A Lower Bound for Adaptively-Secure Collective Coin-Flipping Protocols

Yael Tauman Kalai
Microsoft Research, 1 Memorial Dr, Cambridge, MA 02142, USA
yael@microsoft.com

Ilan Komargodski[1]
Cornell Tech, 2 W Loop Rd, New York, NY 10044, USA
komargodski@cornell.edu

Ran Raz[2]
Department of Computer Science, Princeton University, Princeton, NJ 08544, USA
ran.raz.mail@gmail.com

Abstract

In 1985, Ben-Or and Linial (Advances in Computing Research '89) introduced the collective coin-flipping problem, where n parties communicate via a single broadcast channel and wish to generate a common random bit in the presence of *adaptive* Byzantine corruptions. In this model, the adversary can decide to corrupt a party in the course of the protocol as a function of the messages seen so far. They showed that the majority protocol, in which each player sends a random bit and the output is the majority value, tolerates $O(\sqrt{n})$ adaptive corruptions. They conjectured that this is optimal for such adversaries.

We prove that the majority protocol is optimal (up to a poly-logarithmic factor) among all protocols in which each party sends a single, *possibly long*, message.

Previously, such a lower bound was known for protocols in which parties are allowed to send only a *single* bit (Lichtenstein, Linial, and Saks, Combinatorica '89), or for symmetric protocols (Goldwasser, Kalai, and Park, ICALP '15).

2012 ACM Subject Classification Theory of computation → Complexity theory and logic

Keywords and phrases Coin flipping, adaptive corruptions, byzantine faults, lower bound

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.34

1 Introduction

In the *collective coin-flipping* problem, introduced by Ben-Or and Linial [7], a set of n computationally unbounded parties, each equipped with a private source of randomness, are required to generate a common random bit. The communication model is the "full information" model [7], where all parties communicate via a single broadcast channel. The goal of the parties is to agree on a common random bit even in the case that some $t = t(n)$ of the parties are faulty and controlled by an adversary whose goal is to bias the output of the protocol in some direction. We say that a protocol Π is *resilient* (or *secure*) to t corruptions if for any adversary $\mathcal{A}$ that makes at most t corruptions it holds that

$$\min\left\{\Pr\left[\text{Output of } \mathcal{A}(\Pi) = 0\right], \Pr\left[\text{Output of } \mathcal{A}(\Pi) = 1\right]\right\} \geq \Omega(1),$$

[1] Supported in part by a Packard Foundation Fellowship and by an AFOSR grant FA9550-15-1-0262.
[2] Research supported by the Simons Collaboration on Algorithms and Geometry and by the National Science Foundation grants No. CCF-1714779 and CCF-1412958.

where "Output of $\mathcal{A}(\Pi)$" is a random variable that corresponds to the output of the protocol Π when executed in the presence of the adversary $\mathcal{A}$.

The adversary is Byzantine, namely, once it corrupts a party, it completely controls it and can send arbitrary messages on its behalf. Usually, two types of Byzantine adversaries are considered, static or adaptive ones. A *static* adversary is an adversary that chooses which parties to corrupt ahead of time, before the protocol begins. An *adaptive* adversary, on the other hand, is allowed to choose which parties to corrupt adaptively in the course of the protocol as a function of the messages seen so far. In the case of static adversaries, collective coin-flipping is well studied and almost matching upper and lower bounds are known; see Section 1.1. However, the case of adaptive adversaries is much less understood. In this work, we focus on the setting of adaptive adversaries.

In the seminal work of Ben-Or and Linial [7], they showed that the majority protocol (in which each party sends a uniformly random bit and the output of the protocol is the majority of the bits sent) is resilient to $O(\sqrt{n})$ adaptive corruptions. Moreover, with $\tilde{\Omega}(\sqrt{n})$ corruptions,[3] one can break the security of this protocol. They conjectured that the majority protocol is *optimal*: any collective coin-flipping protocol is resilient to at most $O(\sqrt{n})$ adaptive corruptions, even if parties send multiple messages, each of which may be long.

The first step towards this conjecture was made by Lichtenstein, Linial, and Saks [19]. They proved that there is no *single-bit* and *single-turn* protocol which is resilient to more than $\tilde{\Omega}(\sqrt{n})$ adaptive corruptions. A single-bit protocol is one in which parties' messages consist of a single bit (perhaps over multiple rounds), and a single-turn protocol is such that each party speaks at most once (perhaps with a long message). More recently, Goldwasser, Kalai, and Park [15] proved another special case of the conjecture: Any *symmetric*[4] single-turn protocol cannot be resilient to more than $\tilde{\Omega}(\sqrt{n})$ adaptive corruptions.

Despite significant efforts, more than three decades after posting the conjecture, fully resolving it remains an intriguing open problem.

Our results. We prove that any n-party collective coin-flipping protocol in which each party sends a single, possibly long, message cannot be secure against more than $t = \tilde{\Omega}(\sqrt{n})$ adaptive corruptions.

▶ **Theorem 1.** *Any n-party single-turn collective coin-flipping protocol is insecure against more than $t = \tilde{\Omega}(\sqrt{n})$ adaptive corruptions.*

As a warm-up, in Section 3, we recover the result of Lichtenstein et al. [19] for single-bit single-turn protocols. Whereas the original proof of [19] is based on combinatorial arguments in extremal set theory, our proof is elementary and uses basic tools from probability theory. A different yet related variant to our simplification was previously given by Cleve and Impagliazzo [11]; see Section 1.1 below.

1.1 Related Work

The *full information model* was introduced by Ben Or and Linial [7] to study the collective coin-flipping problem. Since then, this problem was central in the study of distributed protocols.

[3] Throughout this work, the notation $\tilde{\Omega}$ and $\tilde{O}$ suppresses poly-logarithmic factors.

[4] A symmetric protocol Π is one that is oblivious to the order of its inputs: namely, for any permutation $\pi\colon [n] \to [n]$ of the parties, it holds that $\Pi(r_1, \ldots, r_n) = \Pi(r_{\pi(1)}, \ldots, r_{\pi(n)})$.

Static adversaries. The case of static corruptions has been extensively studied since the introduction of the collective coin-flipping problem. The original work of Ben-Or and Linial [7] showed that a polynomial number (i.e., $O(n^{.63})$) of corrupted parties can be tolerated. Later, Ajtai and Linial [1] showed a different protocol that withstands $O(n/\log^2 n)$ corruptions. For single-round single-bit protocols, in which the global coin is obtained by each party contributing one bit for an n-input predefined Boolean function, Kahn, Kalai and Linial [17] showed that no protocol is resilient to more than $\Omega(n/\log n)$ corruptions. Saks [22] introduced a multi-round protocol called the "Baton Passing" game[5] and showed that it is resilient to $O(n/\log n)$ corruptions. The protocol of Saks was modified by Alon and Naor [3] such that it tolerates a constant fraction of corrupted parties. The optimal resilience of $t = (1/2 - \delta)n$ for any $\delta > 0$ was obtained by Boppana and Narayanan [8] shortly afterwards. Since then the focus has been on improving the explicitness of the protocol, the round complexity, and the bias of the output bit. Two of the most notable results are that of Feige [14] and of Russell, Saks, and Zuckerman [21]. Feige gave an explicit $(\log^* n + O(1/\delta))$-round protocol that tolerates $(1/2 - \delta)n$ corruptions for any constant $\delta > 0$. Russell, Saks, and Zuckerman proved that any protocol that is secure against $\Omega(n)$ corruptions must either have at least $(1/2 - o(1)) \cdot \log^* n$ rounds, or communicate multiple bits per round.

