mid 0 \leq i < t\}, \mathsf{CRS}) < 2\log n + \sum_{t=2}^{\infty} \log(\frac{3^{t-1}}{3^{t-1}-1}) < 2\log n + 1$$

as required. □

5.3 Efficiency and low beacon entropy

In this section, we showcase a protocol achieving both low communication and low Randomness beacon entropy usage, which in particular implies that it doesn't have global leader unpredictability (see Definition 4). This is the only protocol in which we make use of the CRS. Specifically, we run Algorithm 3 but instead of using seed_t to select committees and a leader for $\mathsf{CO}[i]$, all the information about the committees and leaders for each iteration are taken from CRS. We assume that CRS is sufficiently long to encode such information. We recall that a static adversary makes its choice of corruption *before* observing the CRS. Formally, we prove the following.

Lemma 6. *For every $\epsilon > 0$ and $\rho < \frac{1-\epsilon}{3}$, and assuming a CRS, there exists a ρ-secure consensus protocol against a static adversary with $O(1)$ expected latency and $O(\lambda)$ uniform communication complexity, in expectation.*

For simplicity, we abuse notation and refer to ℓ_t, com_t for the purposes of this lemma also as the leader and the committee of round t as per described in the CRS.

Observation 3 *Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a static adversary. Then for all t it holds that*

$$\Pr\left[|\mathcal{A}_0(\mathit{\Pi}, t) \cap com_t| \geq \frac{|com_t|}{3}\right] = \mathsf{negl}(\lambda)$$

when com_t is taken from the CRS.

The observation follows from the same arguments as Theorem 2.

Proof. The proof of the lemma follows from the correctness of Algorithm 3 whenever the honest fraction of participating parties in every round exceeds $\frac{2}{3}$. Algorithm 3 was shown to have $O(1)$ expected latency and by the size of com_t for all t we get that the communication complexity of the protocol is uniform $O(\lambda)$ on expectation, as required. □

Acknowledgments. Miranda Christ was supported by a Google CyberNYC grant, an Amazon Research Award, and NSF grants CCF-2312242, CCF-2107187, and CCF-2212233. Joseph Bonneau was supported by a16z crypto and NSF grant CNS-2239975. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Government, a16z, Google, or any other supporting organization.

