Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 71 (2020)

A Tight Lower Bound on Adaptively Secure Full-Information Coin Flip

Iftach Haitner [*†]　　Yonatan Karidi-Heller[†‡]

October 27, 2024

Abstract

In a distributed coin-flipping protocol, Blum [ACM Transactions on Computer Systems '83], the parties try to output a common (close to) uniform bit, even when some adversarially chosen parties try to bias the common output. In an adaptively secure full-information coin flip, Ben-Or and Linial [FOCS '85], the parties communicate over a broadcast channel, and a computationally unbounded adversary can choose which parties to corrupt *along* the protocol execution. Ben-Or and Linial proved that the n-party majority protocol is resilient to $O(\sqrt{n})$ corruptions (ignoring poly-logarithmic factors), and conjectured this is a tight upper bound for any n-party protocol (of any round complexity). Their conjecture was proved to be correct for *single-turn* (each party sends a single message) *single-bit* (a message is one bit) protocols Lichtenstein, Linial, and Saks [Combinatorica '89], *symmetric* protocols Goldwasser, Tauman Kalai, and Park [ICALP '15], and recently for (arbitrary message length) single-turn protocols Tauman Kalai, Komargodski, and Raz [DISC '18]. Yet, the question of many-turn protocols was left entirely open.

In this work, we close the above gap, proving that *no* n-party protocol (of any round complexity) is resilient to $\omega(\sqrt{n})$ (adaptive) corruptions.

[*] Stellar Development Foundation and The Blavatnik School of Computer Science, Tel Aviv University. Email: iftachh@gmail.com.

[†] Research supported by ERC starting grant 638121, and Israel Science Foundation grants 666/19 and 836/23.

[‡] The Blavatnik School of Computer Science, Tel Aviv University. Email: ykaridi@gmail.com.

ISSN 1433-8092

Contents

1 Introduction

In a distributed (also known as collective) coin-flipping protocol, Blum [7], the parties try to output a common (close to) uniform bit, even when some adversarially chosen parties try to bias the output. Coin-flipping protocols are fundamental primitives in cryptography and distributed computation, allowing distrustful parties to agree on a common random string (e.g., public randomness) to be used in their joint computation. More generally, almost any random process/protocol/algorithm encapsulates (some form of) a coin-flipping protocol. Consider a random process whose Boolean output is far from being fixed (e.g., has noticeable variance). Such a process can be thought of as a coin-flipping protocol: the common coin is the output, and the process's randomness corresponds to the parties' messages. Thus, lower bounds on coin-flipping protocols induce limitations on the stability of random processes (see Section 1.2.2 for a concrete example).

The focus of this work is *full-information* coin-flipping protocols, Ben-Or and Linial [4]. In this variant, the parties communicate solely over a single broadcast channel, and the Byzantine adversary[1] is assumed to be computationally *unbounded*. Two types of such adversaries are considered: A *static* adversary that chooses the parties it corrupts *before* the execution begins, and an *adaptive* adversary that can choose the parties it wishes to corrupt *during* the protocol execution (i.e., as a function of the messages seen so far). For static adversaries, full-information coin flips are well understood, and almost matching upper (protocols) and lower (attackers) bounds are known, see Section 1.2. Much less is understood about adaptive adversaries, which are the focus of this work, and significant gaps exist between the upper and lower bounds. Ben-Or and Linial [4] proved that the n-party majority protocol is resilient to $O(\sqrt{n})$ corruptions (ignoring poly-logarithmic factors in n), and conjectured that this is a tight upper bound for any n-party protocol (i.e., of any round complexity). The works of Lichtenstein, Linial, and Saks [19], Goldwasser, Tauman Kalai, and Park [14] made progress towards proving the conjecture for *single-turn* (each party sends a single message) protocols, a case that was eventually proved by Tauman Kalai, Komargodski, and Raz [27]. Yet, the question of many-turn protocols was left entirely open.

1.1 Our Results

We solve this intriguing question, showing that the output of any n-party protocol can be *fully biased* by an *adaptive* adversary corrupting $O(\sqrt{n})$ parties (ignoring poly-logarithmic factors).