Interestingly, many protocols for collective coin-flipping that consist of more than one round of communication per party, achieve a seemingly stronger goal. In these protocols, first an honest leader is elected and then it outputs a bit that is taken as the protocol outcome. This approach, while being useful for the static case, is unsuitable for adaptive adversaries, since the adversary may always wait for the leader to be elected and then corrupt it.

Adaptive adversaries. The literature on collective coin-flipping with adaptive adversaries is much more scarce. The best known protocol is the majority one suggested by Ben-Or and Linial [7]. Lichtenstein, Linial, and Saks [19] proved that there is no protocol in which each party is allowed to send *one* bit (in total) which is resilient to more than $\Omega(\sqrt{n})$ corruptions. The same lower bound was shown by Goldwasser, Kalai and Park [15] for any single-turn *symmetric* protocol (where each message can be long).

Dodis [12] proved that through "black-box" reductions from non-adaptive collective coin-flipping protocols, it is impossible to tolerate significantly more corruptions than the majority protocol. His definition of "black-box" is rather restricted: It only considers sequential composition of non-adaptive coin-flipping protocols, followed by a (non-interactive, predefined) function operating on the coin-flips thus obtained.

Kalai and Komargodski [18] showed that for any collective coin-flipping protocol in which messages are long there is a collective coin-flipping protocol with the same communication pattern, the same output distribution, the same security guarantees, and where parties send messages of length $\ell = \mathsf{polylog}(n, d)$, where d is the number of rounds in the original protocol. In particular, their transformation guarantees that the resulting protocol is resilient against t adaptive (resp. static) corruptions as long as the original one is resilient against t adaptive (resp. static) corruptions. The transformation is non-uniform, that is, they only show that the required protocol exists.

More types of adversaries. En route to resolving the conjecture of Ben-Or and Linial, stronger types of adversaries were considered.

[5] In this game, each party receiving the baton, passes it to a random party that did not have it yet. The last party having the baton is the leader, and the leader chooses the random bit to be outputted.

Cleve and Impagliazzo [11] studied *re-sampling* adaptive adversaries that can decide whether to intervene in the next message or not *after seeing* it. More precisely, at the i'th round, the adversary, after seeing all the messages exchanged in the first $i-1$ rounds *and* the message to be sent in the current round, can ask to rewind the process back to the beginning of the i'th round and have the i'th message be re-sampled. They showed that for any protocol whose expected output is $1/2$ in an honest execution and in which each party sends just one (possibly long) message, there is an adversary that corrupts a *single* party and biases the expectation of the output of the protocol away from $1/2$ by $\Omega(1/\sqrt{n})$ in some direction.

More recently, Goldwasser, Kalai, and Park [15] studied an even stronger variant called *strong* adaptive adversaries in which the adversary sees all messages sent by honest parties in any given round and, based on the message content, decide whether to corrupt a party or not (and alter its message for that same round). Here, a corruption allows the adversary to send any message on behalf of the party (and not only re-sample it, as in [11]). They proved that any one-round protocol (i.e., all parties talk simultaneously once), in which messages can be arbitrarily long, can tolerate at most $\tilde{O}(\sqrt{n})$ such (strong) adaptive corruptions. They got a similar lower bound in the standard adaptive corruptions model for symmetric protocols.

Fair Coin-Flipping. There is a rich literature on coin-flipping protocols in settings with dishonest majority (and static corruptions), starting from the seminal work of Cleve [10]. In such protocols, the output of the protocol is a random bit, and the requirement is that even in the presence of an adversary, the output cannot be skewed towards 0 or towards 1 except with very small probability.[6]

Cleve [10] proved that for r-round coin-flipping protocol there exists a (static) adversary corrupting $1/2$ of the parties and efficiently biases the output by $\Theta(1/r)$. This lower bound was shown to be tight in the two-party case by Moran, Naor, and Segev [20] and in the three-party case (up to a polylog factor) by Haitner and Tsfadia [16]. In the general n-party case, as long as $n \leq \log\log r$, an almost tight upper bound was given by Buchbinder et al. [9]. When there are less than $(2/3)n$ corruptions, Beimel et al. [6] have constructed an n-party r-round coin-flipping protocol with bias $2^{2^k}/r$, tolerating up to $t = (n+k)/2$ corrupt parties. Alon and Omri [2] constructed an n-party r-round coin-flipping protocol with bias $\tilde{O}(2^{2^n}/r)$, tolerating up to t corrupted parties, for constant n and $t < 3n/4$. Very recently, Beimel et al. [5] gave an improved lower bound in the multi-party case: For any n-party r-round coin-flipping protocol with $n^k \geq r$ for $k \in \mathbb{N}$, there exists an adversary corrupting $n-1$ parties and biases the output of the honest party by $1/(\sqrt{r}\log^k r)$.

1.2 Proof Overview

Since we are in the full information model, we can assume (without loss of generality) that any collective coin-flipping protocol (in which the parties do not have private inputs except for a perfect source of randomness), can be transformed into a protocol in which the honest parties' messages consist only of uniformly random bits. A sketch of this folklore fact appears in [18, Section 4]. Thus, from now on, we assume that each party sends a uniformly random message chosen independently of the previous messages.

[6] We emphasize that in our work, we only require that the adversary cannot skew the output with probability $1-o(1)$, whereas in fair protocols the adversary should not skew the output with probability greater than $1/2+o(1)$.

Concretely, we consider protocols in which each party sends a single message of length ℓ, possibly across n rounds. Such a protocol can be thought of as a complete 2^ℓ-ary tree whose leaves are labeled by 0 and 1, and whose internal nodes are labeled by numbers in $[n]$. If a node is labeled by $i \in [n]$, then we say that the node is owned by party i. (Without loss of generality, we can assume that the order in which the parties send messages is fixed in advance). The protocol starts at the root and at each time step we are at an internal node whose owner samples a random string in $\{0,1\}^\ell$ to determine where the protocol proceeds. The protocol ends once we reach a leaf and the output of the protocol is the bit b corresponding to the label of that leaf.

Let us start with the simpler case where $\ell = 1$. In this case, we present an attacker that biases the outcome of any protocol towards 0 with probability $1 - \mathsf{negl}(n)$, while corrupting at most $\tilde{O}(\sqrt{n})$ parties with probability $1 - o(1)$. (An analogous adversary can bias towards 1 with similar parameters.) The adversary at any point in time computes its possible gain in the expected output of the protocol by corrupting the next party (either to 0 or to 1). If the gain is larger than $\epsilon = \frac{1}{\sqrt{n}\cdot\log^2 n}$, then the adversary corrupts and sends the maliciously chosen bit (that biases the output towards 0). A standard application of Azuma's inequality shows that (with high probability) the influence of the parties that were not corrupted on the expected output of the protocol is negligible, as there are at most n of them and the contribution of each of them is at most $\frac{1}{\sqrt{n}\cdot\log^2 n}$. Intuitively, this means that only the corrupted parties influence the final output of the protocol and since the adversary controls these parties, the adversary succeeds in forcing the output to be 0 with high probability. Moreover, since the adversary gains at least $\frac{1}{\sqrt{n}\cdot\log^2 n}$ in the expected value of the protocol, with the corruption of each party, and the total gain is at most 1, with high probability the number of corruptions is at most $\tilde{O}(\sqrt{n})$. This gives an alternative (elementary) proof for the result of [19]. This is formally proved in Section 3.