References

1. Abraham, I., Chan, T.H., Dolev, D., Nayak, K., Pass, R., Ren, L., Shi, E.: Communication complexity of byzantine agreement, revisited. Distributed Comput. **36**(1), 3–28 (2023)
2. Abraham, I., Devadas, S., Dolev, D., Nayak, K., Ren, L.: Synchronous Byzantine Agreement with Expected O(1) Rounds, Expected O(n^2) Communication, and Optimal Resilience. In: Financial Crypto (2019)
3. Ahmadi, M., Ghodselahi, A., Kuhn, F., Molla, A.R.: The cost of global broadcast in dynamic radio networks. Theor. Comput. Sci. **806**, 363–387 (2020)
4. Ahmadi, M., Kuhn, F.: Multi-message broadcast in dynamic radio networks. In: ALGOSENSORS (2016)
5. Alpturer, K., Weinberg, S.M.: Optimal RANDAO manipulation in ethereum. In: AFT (2024)
6. Bar-Joseph, Z., Ben-Or, M.: A tight lower bound for randomized synchronous consensus. In: PODC. ACM (1998)
7. Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable Delay Functions. In: CRYPTO (2018)
8. Boyle, E., Cohen, R., Goel, A.: Breaking the O(n)-bit Barrier: Byzantine Agreement with Polylog Bits Per Party. In: PODC. ACM (2021)
9. Braud-Santoni, N., Guerraoui, R., Huc, F.: Fast byzantine agreement. In: PODC. ACM (2013)
10. Castro, M., Liskov, B., et al.: Practical Byzantine Fault Tolerance. In: OSDI (1999)
11. Chen, J., Micali, S.: Algorand: A secure and efficient distributed ledger. Theor. Comput. Sci. **777**, 155–183 (2019)
12. Choi, K., Manoj, A., Bonneau, J.: Sok: Distributed randomness beacons. In: IEEE Security & Privacy (2023)
13. Chor, B., Merritt, M., Shmoys, D.B.: Simple constant-time consensus protocols in realistic failure models. J. ACM **36**(3), 591–614 (1989)
14. D'Amato, F., Zanolini, L.: A simple single slot finality protocol for ethereum. In: ESORICS (2023)
15. Das, S., Duan, S., Liu, S., Momose, A., Ren, L., Shoup, V.: Asynchronous Consensus without Trusted Setup or Public-Key Cryptography. In: ACM CCS (2024)
16. David, B., Gazi, P., Kiayias, A., Russell, A.: Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In: Eurocrypt (2018)
17. Dolev, D., Reischuk, R.: Bounds on information exchange for byzantine agreement. J. ACM **32**(1), 191–204 (1985)
18. Fitzi, M., Liu-Zhang, C., Loss, J.: A new way to achieve round-efficient byzantine agreement. In: PODC. ACM (2021)
19. Gafni, E., Losa, G.: Brief Announcement: Byzantine Consensus Under Dynamic Participation with a Well-Behaved Majority. In: DISC (2023)
20. Garay, J.A., Kiayias, A.: SoK: A Consensus Taxonomy in the Blockchain Era. In: CT-RSA (2020)
21. Ghinea, D., Goyal, V., Liu-Zhang, C.: Round-optimal byzantine agreement. In: Eurocrypt (2022)
22. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies. In: SOSP (2017)
23. Hajiaghayi, M., Kowalski, D.R., Olkowski, J.: Nearly-optimal consensus tolerating adaptive omissions: Why a lot of randomness is needed? In: PODC. ACM (2024)
24. Kavousi, A., Wang, Z., Jovanovic, P.: SoK: Public Randomness. In: Euro S&P (2024)
25. King, V., Lonargan, S., Saia, J., Trehan, A.: Load balanced scalable byzantine agreement through quorum building, with full information. In: ICDCN (2011)
26. 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)
27. Klonowski, M., Kowalski, D.R., Mirek, J.: Ordered and delayed adversaries and how to work against them on a shared channel. Distributed Comput. **32**(5), 379–403 (2019)
28. Lamport, L., Shostak, R.E., Pease, M.C.: The Byzantine Generals Problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982)
29. Lenstra, A.K., Wesolowski, B.: A random zoo: sloth, unicorn, and trx. Cryptology ePrint Archive, Paper 2015/366 (2015), https://eprint.iacr.org/2015/366
30. Loss, J., Stern, G.: Zombies and ghosts: Optimal byzantine agreement in the presence of omission faults. In: TCC (4). Lecture Notes in Computer Science, vol. 14372, pp. 395–421. Springer (2023)

31. Mostéfaoui, A., Moumen, H., Raynal, M.: Signature-free asynchronous binary byzantine consensus with t $<$ n/3, o(n2) messages, and O(1) expected time. J. ACM **62**(4), 31:1–31:21 (2015)
32. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf (2008)
33. Pease, M.C., Shostak, R.E., Lamport, L.: Reaching agreement in the presence of faults. J. ACM **27**(2), 228–234 (1980)
34. Rabin, M.O.: Transaction protection by beacons. Journal of Computer and System Sciences (1983)
35. Raikwar, M., Gligoroski, D.: SoK: Decentralized randomness beacon protocols. In: Australasian Conference on Information Security and Privacy (2022)
36. Robinson, P., Scheideler, C., Setzer, A.: Breaking the $\tilde{\Omega}(\sqrt{n})$ barrier: Fast consensus under a late adversary. In: SPAA. ACM (2018)
37. Syta, E., Jovanovic, P., Kogias, E.K., Gailly, N., Gasser, L., Khoffi, I., Fischer, M.J., Ford, B.: Scalable bias-resistant distributed randomness. In: IEEE Security & Privacy (2017)