Theorem 1.1 (Biasing full-information coin-flipping protocols, informal)**.** *For any n-party full-information coin-flipping protocol, there exists $b \in \{0,1\}$ and an (unbounded) adversary that, by adaptively corrupting $O(\sqrt{n})$ of the parties, forces the outcome of the protocol to b, except with probability $o(1)$.*

The above lower bound matches (up to poly-logarithmic factors) the upper bound achieved by the n-party majority protocol [4]. The bound extends to biased protocols, i.e., protocols with expected outcome (in an all-honest execution) different from $1/2$. We also remark that the one-side restriction (only possible to bias the protocol outcome to some $b \in \{0,1\}$) is inherent, as there exists, for instance, an n-party (single-turn) protocol that is resilient to $\Theta(n)$ corruptions trying to bias its outcome towards one.[2]

[1] Once it corrupts a party, it completely controls it and can send arbitrary messages on its behalf.

[2] Consider the n-party single-turn protocol in which each party broadcasts a $(1/n, 1 - 1/n)$-biased bit (i.e., equals

Although the one-sidedness is inherent in the adaptive case, we can overcome this by allowing the adversary to perform *strongly adaptive* corruptions, i.e., the adversary can decide whether to corrupt a party *after* seeing the message it is about to send.

Theorem 1.2 (Biasing full-information coin-flipping protocols via strongly adaptive attacks, informal)**.** *For any n-party full-information coin-flipping protocol and* any non-insignificant output *$b \in \{0,1\}$, there exists an (unbounded) adversary that by performing at most $O(\sqrt{n})$ strongly-adaptive corruptions, forces the outcome of the protocol to b, except with probability $o(1)$.*

1.2 Related Work

1.2.1 Full-Information Coin Flip

We recall the main known results for n-party full-information coin-flipping protocols.

Adaptive adversaries. In the following, we ignore poly-logarithmic factors in n.

Upper bounds (protocols). Ben-Or and Linial [4] proved that the majority protocol is resilient to $O(\sqrt{n})$ corruptions.

Lower bounds (attacks). Lichtenstein, Linial, and Saks [19] proved that no *single-bit* (messages are one bit) single-turn protocol is resilient to $\Omega(\sqrt{n})$ adaptive corruptions (hence, majority is optimal for such protocols). Dodis [9] proved that it is impossible to create a coin-flipping protocol resilient to $\Omega(\sqrt{n})$ adaptive corruptions by *sequentially repeating* another coin-flipping protocol, and then applying a deterministic function to the outcomes. Goldwasser, Tauman Kalai, and Park [14] proved that that no *symmetric* single-turn (many-bit) protocol is resilient to $\Omega(\sqrt{n})$ adaptive corruptions. Their result extends to *strongly adaptive* attacks (the attacker can decide to corrupt a party *after* seeing the message it is about to send) on single-turn protocols. Tauman Kalai, Komargodski, and Raz [27] fully answered the single-turn case by proving that no single-turn protocol is resilient to $\Omega(\sqrt{n})$ adaptive corruptions. Lastly, Etesami, Mahloujifar, and Mahmoody [11] presented an *efficient* strongly adaptive attack on protocols of certain properties (e.g., public coins).

Static adversaries. The case of static adversaries is well studied and understood.

Upper bounds (protocols). Ben-Or and Linial [4] presented a protocol that tolerates $O\big(n^{0.63}\big)$ corrupted parties (an improvement on the $O(\sqrt{n})$ corrupted parties it takes to bias the majority protocol). Ajtai and Linial [1] presented a protocol that tolerates $O(n/\log^2 n)$ corruptions. Saks [26] presented a protocol that tolerates $O(n/\log n)$ corruptions. The protocol of [26] was later improved by Alon and Naor [2] to tolerate a constant fraction of corrupted parties. Shortly afterwards, Boppana and Narayanan [8] presented an optimal protocol resilient to $(1/2 - \delta)n$ corruptions for any $\delta > 0$.

Lower bounds (attacks). Kahn, Kalai, and Linial [18] proved that no single-bit single-turn protocol can tolerate $\Omega(n/\log n)$ corruptions. Russell, Saks, and Zuckerman [25] proved that a protocol tolerating $\Omega(n)$ corruptions is either many-bit or has $\Omega(1/2 - o(1)) \cdot \log^*(n)$ rounds.

zero with probability $1/n$) and the protocol output is set to the AND of these bits. It is clear that the protocol expected outcome is $(1 - 1/n)^n \approx 1/e$ (can be made $1/2$ by slightly changing the distribution), and that even $n/2$ adaptive corruptions cannot change the protocol outcome to a value larger than $(1 - 1/n)^{n/2} \approx \sqrt{1/e}$.

The reader is referred to Dodis [10] for a more elaborated, somewhat outdated survey on full-information coin flip and friends.