The proof for the case $\ell > 1$ is more involved. We define two adversaries $\mathcal{A}_0$ and $\mathcal{A}_1$, where $\mathcal{A}_b$ tries to bias the outcome of the protocol towards b. Here, as opposed to the case $\ell = 1$, only one of the adversaries will be guaranteed to succeed. For $\mathcal{A}_0$, we associate with each node v in the protocol tree three values (we do the same for $\mathcal{A}_1$):

1. α_v : The expectation of the outcome of the protocol in the presence of the adversary $\mathcal{A}_0$, given that the protocol is at node v.
2. c_v^0 : A bit that is 1 if and only if the adversary $\mathcal{A}_0$ corrupts node v.
3. p_v^0 : A "penalty" value that is proportional to the expected number of corruptions made by $\mathcal{A}_0$ from node v onward.

We set these values inductively from the leaves of the protocol tree to the root. For a leaf labeled by $b \in \{0,1\}$, we set $\alpha_v = b$ and $c_v^0 = p_v^0 = 0$.

Going one level up to the parents of the leaves, for each such node we compute the expected α value if we proceed to a random child, compared with the minimal possible α value over all children (this corresponds to the maximal gain possible via corruption). If the possible gain by corruption is larger than $\epsilon = \frac{1}{\sqrt{n}\cdot\log^3 n}$, the adversary will corrupt v, so we set $c_v^0 = 1$, and we update the penalty value by setting it to be $p_v^0 = \epsilon$, to appropriately accommodate for this.

In the next levels, the situation is more complicated as we need to take into account the penalty values. For example, if there is a strategy for corrupting the next message that will increase our chance of outputting 0 by much, but has a high penalty (i.e., will require many corruptions in the future), this move is not always worthwhile for the attacker. So, instead of comparing only the expected outcome of the protocol, we take into account also the penalty.

For every node v, we define $\alpha'_v = \alpha_v + p^0_v$, and compare the expected gain versus the best possible gain *with respect to* α'_v (rather than α_v). Namely, we compute the expected α' value if we proceed to a random child, and compare it to the minimal possible α' value of all children. If this gap is larger than ϵ, the adversary corrupts v, and thus we set $c^0_v = 1$ and we set the penalty value p^0_v to be $p^0_v = p^0_u + \epsilon$, where u is the child that the adversary proceeds to.

The inductive process ends with a triple of values $(\alpha_{\mathsf{root}}, c^0_{\mathsf{root}}, p^0_{\mathsf{root}})$, corresponding to the root node and the adversary $\mathcal{A}_0$. The penalty value p^0_{root} is equal to ϵ times the expected number of corruptions that the adversary $\mathcal{A}_0$ makes. The probability that the protocol outputs 0 with adversary $\mathcal{A}_0$ is $1 - \alpha_{\mathsf{root}}$.

Similarly, we define the adversary $\mathcal{A}_1$ and obtain the values $(\beta_{\mathsf{root}}, c^1_{\mathsf{root}}, p^1_{\mathsf{root}})$, where the penalty value p^1_{root} is equal to ϵ times the expected number of corruptions that the adversary $\mathcal{A}_1$ makes, and the probability that the protocol outputs 1 with adversary $\mathcal{A}_1$ is β_{root}.

It is not possible to prove that both adversaries succeed with high probability (as there are protocols that can only be biased towards one of the two possible values, with the corruption of $\tilde{O}(\sqrt{n})$ parties). Technically, the problem with using an argument similar to the case $\ell = 1$ is that we cannot apply Azuma's inequality as before, because we do not have an upper bound on the absolute value of each variable.

Nevertheless, we are able to prove that at least one of the two adversaries succeeds with high probability, while corrupting $\tilde{O}(\sqrt{n})$ parties. This argument is more complicated, but the main idea is to define another "adversary", "in between" $\mathcal{A}_0$ and $\mathcal{A}_1$. (In the actual proof we refer to that adversary as a random walk). The new adversary is defined similarly to $\mathcal{A}_0$ and $\mathcal{A}_1$, but instead of minimizing α'_v (or maximizing β'_v) it tries to minimize $\beta'_v - \alpha'_v$ (after they were defined by the definitions of $\mathcal{A}_0$ and $\mathcal{A}_1$). Very roughly speaking, since the new adversary is "sandwiched" between $\mathcal{A}_0$ and $\mathcal{A}_1$, we are able to apply Azuma's theorem for the new adversary and to derive a contradiction. Technically, the contradiction is derived by showing that if α'_{root} is not close to 0 and β'_{root} is not close to 1, then the new adversary gets (with high probability) to a leaf that is labeled with neither 0 nor 1.

The full proof is the technical heart of the paper and is given in Section 4.

2 Definitions & Preliminaries

For an integer $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. Throughout the paper, we denote by Π a collective coin-flipping protocol, denote by $n \in \mathbb{N}$ the number of parties participating in the protocol, and denote the parties by $P_1, \ldots, P_n$. We assume that Π, when executed honestly, outputs the bit 0 (and similarly for 1) with probability $\Omega(1)$.

Communication model. The full information model [7] is a synchronous model. Namely, each protocol consists of *rounds* in which parties send messages. There exists a global counter which synchronizes parties in between rounds but they are asynchronous within a round. The parties communicate via a broadcast channel.

We define two restricted types of protocols: single-bit and single-turn.

▶ **Definition 2** (Single-bit protocol)**.** We say that a protocol is a single-bit protocol for n parties if this protocol is executed in rounds such that in each round each party sends a single random bit.

▶ **Definition 3** (Single-turn protocol)**.** We say that a protocol is a single-turn protocol for n parties if this protocol is executed in n rounds such that party P_i sends a single (possibly long) message at round i.

The above two restricted families of protocols can be naturally described by a game tree (of arity two in the single-bit case and bigger arity in the single-turn case) whose leaves are labeled by 0 and 1, and whose internal nodes (including the root) are labeled by numbers in $[n]$.

Without loss of generality, we restrict our attention to public-coin protocols.

▶ **Definition 4** (Public-coin protocol)**.** A protocol is *public-coin* if each honest party broadcasts all of the randomness he generates (i.e., his "local coin-flips"), and does not send any other messages.

Corruption model. We consider the Byzantine model, where a bound $t = t(n) \leq n$ is specified, and the adversary is allowed to corrupt up to t parties. The adversary can see the entire transcript (i.e., all the messages sent thus far), has full control over all the corrupted parties, and can broadcast any messages on their behalf. Moreover, the adversary has control over the order of the messages sent within each round of the protocol (i.e., "rushing").

Within this model, two main types of adversaries were considered in the literature: *static* adversaries, who need to specify the parties they corrupt *before* the protocol begins, and *adaptive* adversaries, who can corrupt the parties *adaptively* based on the transcript so far. We focus on adaptive adversaries

▶ **Definition 5** (Adaptive adversary)**.** Within each round, the adversary chooses parties one-by-one to send their messages; and he can perform corruptions at any point during this process based on the messages sent thus far and the protocol specification.

Security. The security of a collective coin-flipping protocol is usually measured by the extent to which an adversary can, by corrupting a subset of parties, bias the protocol outcome towards his desired bit.

▶ **Definition 6** (ϵ-security)**.** Fix $\epsilon = \epsilon(n)$ and $t = t(n)$. A coin-flipping protocol Π is ϵ-secure against t adaptive corruptions if for all $n \in \mathbb{N}$, it holds that for any adaptive adversary $\mathcal{A}$ that corrupts at most t parties,

$$\min\left\{\Pr\left[\text{Output of } \mathcal{A}(\Pi) = 0\right], \Pr\left[\text{Output of } \mathcal{A}(\Pi) = 1\right]\right\} \geq \epsilon(n),$$

where "Output of $\mathcal{A}(\Pi)$" is a random variable that corresponds to the output of the protocol Π when executed in the presence of the adversary $\mathcal{A}$.