1.2.2 Data-Poisoning Attacks

Consider a learning algorithm that tries to learn a hypothesis from a training set of samples from different sources. The random process corresponding to this learning task can be naturally viewed as a coin-flipping protocol. Moreover, as first noticed by Mahloujifar and Mahmoody [20], an attacker on the latter coin flip induces a so-called *data-poisoning attack*: increasing the probability of a desired property (i.e., poisoning the training data) by tampering with a small number of sources. For this application, however, the attacker would better be able to force a predetermined output (rather than forcing some output, as our attack achieves). Hence, the attack we apply on the coin-flipping protocol should be *bidirectional* (have the ability to (almost-) fully determine the coin, rather than biasing it to some arbitrary value). While this goal is unachievable in some models (see Footnote 2), it is achievable in some important ones (e.g., [12]).

Materializing their observation, Mahloujifar and Mahmoody [20] translated a two-directional *static* attack into a static data-poisoning attack on learning algorithms. Their attack was further improved in [22, 23]. Mahloujifar and Mahmoody [21] translated the two-directional *adaptive* attack of [27] on single-bit, single-turn coin-flipping protocols into an adaptive data-poisoning attack. Finally, Etesami et al. [12] facilitated their strongly adaptive attack on single-turn coin-flipping protocols (see Section 1.2) to obtain a strongly-adaptive data-poisoning attack.

The adaptive adversaries above cast attacks on *single-turn* coin-flipping protocols into data-poisoning attacks that tamper some samples. With the tools we present (see Theorem 1.1), it is now possible to discuss data-poisoning attacks that scale with the amount *sources*, rather than the amount of samples (which we may have orders of magnitude more of).

Open Questions

In this work, we show that the expected outcome of *any* n-party full-information coin-flipping protocol can be biased to either $o(1)$ or to $1 - o(1)$, using $O(\sqrt{n})$ corruptions. The above $o(1)$, however, stands for $1/\operatorname{loglog}(n)$, and it remains an intriguing question whether it can be pushed to $2^{-\operatorname{polylog}(n)}$ as can be achieved, for instance, when attacking the n-party majority protocol. Such attacks are known for *uniform* single-bit single-turn protocols (a secondary result of [27]) and for *strongly adaptive* attackers against single-turn protocols [11].

Paper Organization

A rather elaborate description of our attack on coin-flipping protocols is given in Section 2. Basic notations, definitions, and facts are given in Section 3. We also present there some useful manipulations of coin-flipping protocols. In Section 4, we show how to attack protocols of certain structure, that we refer to as *robust*, and in Section 5 we extend this attack to arbitrary protocols. Finally, in Section 6 we present a *bidirectional* strongly adaptive attack (a bidirectional attack is impossible in the standard adaptive model).

2 Our Technique

This section gives a rather elaborate description and analysis of our adaptive attack on full-information coin-flipping protocols. Let Π be an n-party, ℓ-round, full-information coin-flipping protocol. We prove that one can either bias the expected outcome of Π to less than $\varepsilon := 1/\operatorname{loglog}(n)$, or to more than $1-\varepsilon$.

Similarly to previous adaptive attacks on full-information coin-flipping protocols, our attack makes use of the "jumps" in the protocol's expected outcome; assume without loss of generality (see Section 3.4 for justification) that in each round only a *single* party sends a message, and let $\mathrm{Msg} = (\mathrm{Msg}_1, \ldots, \mathrm{Msg}_\ell)$ denote the protocol transcript (i.e., parties' messages) in a random all-honest execution of Π. For $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg})$, let $\Pi(\mathrm{msg})$ denote the final outcome of the execution described by msg. For $\mathrm{msg}_{\leq i} \in \mathrm{Supp}(\mathrm{Msg}_{\leq i} := (\mathrm{Msg}_1, \ldots, \mathrm{Msg}_i))$ let $\Pi(\mathrm{msg}_{\leq i}) := \mathbb{E}\left[\Pi(\mathrm{Msg}) \mid \mathrm{Msg}_{\leq i} = \mathrm{msg}_{\leq i}\right]$ be the expected outcome given a partial transcript, and let

$$\mathrm{jump}(\mathrm{msg}_{\leq i}) := \Pi(\mathrm{msg}_{\leq i}) - \Pi(\mathrm{msg}_{< i})$$

be the "jump" in the expected outcome induced by the i^{th} message. Accordingly, we refer to $\mathrm{jump}(\mathrm{Msg}_{\leq i})$ as the i^{th} jump in the protocol execution. Our attack manipulates these jumps in a very different manner than what previous attacks did. First, the decision whether to corrupt a given message is based on the (conditional) *variance* of the jumps (L_2 norm), a more subtle measure than the *maximal* possible change (L_∞ norm) considered by previous attacks. Second, even when the attacker decides that the next message is useful for biasing the protocol's outcome, it only *gently* alters the message: It corrupts the party about to send the message only with a certain probability, and even when corrupting, only moderately changes the message distribution. Being gentle allows the attack to bypass the main obstacles in attacking many-turn protocols. The gentleness also makes analyzing the attack's performance easier; the transcript of the gently attacked execution is not "too different" from the all-honest (un-attacked) execution of the protocol. Consequently, the analysis requires only a good understanding of the all-honest execution, and not of the typically very complicated execution the attack induces.[3] Details below.

We start by describing an attack on *robust* protocols: for some $b \in \{0,1\}$, the protocol has *no* ${}^{1}/_{\sqrt{n}}$ jumps towards b,[4] and for concreteness, we assume $b = 0$. That is, for every i (w.p. one)

$$\Pi(\mathrm{Msg}_{\leq i}) \geq \Pi(\mathrm{Msg}_{< i}) - {}^{\varepsilon^2}/_{\sqrt{n}} \tag{1}$$

We first consider single-turn robust protocols and then generalize to many-turn (robust) protocols. The extension to arbitrary (non-robust) protocols is described in Section 2.3.

2.1 Attacking Robust Single-Turn Coin Flip

Our attack gently biases the expected outcome of the protocol by carefully manipulating the message distributions of the corrupt parties. This manipulation (to be applied to the party's next message distribution) is defined as follows.

[3] Gentle attacks, in the general sense that the attacker does not try to maximize the effect of the attack in each round but instead keeps the attacked execution similar to the all-honest one, were found helpful in many other settings. A partial list includes attacking different types of coin-flipping protocols [24, 16, 5, 3], and proving parallel repetition of computationally sound proofs [17, 15, 6].

[4] An almost accurate example of a (bidirectional) robust protocol is the single-bit, single-turn, n-party *majority* protocol: in each round, a single party broadcasts an unbiased coin, and the protocol's final output is set to the majority of the coins. It is well known that the absolute value of most jumps is (typically) order of ${}^{1}/_{\sqrt{n}}$.

Definition 2.1 (Biased distribution). *For a distribution P, $\alpha \geq 0$ and mapping $f \colon \operatorname{Supp}(P) \mapsto [-1/\alpha, \infty)$ with $\mathbb{E}\big[f(P)\big] = 0$, let $\operatorname{Biased}^f_\alpha\big(P\big)$ be the distribution defined by*

$$\mathbb{P}\big[\operatorname{Biased}^f_\alpha\big(P\big) = e\big] := \mathbb{P}\big[P = e\big] \cdot (1 + \alpha \cdot f(e))$$

That is, the distribution P is "nudged" towards larger values of f, i.e., increasing the probability of positive elements (causing $\mathbb{E}\big[f\big(\operatorname{Biased}^f_\alpha\big(P\big)\big)\big] \geq \mathbb{E}\big[f(P)\big]$), and the larger α is the larger the bias is. Equipped with this definition, our attacker on a (n-party, single-turn, n-round, robust) coin-flipping protocol Π is defined in as follows:

Algorithm 2.2 (Single-turn attacker, informal). *For $i = 1$ to n, do the following* before *the i^{th} message is sent:*

1. *Let $\operatorname{msg}_{<i}$ be the previously sent messages. Let $Q_i := \operatorname{Msg}_i|_{\operatorname{Msg}_{<i}=\operatorname{msg}_{<i}}$, $\operatorname{jump}_i := \operatorname{jump}(\operatorname{msg}_{<i}, \cdot)$, let $v_i := \operatorname{Var}[\operatorname{jump}_i(Q_i)]$.*
2. *If $v_i \geq \varepsilon^4/n$, corrupt the i^{th} party with probability $1/\varepsilon^4 \cdot \sqrt{v_i}$. If corrupted, instruct it to send the next message according to $\operatorname{Biased}^{\operatorname{jump}_i}_{1/\sqrt{v_i}}\big(Q_i\big)$.*

That is, a party is corrupted with probability proportional to the (conditional) standard deviation it induces on the expected outcome of Π (i.e., messages inducing larger variance on the protocol outcome are more likely to be corrupted). If corrupted, the message distribution is modified so that the change it induces on the expected outcome of Π is biased towards one, where the bias is proportional to the inverse of the standard deviation (i.e., messages with smaller variance are leveraged more aggressively, to "compensate" for the fact they have small variance).