We next define a secure protocol as one where an adversary cannot "almost always" get the outcome he wants.

▶ **Definition 7** (Security)**.** A coin-flipping protocol is secure against $t = t(n)$ corruptions if it is ϵ-secure against t corruptions for some constant $\epsilon \in (0, 1)$.

2.1 Azuma's Inequality

We state Azuma's inequality which is extensively used in our proofs. This formulation is standard and can be found, for example, in Alon-Spencer [4] and in Dubhashi-Panconesi [13].

▶ **Theorem 8.** *Let $X_1, \ldots, X_N$ be random variables, such that for every $i \in [N]$, $|X_i| \leq \epsilon_i$. If for every $i \in [N]$ it holds that $\mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}] \leq 0$, then for any $s \geq 0$,*

$$\Pr\left[\sum_{i=1}^{N} X_i \geq s\right] \leq 2 \cdot e^{-\frac{s^2}{2\sum_{i=1}^{N} \epsilon_i^2}}$$

Similarly, if for every $i \in [N]$ it holds that $\mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}] \geq 0$, then for any $s \geq 0$,

$$\Pr\left[\sum_{i=1}^{N} X_i \leq -s\right] \leq 2 \cdot e^{-\frac{s^2}{2\sum_{i=1}^{N} \epsilon_i^2}}$$

3 A Lower Bound for Single-Bit Single-Turn Protocols

In this section, we give a simplified proof for the following theorem, originally proved in [19]

▶ **Theorem 9** ([19]). *There does not exist a single-bit single-turn collective coin-flipping protocol that is resilient to more than $\tilde{\Omega}(\sqrt{n})$ adaptive corruptions.*

Proof. Fix any single-bit single-turn collective coin-flipping protocol Π. Consider the binary protocol tree of depth n corresponding to Π. We construct an adversary $\mathcal{A}_0$ that with probability $1 - o(1)$, biases the outcome towards 0 while corrupting at most $\tilde{O}(\sqrt{n})$ players.[7]

For each node v in the protocol tree, we associate a sequence of bits $b_1, \ldots, b_i$ that lead to it from the root of the tree, and a value α_v which stands for the probability that the outcome of the protocol is 0, when executed *honestly* starting from the node v. Namely, $\alpha_v \triangleq \Pr[\Pi^v = 0]$, where Π^v is a random variable that corresponds to the output of the protocol Π when executed honestly starting from node v. Let $p_0 \triangleq \Pr[\Pi^{\mathsf{root}} = 0] \geq \Theta(1)$ be the probability that the protocol, executed honestly from the root, outputs 0. Further, observe that for every *leaf* v that is labeled by $b \in \{0, 1\}$, it holds that $\alpha_v = 1 - b$.

Let $\epsilon \triangleq \frac{1}{\sqrt{n} \cdot \log^2 n}$. Given that the protocol is in node v, the adversary $\mathcal{A}_0$ computes two values

$$\alpha_v^{\mathsf{min}} = \min\{\alpha_{v0}, \alpha_{v1}\} \qquad \text{and} \qquad \alpha_v^{\mathsf{max}} = \max\{\alpha_{v0}, \alpha_{v1}\},$$

where α_{v0} is the value associated with the left child of v and α_{v1} is the value associated with the right child of v. Note that $\alpha_v = (\alpha_{v0} + \alpha_{v1})/2$. If $\alpha_v \geq \alpha_v^{\mathsf{min}} + \epsilon$ (or, equivalently, $\alpha_v \leq \alpha_v^{\mathsf{max}} - \epsilon$), then the adversary corrupts the party that is associated with node v and sends $b \in \{0, 1\}$ such that $\alpha_{vb} > \alpha_{v\bar{b}}$ (where $\bar{b} = 1 - b$). Otherwise, if $\alpha_v^{\mathsf{max}} - \epsilon < \alpha_v < \alpha_v^{\mathsf{min}} + \epsilon$, then the adversary $\mathcal{A}_0$ does not corrupt the corresponding party and lets it send a random bit. This completes the description of the adversary.

We next show that with overwhelming probability over the execution of the protocol with the adversary $\mathcal{A}_0$, the leaf with which the protocol concludes is a leaf that is labeled with 0. In addition, with overwhelming probability, the number of corruptions along the way is bounded by $1/\epsilon$.

Let $(b_1, \ldots, b_n) \in \{0, 1\}^n$ be a random variable corresponding to the n bits sent during the execution of the protocol $\Pi_{\mathcal{A}_0}$. Namely, if $\mathcal{A}_0$ corrupts the party sending the i'th bit in the protocol Π, given that the previous $i - 1$ bits sent were $(b_1, \ldots, b_{i-1})$, and sends the bit $b^* \in \{0, 1\}$ on its behalf, then we set $b_i = b^*$. Otherwise, if $\mathcal{A}_0$ does not corrupt this party, then b_i is randomly chosen in $\{0, 1\}$. Every prefix of the n bits $b_1, \ldots, b_n$ sent during the course of the protocol, corresponds to a node v in the protocol tree. Thus, we can write $\alpha_{b_1,\ldots,b_i}$ for α_v, where the vertex v corresponds to the path $b_1, \ldots, b_i$ from the root to v in the protocol tree. Let δ_i be a random variable defined as

$$\delta_i \triangleq \alpha_{b_1,\ldots,b_i} - \alpha_{b_1,\ldots,b_{i-1}}.$$

[7] One can analogously construct an adversary $\mathcal{A}_1$ that with probability $1 - o(1)$, biases the outcome towards 1 while corrupting at most $\tilde{O}(\sqrt{n})$ players.

Denote by $I \subseteq [n]$ the set of indices in which the adversary $\mathcal{A}_0$ corrupts the corresponding party. It holds that

$$\sum_{i=1}^{n} \delta_i = \sum_{i \in I} \delta_i + \sum_{i \notin I} \delta_i = \alpha_{b_1,\ldots,b_n} - \alpha_{\mathsf{root}}. \tag{1}$$

We first argue that with overwhelming probability $\sum_{i \notin I} \delta_i \leq o(1)$.

▶ **Claim 10.** $\Pr\left[\left|\sum_{i \notin I} \delta_i\right| \geq \frac{1}{\log n}\right] \leq \mathsf{negl}(n)$.

Proof. Define n random variables $X_1, \ldots, X_n$ as follows: For every $i \in I$ we set $X_i = 0$, and for every $i \notin I$ we define $X_i = \delta_i$. Note that for every $i \in [n]$, it holds that $|X_i| \leq \epsilon$ and

$$\mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}] = 0.$$

Thus, by Azuma's inequality, for any $s > 0$,

$$\Pr\left[\left|\sum_{i=1}^{n} X_i\right| \geq s\right] \leq 4 \cdot e^{-\frac{s^2}{2n\epsilon^2}}.$$

Setting $s = \epsilon \cdot \sqrt{n} \cdot \log n = \frac{1}{\log n}$, we conclude that

$$\Pr\left[\left|\sum_{i=1}^{n} X_i\right| \geq \frac{1}{\log n}\right] \leq \mathsf{negl}(n). \qquad \blacktriangleleft$$

We condition on the event that $\left|\sum_{i \notin I} \delta_i\right| \leq \frac{1}{\log n}$ occurs. Also, recall that $\alpha_{\mathsf{root}} = \Theta(1)$. Plugging these into Equation (1), we get that

$$\alpha_{b_1,\ldots,b_n} \geq \alpha_{\mathsf{root}} + \sum_{i \in I} \delta_i - \frac{1}{\log n}.$$

By the definition of $\mathcal{A}_0$, whenever it corrupts an index i, it causes δ_i to be positive. Thus,

$$\alpha_{b_1,\ldots,b_n} \geq \alpha_{\mathsf{root}} - \frac{1}{\log n}.$$

This implies that $\alpha_{b_1,\ldots,b_n} = 1$ since $\alpha_{b_1,\ldots,b_n} \in \{0, 1\}$ and $\alpha_{\mathsf{root}} \geq \Omega(1)$.