Example 2.3 (Attacking single-turn majority). *If Π is the single-turn, n-party, single-bit, majority protocol, then (typically) each v_i is of (absolute) order $1/n$. Thus, in expectation, the above attack corrupts $1/\varepsilon^3 \cdot \sqrt{n}$ parties. If corrupted, the party's bit message is set to 1 with probability $\approx 1/2 \cdot (1 + \sqrt{n} \cdot 1/\sqrt{n}) = 1$.*

In the following, we argue that the attacker indeed biases the expected outcome of Π to $1 - \varepsilon$ and that the expected number of corruptions is $\widetilde{O}(\sqrt{n})$. Therefore, a Markov bound yields the existence of the required attacker. We prove the success of our attack by showing that the attacked protocol has too little "liveliness" to resist the attacker's bias. Consequently, the outcome is (with high probability) the value the attacker biases towards. Our notion of liveliness is the *conditional variance* of some underlying distribution induced by the attack.

Let $\operatorname{Msg}^{\mathsf{A}} = (\operatorname{Msg}^{\mathsf{A}}_1, \ldots, \operatorname{Msg}^{\mathsf{A}}_n)$ be the message distribution induced by the above attack. For $i \in [n]$, let Q_i be the value of Q_i in the attacked execution, determined by $\operatorname{Msg}^{\mathsf{A}}_{i-1}$, and sample $Y_i \leftarrow \operatorname{jump}(\operatorname{Msg}^{\mathsf{A}}_{<i}, Q_i)$. Q_i correspond to an honest message distribution, therefore, $\mathbb{E}\big[Y_i \mid \operatorname{Msg}^{\mathsf{A}}_{<i}\big] = 0$. Hence, the sequence $Y = (Y_1, \ldots, Y_n)$ is a *martingale difference sequence* with respect to $(\operatorname{Msg}^{\mathsf{A}}_i, Y_i)_{i=1}^n$.[5] This martingale can be seen as an honest execution based on a corrupted history.

We show that $Y_1, \ldots, Y_n$ have little "liveliness": their overall impact on the outcome is small. It follows protocol's outcome is determined solely by the adversary's manipulations. Those manipulations push the outcome towards 1, so we conclude that the protocol outcome must be 1. The

[5] The Y_i's are sampled such that they are independent of each other even when conditioned on $\operatorname{Msg}^{\mathsf{A}}$.

core of our analysis lies in the following lemma, where we argue about the liveliness of $Y_1, \ldots, Y_n$ (defined as the sum of conditional variances of $Y_1, \ldots, Y_n$).

Lemma 2.4. $\mathbb{E}\big[\sum_{i=1}^{n} \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]\big] \leq \varepsilon^3$.

The proof of Lemma 2.4 is sketched below, but we first use it to analyze the attack's quality. Think of $\mathrm{Var}[\sum_{i=1}^{n} Y_i]$ as the protocol "liveliness" described before. Y is a martingale difference sequence with respect to $\mathrm{Msg}^{\mathsf{A}}_i$, i.e., $\mathbb{E}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] = 0$, and so it is easy to verify that

$$\begin{aligned}\mathrm{Var}\big[\sum_{i=1}^{n} Y_i\big] &= \sum_{i=1}^{n} \mathrm{Var}\big[Y_i\big] = \sum_{i=1}^{n} \big(\mathbb{E}\big[\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] + \mathrm{Var}\big[\mathbb{E}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big]\big) \\ &= \sum_{i=1}^{n} \mathbb{E}\big[\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] = \mathbb{E}\big[\sum_{i=1}^{n} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \leq \epsilon^3.\end{aligned}$$

The first equality follows by Fact 3.7 (martingale increments are orthogonal), the second by the law of total variance (Fact 3.9), and the last inequality by Lemma 2.4. Thus, by Chebyshev's inequality

$$\mathbb{P}\big[\big|\sum_{i=1}^{n} Y_i\big| \geq \varepsilon\big] \leq \varepsilon \tag{2}$$

Consider the *sub*-martingale $S = (S_0, \ldots, S_n)$ with respect to $\big\{\mathrm{Msg}^{\mathsf{A}}_i\big\}_{i=1}^{n}$ defined by $S_i := \Pi(\mathrm{Msg}^{\mathsf{A}}_{\leq i})$, i.e., the expected honest outcome given $\mathrm{Msg}^{\mathsf{A}}_{\leq i}$. By definition, $S_0 = \mathbb{E}\big[\Pi(\mathrm{Msg})\big] = {}^1\!/_2$ and $S_n = \Pi(\mathrm{Msg}^{\mathsf{A}}) \in \{0, 1\}$. In addition, the attack only *increases* the conditional expectation $\mathbb{E}\big[S_{i+1} - S_i \mid \mathrm{Msg}^{\mathsf{A}}_{\leq i}\big]$ and originally the protocol jumps have (conditional) expectation of zero, hence, it holds that $\mathbb{E}\big[S_{i+1} - S_i \mid \mathrm{Msg}^{\mathsf{A}}_{\leq i}\big] \geq 0$. So indeed, S constitutes a *sub*-martingale sequence. Since $(S_{i+1} - S_i)$ is a "biased towards one" variant of Y_i, there exists a (rather) straightforward coupling between S and Y for which

$$S_i - S_{i-1} \geq Y_i.$$

By definition $S_0 = 1/2$, and thus $\mathbb{P}\big[S_n \leq 0\big] = \mathbb{P}\big[\sum_{i=1}^{n} S_i - S_{i-1} \leq -{}^1\!/_2\big]$. By the properties of the aforementioned coupling, $\mathbb{P}\big[\sum_{i=1}^{n}(S_i - S_{i-1}) \leq -{}^1\!/_2\big] \leq \mathbb{P}\big[\sum_{i=1}^{n} Y_i \leq -{}^1\!/_2\big]$, and by Equation (2), $\mathbb{P}\big[\sum_{i=1}^{n}(S_i - S_{i-1}) \leq -{}^1\!/_2\big] \leq \varepsilon$. Finally, $S_n \in \{0, 1\}$ therefore, $\mathbb{P}\big[S_n = 1\big] = 1 - \mathbb{P}\big[S_n \leq 0\big] \geq 1 - \varepsilon$. Namely, the output of the attacked protocol is 1 with a probability of at least $1 - \varepsilon$.

We conclude the attack analysis by bounding the expected number of corruptions. By construction, the probability the attacker corrupts the i^{th} party is at most

$$\begin{aligned}{}^1\!/_{\varepsilon^4} \cdot \sqrt{\mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]} &= {}^1\!/_{\varepsilon^4} \cdot {}^{\mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]}\!/_{\sqrt{\mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]}} \\ &\leq {}^1\!/_{\varepsilon^4} \cdot (\mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}] \cdot \sqrt{{}^n\!/_{\varepsilon^4}}) = {}^{\sqrt{n}}\!/_{\varepsilon^6} \cdot \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}].\end{aligned}$$

Overall, the total amount of corruptions is at most

$$\mathbb{E}\big[\sum_{i=1}^{n} {}^{\sqrt{n}}\!/_{\varepsilon^6} \cdot \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]\big] = {}^{\sqrt{n}}\!/_{\varepsilon^6} \cdot \mathbb{E}\big[\sum_{i=1}^{n} \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]\big].$$

Consequently, by Lemma 2.4, the expected number of corruptions is at most ${}^{\sqrt{n}}\!/_{\varepsilon^6} \cdot \varepsilon^3 = {}^{\sqrt{n}}\!/_{\varepsilon^3} = \widetilde{O}(\sqrt{n})$.

Proving Lemma 2.4. By definition of Biased, for any $p \in [0,1]$ it holds that

$$\Big(p \cdot \mathrm{Biased}^{f}_{\alpha}\big(P\big) + (1-p) \cdot P\Big) \equiv \mathrm{Biased}^{f}_{p\cdot\alpha}\big(P\big) \tag{3}$$

Also note that for any distribution P, constant $\alpha \geq 0$ and function $f\colon \mathrm{Supp}(P) \mapsto [-{}^{1}/_{\alpha}, \infty)$ with $\mathbb{E}\big[f(P)\big] = 0$, it holds that

$$\begin{aligned}\mathbb{E}\big[f\big(\mathrm{Biased}^{f}_{\alpha}\big(P\big)\big)\big] &= \sum_{x \in \mathrm{Supp}(P)} f(x) \cdot \mathbb{P}\big[\mathrm{Biased}^{f}_{\alpha}\big(P\big) = x\big] \\ &= \sum_{x \in \mathrm{Supp}(P)} f(x) \cdot \mathbb{P}\big[P = x\big] \cdot (1 + \alpha \cdot f(P)) \\ &= \mathbb{E}\big[f(P) \cdot (1 + \alpha \cdot f(P))\big] = \mathbb{E}\big[f\big] + \alpha \cdot \mathbb{E}\big[f^2(P)\big] = 0 + \alpha \cdot \mathrm{Var}\big[f(P)\big].\end{aligned} \tag{4}$$