We proceed with the bound on the number of corruptions made by $\mathcal{A}_0$. By Equation (1), the fact that $\alpha_{\mathsf{root}} \in (0, 1)$, and that with overwhelming probability $\left|\sum_{i \notin I} \delta_i\right| \leq \frac{1}{\log n}$ and $\alpha_{b_1,\ldots,b_n} = 1$, it holds that (with overwhelming probability)

$$\sum_{i \in I} \delta_i = \alpha_{b_1,\ldots,b_n} - \alpha_{\mathsf{root}} - \sum_{i \notin I} \delta_i \leq 1.$$

Since for each $i \in I$, it holds that $\delta_i \geq \epsilon$, the number of corruptions is bounded by $1/\epsilon$, as required. ◀

4 A Lower Bound for Single-Turn Protocols

In this section, we prove our lower bound for single-turn collective coin-flipping protocols.

▶ **Theorem 11.** *There does not exist a single-turn collective coin-flipping protocol that is resilient to more than $\tilde{\Omega}(\sqrt{n})$ adaptive corruptions.*

Proof. Fix any single-turn collective coin-flipping protocol Π. Since we are in the full information model, we can assume without loss of generality that the protocol is public-coin (see Definition 4). Namely, each player sends a *random* message from some universe $\{0,1\}^\ell$. We denote $L \triangleq 2^\ell$.

Consider the L-ary protocol tree corresponding to Π. We define two adversaries $\mathcal{A}_0$ and $\mathcal{A}_1$, where $\mathcal{A}_0$ tries to bias the output towards 0 and $\mathcal{A}_1$ tries to bias the output towards 1. We prove that at least one of these adversaries succeeds with probability $1 - o(1)$ while corrupting at most $\sqrt{n} \cdot \mathsf{polylog}(n)$ players.

To this end, we associate with each node v in the protocol tree, three pairs of values

$$(\alpha_v, \beta_v),\ (c_v^0, c_v^1),\ \text{and}\ (p_v^0, p_v^1).$$

Intuitively, α_v is the expectation of the outcome of the protocol in the presence of the adversary $\mathcal{A}_0$, given that we are at node v, and β_v is the expectation of the outcome of the protocol in the presence of the adversary $\mathcal{A}_1$, given that we are at node v. The pair (c_v^0, c_v^1) is a pair of bits, where $c_v^0 = 1$ if and only if $\mathcal{A}_0$ corrupts node v, and $c_v^1 = 1$ if and only if $\mathcal{A}_1$ corrupts node v.[8] The pair (p_v^0, p_v^1) are a pair of "penalty" values . Intuitively, the penalty p_v^0 (resp. p_v^1) is proportional to the expected number of corruptions the adversary $\mathcal{A}_0$ (resp. $\mathcal{A}_1$) does, from node v onwards.

The penalty values $\{p_v^0\}_{v \in V}$, along with the values $\{\alpha_v\}_{v \in V}$, are used by the adversary $\mathcal{A}_0$ to decide which nodes to corrupt (i.e., for which nodes v to set $c_v^0 = 1$). Similarly, the penalty values $\{p_v^1\}_{v \in V}$, along with the values $\{\beta_v\}_{v \in V}$, are used by the adversary $\mathcal{A}_1$ to decide which nodes to corrupt (i.e., for which nodes v to set $c_v^1 = 1$).

Formally, the values (α_v, β_v), (c_v^0, c_v^1), and (p_v^0, p_v^1) are defined by induction starting from the leaves. For any leaf v labeled by 0 we define $\alpha_v = \beta_v = 0$, and for any leaf v labeled by 1 we define $\alpha_v = \beta_v = 1$. For all leaves v we define $c_v^0 = c_v^1 = 0$ and $p_v^0 = p_v^1 = 0$.

Let $k = \sqrt{n} \cdot \log^3 n$ and let $\epsilon = \frac{1}{k}$. For a non-leaf node v, suppose that its L children are associated with

$$\{(\alpha_i, \beta_i)\}_{i=1}^L \text{ and } \{(p_i^0, p_i^1)\}_{i=1}^L.$$

For every $i \in [L]$, define

$$\alpha'_i = \alpha_i + p_i^0 \text{ and } \beta'_i = \beta_i - p_i^1.$$

Let

$$\alpha_{\mathsf{avg}} \triangleq \frac{1}{L} \cdot \sum_{i=1}^{L} \alpha_i \quad , \quad \alpha'_{\mathsf{avg}} \triangleq \frac{1}{L} \cdot \sum_{i=1}^{L} \alpha'_i \quad , \quad \alpha'_{\mathsf{min}} \triangleq \min\{\alpha'_1, \ldots, \alpha'_L\}$$

and let

$$\beta_{\mathsf{avg}} \triangleq \frac{1}{L} \cdot \sum_{i=1}^{L} \beta_i \quad , \quad \beta'_{\mathsf{avg}} \triangleq \frac{1}{L} \cdot \sum_{i=1}^{L} \beta'_i \quad , \quad \beta'_{\mathsf{max}} = \max\{\beta'_1, \ldots, \beta'_L\}$$

If $\alpha'_{\mathsf{min}} \leq \alpha'_{\mathsf{avg}} - \epsilon$, then set $c_v^0 = 1$. In this case, if the protocol arrives at node v, then the adversary $\mathcal{A}_0$ corrupts node v and proceeds to its child i^* with minimal α'; i.e., $i^* = \mathsf{argmin}_{\mathsf{i} \in \mathsf{child(v)}}\{\alpha'_\mathsf{i}\}$, and we set $\alpha_v = \alpha_{i^*}$ and $p_v^0 = p_{i^*}^0 + \epsilon$. Otherwise, set $c_v^0 = 0$. In this

[8] When we say that an adversary corrupts node v we mean that it corrupts the party associated with node v.

case, the adversary $\mathcal{A}_0$ does not corrupt node v, and we set $\alpha_v = \alpha_{\mathsf{avg}}$ and $p_v^0 = \frac{1}{L} \cdot \sum_{i=1}^{L} p_i^0$. We denote

$$\alpha'_v = \alpha_v + p_v^0.$$

Similarly, if $\beta'_{\mathsf{max}} \geq \beta'_{\mathsf{avg}} + \epsilon$, then set $c_v^1 = 1$. In this case, if the protocol arrives at node v, then the adversary $\mathcal{A}_1$ corrupts node v and proceeds to its child i^* with maximal β'; i.e., $i^* = \mathsf{argmax}_{\mathsf{i} \in \mathsf{child(v)}}\{\beta'_{\mathsf{i}}\}$, and we set $\beta_v = \beta_{i^*}$ and $p_v^1 = p_{i^*}^1 + \epsilon$. Otherwise, set $c_v^1 = 0$. In this case, the adversary $\mathcal{A}_1$ does not corrupt node v, and we set $\beta_v = \beta_{\mathsf{avg}}$ and $p_v^1 = \frac{1}{L} \cdot \sum_{i=1}^{L} p_i^1$. We denote

$$\beta'_v = \beta_v - p_v^1.$$

In what follows, we denote by α_{root} and β_{root} the α and β values of the root, respectively. Similarly, we denote by α'_{root} and β'_{root} the α' and β' values of the root, respectively. We denote by p^0_{root} and p^1_{root} the penalty values of the root.