Let V_i be the value of the variable v_i in the execution of the attack (determined by $\mathrm{Msg}^{\mathsf{A}}_{<i}$), and let C_i be the event $\{V_i \geq {}^{\varepsilon^4}/_{n}\}$, i.e., whether the adversary *even considers* to corrupt the current party (see Step 2 of the attacker). Using the above notation, proving Lemma 2.4 translates to proving that

$$\mathbb{E}\big[\sum_{i=1}^{n} V_i\big] \leq \varepsilon^3 \tag{5}$$

Let $\mathrm{msg}_{<i} \in \mathrm{Supp}(\mathrm{Msg}^{\mathsf{A}}_{<i})$ be a partial transcript. Denote $\nu_i = \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}]$ (i.e., matching v_i or V_i). For transcripts satisfying $\nu_i \geq {}^{\varepsilon^4}/_{n}$ (i.e., the event C_i holds), applying Equations (3) and (4) with respect to

$$P := \mathrm{Msg}_i\big|_{\mathrm{Msg}_{<i} = \mathrm{msg}_{<i}},\ p := {}^{1}/_{\varepsilon^4} \cdot \sqrt{\nu_i},\ \alpha := {}^{1}/_{\sqrt{\nu_i}} \ \text{ and }\ \mathrm{jump}_i := \mathrm{jump}(\mathrm{msg}_{<i}, \cdot),$$

yields

$$\begin{aligned}\mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}\big] &= \mathbb{E}\big[\mathrm{jump}_i\big(\mathrm{Biased}^{\mathrm{jump}_i}_{1/\varepsilon^4}\big(\mathrm{Msg}_i\big|_{\mathrm{Msg}_{<i} = \mathrm{msg}_{<i}}\big)\big)\big] \\ &= {}^{1}/_{\varepsilon^4} \cdot \mathrm{Var}\big[\mathrm{jump}(\mathrm{Msg}_{\leq i}) \mid \mathrm{Msg}_{<i} = \mathrm{msg}_{<i}\big] \\ &= {}^{1}/_{\varepsilon^4} \cdot \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}\big].\end{aligned} \tag{6}$$

Hence,

$$\mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] = \mathbb{1}_{C_i} \cdot \mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] = \mathbb{1}_{C_i} \cdot {}^{1}/_{\varepsilon^4} \cdot V_i \tag{7}$$

The first equality holds by construction (if C_i doesn't hold, the party is not corrupted and thus expectation is zero). The second equality by Equation (6). Therefore,

$$\mathbb{E}\big[S_n - S_0\big] = \mathbb{E}\big[\sum_{i=1}^{n}(S_i - S_{i-1})\big] = \mathbb{E}\big[\sum_{i=1}^{n} \mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \leq {}^{1}/_{\varepsilon^4} \cdot \mathbb{E}\big[\sum_{i=1}^{n} \mathbb{1}_{C_i} \cdot V_i\big] \tag{8}$$

The second equality follows by the law of total expectation (Fact 3.8). Since S_0 and S_n take values in $[0,1]$, it follows that

$$\mathbb{E}\big[\sum_{i=1}^{n} \mathbb{1}_{C_i} \cdot V_i\big] \leq \varepsilon^4 \tag{9}$$

In addition, by definition of C_i, for any i it holds that

$$\mathbb{1}_{\neg C_i} \cdot V_i \leq {}^{\varepsilon^4}\!/_{n} \tag{10}$$

Combining Equations (9) and (10), we deduce that $\mathbb{E}\left[\sum_{i=1}^{n} V_i\right] \leq \varepsilon^4 + {}^{\varepsilon^4}\!/_{n} \cdot n < \varepsilon^3$.