The following claim follows immediately from the definition of p^0_{root} and p^1_{root}.

▶ **Claim 12.** *For every $b \in \{0,1\}$, it holds that*

$$p^b_{\mathsf{root}} = \frac{1}{k} \cdot \mathbb{E}[\#\ \textit{of corruptions}\ \mathcal{A}_b\ \textit{makes}].$$

In what follows, we denote by $\Pi_{\mathcal{A}_0}$ the random variable which is the outcome of protocol Π with adversary $\mathcal{A}_0$, and similarly we denote by $\Pi_{\mathcal{A}_1}$ the random variable which is the outcome of protocol Π with adversary $\mathcal{A}_1$ (in both $\Pi_{\mathcal{A}_0}$ and $\Pi_{\mathcal{A}_1}$ the randomness is over the coin tosses of the honest players). In order to complete the proof of the theorem it suffices to prove the following two lemmas.

▶ **Lemma 13.** $\Pr[\Pi_{\mathcal{A}_0} = 0] = 1 - \alpha_{\mathsf{root}}$ *and* $\Pr[\Pi_{\mathcal{A}_1} = 1] = \beta_{\mathsf{root}}$.

▶ **Lemma 14.** $\alpha'_{\mathsf{root}} = o(1)$ *or* $\beta'_{\mathsf{root}} = 1 - o(1)$.

The reason why these two lemmas suffice is that for any node v in the protocol tree (and in particular for the root), $\alpha_v \leq \alpha'_v$ and $\beta_v \geq \beta'_v$. Thus, the two lemmas imply that either

$$\Pr[\Pi_{\mathcal{A}_0} = 0] = 1 - o(1) \qquad \text{or} \qquad \Pr[\Pi_{\mathcal{A}_1} = 1] = 1 - o(1).$$

Moreover, by definition, $\alpha'_{\mathsf{root}} = \alpha_{\mathsf{root}} + p^0_{\mathsf{root}}$ and $\beta'_{\mathsf{root}} = \beta_{\mathsf{root}} - p^1_{\mathsf{root}}$. Thus, if $\alpha'_{\mathsf{root}} = o(1)$ then Claim 12, together with the fact that $\alpha_{\mathsf{root}} \geq 0$ (see Lemma 13), implies that the adversary $\mathcal{A}_0$ is expected to make only $o(k)$ corruptions. By Markov's inequality $\mathcal{A}_0$ makes $o(k)$ corruptions with probability $1 - o(1)$. Similarly, if $\beta'_{\mathsf{root}} = 1 - o(1)$ then Claim 12, together with the fact that $\beta_{\mathsf{root}} \leq 1$ (see Lemma 13), implies that the adversary $\mathcal{A}_1$ is expected to make only $o(k)$ corruptions. By Markov's inequality $\mathcal{A}_1$ makes $o(k)$ corruptions with probability $1 - o(1)$. Since we set $k = \sqrt{n} \cdot \log^3 n$, this completes the proof.

We proceed with the proof of Lemma 13, followed by the proof of Lemma 14.

4.1 Proof of Lemma 13

We prove the more general statement that for *any* node v in the protocol tree, the probability that $\Pi_{\mathcal{A}_0} = 0$ (respectively, $\Pi_{\mathcal{A}_1} = 1$), conditioned on the event that the protocol arrives at node v, is $1 - \alpha_v$ (respectively, β_v). To this end, for any node v in the protocol tree, denote by Π^v the protocol execution starting from node v. We prove that for every node v,

$$\Pr[\Pi^v_{\mathcal{A}_0} = 0] = 1 - \alpha_v \text{ and } \Pr[\Pi^v_{\mathcal{A}_1} = 1] = \beta_v. \tag{2}$$

The proof is by induction from the leaves to the root. For leaf nodes, Equation (2) holds trivially. Suppose that Equation (2) holds for nodes at layer $d+1$, and we shall prove that it holds for nodes at layer d. To this end, fix a node v at layer d, and denote its L (layer $d+1$) children by $u_1, \ldots, u_L$. To be consistent with our previous notation, we denote $\alpha_i \triangleq \alpha_{u_i}$ and let $\alpha_{\mathsf{avg}} = \frac{1}{L}\sum_{i=1}^{L} \alpha_i$. We show that $\Pr[\Pi^v_{\mathcal{A}_0} = 0] = 1 - \alpha_v$ and mention that the proof that $\Pr[\Pi^v_{\mathcal{A}_1} = 1] = \beta_v$ is analogous.

We distinguish between two cases:

- **Case 1: $c^0_v = 0$.** This case corresponds to the case where $\mathcal{A}_0$ does not corrupt node v. In this case,

$$\Pr[\Pi^v_{\mathcal{A}_0} = 0] = \frac{1}{L}\sum_{i=1}^{L} \Pr[\Pi^{u_i}_{\mathcal{A}_0} = 0] = \frac{1}{L}\sum_{i=1}^{L}(1-\alpha_i) = 1 - \alpha_{\mathsf{avg}} = 1 - \alpha_v,$$

  where the second equality follows from the induction assumption, and the other equalities follow from the definition of $\mathcal{A}_0$, α_{avg} and α_v.
- **Case 2: $c^0_v = 1$.** This case corresponds to the case where $\mathcal{A}_0$ corrupts node v. We denote by i^* the child with minimal α'. In this case,

$$\Pr[\Pi^v_{\mathcal{A}_0} = 0] = \Pr[\Pi^{u_{i^*}}_{\mathcal{A}_0} = 0] = (1 - \alpha_{i^*}) = 1 - \alpha_v,$$

  where the second equality follows from our induction assumption, and the other equalities follow from the definition of $\mathcal{A}_0$, and α_v.

This completes the proof of the lemma.

4.2 Proof of Lemma 14

Suppose towards contradiction that there exists a constant $c > 0$ such that $\alpha'_{\mathsf{root}} > c$ and $\beta'_{\mathsf{root}} < 1 - c$. We prove that at each layer of the circuit there exists a node v for which $\alpha'_v > c - o(1)$ and $\beta'_v < 1 - c + o(1)$. This would imply a contradiction since at each leaf v it holds that either $\alpha'_v = 0$ or $\beta'_v = 1$.

We define a random walk on the protocol tree from the root to the leaves. Since Π is a single turn protocol on n players, the protocol tree is of depth n. We denote the nodes on the walk by $v_0, v_1, \ldots, v_n$, where v_0 is the root and v_n is a leaf. The random walk is defined as follows:

1. Let V_1 be the set of all nodes v such that for every child $u \in \mathsf{child}(v)$ it holds that

$$|\alpha'_u - \alpha'_v| \leq \epsilon \cdot \log n \qquad \text{and} \qquad |\beta'_u - \beta'_v| \leq \epsilon \cdot \log n.$$

   If we are at node $v_i \in V_1$, then v_{i+1} is a random child of v_i.
2. Let V_2 be the set of all nodes that are not in V_1. If $v_i \in V_2$, then choose a child $v_{i+1} \in \mathsf{child}(v_i)$ that minimizes the value $\beta'_u - \alpha'_u$. Namely, $v_{i+1} = \mathsf{argmin}_{u \in \mathsf{child}(v_i)} \{\beta'_u - \alpha'_u\}$.

Recall that in order to get a contradiction, it suffices to prove that with overwhelming probability, $\alpha'_{v_n} \geq c - o(1)$ and $\beta'_{v_n} \leq 1 - c + o(1)$. To this end, we define n random variables $X_1, \ldots, X_n$, and n random variables $Y_1, \ldots, Y_n$, as follows:

$$X_{i+1} = \alpha'_{v_{i+1}} - \alpha'_{v_i} \qquad \text{and} \qquad Y_{i+1} = \beta'_{v_{i+1}} - \beta'_{v_i}.$$

Notice that

$$\alpha'_{v_n} = \alpha'_{v_0} + \sum_{i=1}^{n} X_i \qquad \text{and} \qquad \beta'_{v_n} = \beta'_{v_0} + \sum_{i=1}^{n} Y_i.$$

To get a contradiction it suffices to prove that for any constant $t > 0$, with overwhelming probability (over the random walk)

$$\sum_{i=1}^{n} X_i \geq -t \qquad \text{and} \qquad \sum_{i=1}^{n} Y_i \leq t. \tag{3}$$

To this end, we partition the set $[n]$ into two sets $I_1, I_2 \subseteq [n]$, such that $i \in I_b$ if and only if $v_i \in V_b$ for $b \in \{0,1\}$ and $i \in [n]$ and where V_1 and V_2 are the sets defined above. Namely,

$$I_1 = \{i : v_i \in V_1\} \qquad \text{and} \qquad I_2 = \{i : v_i \in V_2\}.$$

In order to prove Equation (3), it suffices to prove the following two claims.

▶ **Claim 15.** *For any constant $t > 0$, with overwhelming probability (over the random walk),*

$$\sum_{i \in I_1} X_i \geq -t \qquad \textit{and} \qquad \sum_{i \in I_1} Y_i \leq t.$$

▶ **Claim 16.** *For any constant $t > 0$, with overwhelming probability (over the random walk),*

$$\sum_{i \in I_2} X_i \geq -t \qquad \textit{and} \qquad \sum_{i \in I_2} Y_i \leq t. \tag{4}$$

We start by stating the following claim which we will use in the proofs of Claims 15 and 16.

▶ **Claim 17.** *For every node v in the protocol tree,*

$$\alpha'_v \leq \alpha'_{\mathsf{avg}} \qquad \textit{and} \qquad \beta'_v \geq \beta'_{\mathsf{avg}},$$

where α'_{avg} denotes the average of the values of $\{\alpha'_u\}_{u \in \mathsf{child}\{v\}}$ over the children of v, and β'_{avg} denotes the average of the values of $\{\beta'_u\}_{u \in \mathsf{child}\{v\}}$ over the children of v.

Proof of Claim 17. Fix a node v in the protocol tree. We show that $\alpha'_v \leq \alpha'_{\mathsf{avg}}$ and note that the proof that $\beta'_v \geq \beta'_{\mathsf{avg}}$ is analogous.

If $c^0_v = 0$, then $\alpha'_v = \alpha'_{\mathsf{avg}}$ and the claim holds. Suppose that $c^0_v = 1$. In this case, $\alpha'_v = \alpha'_{\mathsf{min}} + \epsilon$, where $\alpha'_{\mathsf{min}} = \min_{u \in \mathsf{child}\{v\}}\{\alpha'_u\}$ is the minimal value of α' over all the children of v. Also, by definition, $\alpha'_{\mathsf{min}} \leq \alpha'_{\mathsf{avg}} - \epsilon$. Thus, $\alpha'_v \leq \alpha'_{\mathsf{avg}} - \epsilon + \epsilon = \alpha'_{\mathsf{avg}}$, as desired. ◀

Proof of Claim 15. By definition, for every $i \in I_1$, $|X_i|, |Y_i| \leq \epsilon \cdot \log n$. Claim 17 implies that

$$\mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}] \geq 0 \qquad \text{and} \qquad \mathbb{E}[Y_i \mid Y_1, \ldots, Y_{i-1}] \leq 0.$$

We extend the series of random variables $(X_i)_{i \in I_1}$ and $(Y_i)_{i \in I_1}$, and define two sequences of n random variables

$$(X'_1, \ldots, X'_n) \qquad \text{and} \qquad (Y'_1, \ldots, Y'_n)$$

such that for every $i \in [n]$ it holds that

$$X'_i = \begin{cases} X_i & \text{if } i \in I_1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad Y'_i = \begin{cases} Y_i & \text{if } i \in I_1 \\ 0 & \text{otherwise.} \end{cases}$$

Note that (by Claim 17) it still holds that for every $i \in [n]$,

$$\mathbb{E}[X'_i \mid X'_1, \ldots, X'_{i-1}] \geq 0 \qquad \text{and} \qquad \mathbb{E}[Y'_i \mid Y'_1, \ldots, Y'_{i-1}] \leq 0.$$

Thus, by Azuma's inequality (see Theorem 8), for any real $s > 0$,

$$\Pr\left[\sum_{i=1}^{n} X'_i \leq -s\right] \leq 2 \cdot e^{-\frac{s^2}{2n(\epsilon \cdot \log n)^2}} \qquad \text{and} \qquad \Pr\left[\sum_{i=1}^{n} Y'_i \geq s\right] \leq 2 \cdot e^{-\frac{s^2}{2n(\epsilon \cdot \log n)^2}}.$$

By definition $\sum_{i \in I_1} X_i = \sum_{i=1}^{n} X'_i$ and $\sum_{i \in I_1} Y_i = \sum_{i=1}^{n} Y'_i$ and thus

$$\Pr\left[\sum_{i \in I_1} X_i \leq -\epsilon \cdot \sqrt{n} \cdot \log^2 n\right] = \mathsf{negl}(n) \quad \text{and} \quad \Pr\left[\sum_{i \in I_1} Y_i \geq \epsilon \cdot \sqrt{n} \cdot \log^2 n\right] = \mathsf{negl}(n).$$

Since we set $\epsilon = \frac{1}{k} = \frac{1}{\sqrt{n} \cdot \log^3 n}$, we have that $\epsilon \cdot \sqrt{n} \cdot \log^2 n = o(1)$, which completes the proof. ◀

We proceed with the proof of Claim 16. In the proof, we make use of the following two claims.

▶ **Claim 18.** *For any node v in the protocol tree and for any $u \in \mathsf{child}(v)$,*

$$\beta'_u \leq \beta'_v + \epsilon \qquad \textit{and} \qquad \alpha'_u \geq \alpha'_v - \epsilon.$$

Proof. Fix any node v in the protocol tree and fix any child $u \in \mathsf{child}(v)$. We prove that $\beta'_u \leq \beta'_v + \epsilon$. The proof that $\alpha'_u \geq \alpha'_v - \epsilon$ is analogous and thus omitted.

We distinguish between two cases. First, if $c_v^1 = 0$, then $\beta'_v = \beta'_{\mathsf{avg}}$ and all the children of v have β' which is at most $\beta'_{\mathsf{avg}} + \epsilon$ which implies that $\beta'_u \leq \beta'_v + \epsilon$. Second, if $c_v^1 = 1$, then $\beta'_v = \beta'_{\mathsf{max}} - \epsilon$, where $\beta'_{\mathsf{max}} = \max_{u \in \mathsf{child}(v)}\{\beta'_u\}$ is the maximal β' of all the children of v. This also implies that $\beta'_u \leq \beta'_v + \epsilon$. ◀

▶ **Claim 19.** *For every node v in the protocol tree, it holds that $\beta'_v \geq \alpha'_v$.*

Proof. The proof is by induction from the leaves to the root. For any leaf v, it holds that $\beta'_v = \alpha'_v$ by definition, and in particular $\beta'_v \geq \alpha'_v$. Suppose that $\beta'_v \geq \alpha'_v$ holds for every node v in layer $d+1$ and we prove that it holds for every node in layer d.

To this end, fix any node v in layer d. Suppose that its L children (in layer $d+1$) are associated with values $\{(\alpha'_i, \beta'_i)\}_{i=1}^{L}$, and denote

$$\alpha'_{\mathsf{avg}} \triangleq \frac{1}{L}\sum_{i=1}^{L} \alpha'_i \qquad \text{and} \qquad \beta'_{\mathsf{avg}} \triangleq \frac{1}{L}\sum_{i=1}^{L} \beta'_i.$$

The induction assumption implies that $\beta'_{\mathsf{avg}} \geq \alpha'_{\mathsf{avg}}$. This, together with Claim 17, implies that

$$\beta'_v \geq \beta'_{\mathsf{avg}} \geq \alpha'_{\mathsf{avg}} \geq \alpha'_v,$$

as desired. ◀

Proof of Claim 16. We first show that for every $i \in I_2$,

$$\beta'_{v_{i+1}} - \alpha'_{v_{i+1}} \leq \left(\beta'_{v_i} - \alpha'_{v_i}\right) - \epsilon \cdot (\log n - 1). \tag{5}$$

Fix any $v_i \in V_2$. By definition of V_2, there exists a child $u \in \mathsf{child}(v_i)$ such that

$$\left|\alpha'_u - \alpha'_{v_i}\right| \geq \epsilon \cdot \log n \qquad \text{or} \qquad \left|\beta'_u - \beta'_{v_i}\right| \geq \epsilon \cdot \log n.$$

Claim 18 implies that there exists a child $u \in \mathsf{child}(v_i)$ such that

$$\alpha'_u \geq \alpha'_{v_i} + \epsilon \cdot \log n \qquad \text{or} \qquad \beta'_u \leq \beta'_{v_i} - \epsilon \cdot \log n.$$

For concreteness, suppose that $\alpha'_u \geq \alpha'_{v_i} + \epsilon \cdot \log n$ (the proof for $\beta'_u \leq \beta'_{v_i} - \epsilon \cdot \log n$ is analogous). Claim 18 implies that $\beta'_u \leq \beta'_{v_i} + \epsilon$. These two inequalities imply that

$$\beta'_u - \alpha'_u \leq \beta'_{v_i} + \epsilon - \alpha'_{v_i} - \epsilon \cdot \log n = (\beta'_{v_i} - \alpha'_{v_i}) - \epsilon \cdot (\log n - 1).$$

This implies Inequality (5), since v_{i+1} was chosen to minimize the value of $\beta'_{v_{i+1}} - \alpha'_{v_{i+1}}$.

Inequality (5) implies that, with overwhelming probability,

$$\begin{aligned}
|I_2| \cdot \epsilon \cdot (\log n - 1) &\leq \sum_{i \in I_2} \left(\beta'_{v_i} - \alpha'_{v_i}\right) - \left(\beta'_{v_{i+1}} - \alpha'_{v_{i+1}}\right) \\
&\leq \sum_{i \in I_2} \left(\beta'_{v_i} - \alpha'_{v_i}\right) - \left(\beta'_{v_{i+1}} - \alpha'_{v_{i+1}}\right) + \\
&\quad \sum_{i \in I_I} \left(\beta'_{v_i} - \alpha'_{v_i}\right) - \left(\beta'_{v_{i+1}} - \alpha'_{v_{i+1}}\right) + 1 \\
&= (\beta'_{\mathsf{root}} - \alpha'_{\mathsf{root}}) - \left(\beta'_{v_n} - \alpha'_{v_n}\right) + 1 \leq 2, \qquad (6)
\end{aligned}$$

where the first inequality follows by Equation (5) and summing over all $i \in I_2$, the second inequality follows by Claim 15, and the last inequality follows by our assumption that $\alpha'_{\mathsf{root}} > c$ and $\beta'_{\mathsf{root}} < 1 - c$ together with Claim 19.

Note that Claim 18 implies that for every $i \in [n]$, it holds that $X_i \geq -\epsilon$ and $Y_i \leq \epsilon$. This, together with Equation (6), implies that

$$\sum_{i \in I_2} X_i \geq -\epsilon \cdot |I_2| \geq -\frac{2}{\log n - 1} \qquad \text{and} \qquad \sum_{i \in I_2} Y_i \leq \epsilon \cdot |I_2| \leq \frac{2}{\log n - 1},$$

as desired. ◀

◀

References

1. Miklós Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.
2. Bar Alon and Eran Omri. Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In *Theory of Cryptography - 14th International Conference, TCC 2016-B*, pages 307–335, 2016.
3. Noga Alon and Moni Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM J. Comput.*, 22(2):403–417, 1993.
4. Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, third edition, 2008.
5. Amos Beimel, Iftach Haitner, Nikolaos Makriyannis, and Eran Omri. Tighter bounds on multi-party coin flipping, via augmented weak martingales and di erentially private sampling. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:168, 2017.
6. Amos Beimel, Eran Omri, and Ilan Orlov. Protocols for multiparty coin toss with a dishonest majority. *J. Cryptology*, 28(3):551–600, 2015.

7 Michael Ben-Or and Nathan Linial. Collective coin flipping. *Advances in Computing Research*, 5:91–115, 1989.

8 Ravi B. Boppana and Babu O. Narayanan. The biased coin problem. *SIAM J. Discrete Math.*, 9(1):29–36, 1996.

9 Niv Buchbinder, Iftach Haitner, Nissan Levi, and Eliad Tsfadia. Fair coin flipping: Tighter analysis and the many-party case. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2580–2600. SIAM, 2017.

10 Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 364–369. ACM, 1986.

11 Richard Cleve and Russell Impagliazzo. Martingales, collective coin flipping and discrete control processes (extended abstract), 1993. Unpublished manuscript.

12 Yevgeniy Dodis. Impossibility of black-box reduction from non-adaptively to adaptively secure coin-flipping. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(39), 2000.

13 Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. doi:10.1017/CBO9780511581274.

14 Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science, FOCS*, pages 142–153, 1999.

15 Shafi Goldwasser, Yael Tauman Kalai, and Sunoo Park. Adaptively secure coin-flipping, revisited. In *42nd International Colloquium on Automata, Languages and Programming„ ICALP*, pages 663–674, 2015.

16 Iftach Haitner and Eliad Tsfadia. An almost-optimally fair three-party coin-flipping protocol. *SIAM J. Comput.*, 46(2):479–542, 2017.

17 Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, FOCS*, pages 68–80, 1988.

18 Yael Tauman Kalai and Ilan Komargodski. Compressing communication in distributed protocols. In *Distributed Computing - 29th International Symposium, DISC*, pages 467–479, 2015.

19 David Lichtenstein, Nathan Linial, and Michael E. Saks. Some extremal problems arising form discrete control processes. *Combinatorica*, 9(3):269–287, 1989.

20 Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. *J. Cryptology*, 29(3):491–513, 2016.

21 Alexander Russell, Michael E. Saks, and David Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. *SIAM J. Comput.*, 31(6):1645–1662, 2002.

22 Michael E. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM J. Discrete Math.*, 2(2):240–244, 1989.