2.2 Attacking Robust Many-Turn Coin Flip

When attacking many-turn coin-flipping protocols, one encounters two additional problems:

Identify influential parties. In many-turn protocols, each message might have *very little influence on the protocol's outcome.* So, it is unclear for an online attacker to decide which parties to corrupt; e.g., a party that sends insignificant messages at the beginning of the protocol might turn out to be influential in the future, or on the flip side a party that had a significant influence on the protocol will not necessarily have significant influence in the rest of the execution.

Preserve corrupted parties' influence. Even if we have successfully identified potentially influential parties in the protocol, we must not alter their behavior in a way that makes it obvious they are corrupted. If the corrupted parties' messages differ vastly from the honest execution, it might no longer be the case that those parties stand a chance at significantly influencing the protocol's outcome.

Let us exemplify the above obstacles using the following two examples, respectively.

Example 2.5 (Shrinking majority). *Consider a "shrinking" n-party n^2-round majority protocol: a majority protocol consisting of n super-rounds in which every player sends a single bit and in addition, a random (determined by the rounds' coins) party is cast out (meaning from this super-round on, its bits are ignored). In such a case, the attacker must decide whether to corrupt a party without being certain that it will "survive" for many rounds.*

Example 2.6 (Punishment mechanism). *Consider the n-party n^2-round majority protocol, i.e., each party sends n bits, that is equipped with the following "punishment" mechanism: once a party's coins are "too suspicious", say contain a 1-run of length $\log^2 n$, its coins are ignored from this point on. So, once corrupting a party, the attack should not attempt to bluntly maximize the effect of the messages it sends.*

We tackle both problems (respectively) following two general ideas.

Corrupt parties at random based on their past influence. Our attacker decides at random whether to corrupt a party, based on their past influence. That is, the corruption process can be viewed as a lottery: a party starts with a single ticket (i.e., chance to become corrupted), and every time it contributes a certain amount of influence on the protocol it gets another ticket (i.e., another chance to become corrupted).[6] While the above approach does not identify an influential party before it starts affecting the outcome, it does so before the party significantly affects the outcome.

[6] In the actual attack, we formalize this approach by partitioning the parties into *pseudo-parties* with bounded influence and corrupt each at random *independently* of each other.

Gently modify corrupted parties. Once deciding to corrupt a party, our attacker only *subtly* alters its messages, like in the single-turn case where we use the Biased transformation to subtly alter the messages of the corrupted parties; such a gentle attack assures that we do not "drift" too much from a "typical" execution, and in particular, influential parties remain influential even in the attacked protocol.

Highly influential messages. In addition to that above, we bias highly influential messages exactly like the single-turn case, i.e., the probability we corrupt the matching party is only related to the influence of the message—without taking into consideration the number of parties.

2.2.1 The Attack

The above intuition takes form as the following attacker (against a n-party, ℓ-round, robust protocol Π).

Algorithm 2.7 (Many-turn protocols attacker, informal)**.**

For $i = 1$ to ℓ, do the following before *the i^{th} message is sent:*

1. *Let* $\mathrm{msg}_{<i}$ *be the previously sent messages, let* $Q_i := \mathrm{Msg}_{\leq i}|_{\mathrm{Msg}_{<i}=\mathrm{msg}_{<i}}$*, let* $\mathrm{jump}_i := \mathrm{jump}(\mathrm{msg}_{<i}, \cdot)$*, let* $v_i := \mathrm{Var}[\mathrm{jump}_i(Q_i)]$.
2. *If $v_i \geq \varepsilon^4/n$, corrupt the party sending the i^{th} message with probability $1/\varepsilon^4 \cdot \sqrt{v_i}$. If corrupted, instruct it to send its next message according to* $\mathrm{Biased}^{\mathrm{jump}_i}_{1/\sqrt{v_i}}(Q_i)$*. (i.e., like in the singe-turn case.)*

   Else, if the i^{th} message is the first message to be sent by the party, or the overall contribution of the messages it sent since the last time it was considered for corruption exceeded ε^4/n,[7] *corrupt this party with probability $1/\varepsilon^4 \cdot 1/\sqrt{n}$ (if it was decided not to corrupt the party, we consider this party honest from now on).*

   If corrupted (at the last decision point), instruct it to send its next message according to $\mathrm{Biased}^{\mathrm{jump}_i}_{\sqrt{n}}(Q_i)$*.*

That is, a large-jump party is treated like in the single-turn case. In contrast, a small-jumps party is corrupted with (fixed) probability proportional to $1/\sqrt{n}$ (when corrupted, *all* messages of the party are modified).

Example 2.8 (Attacking many-turn majority)**.** *Consider the n-party, n^2-round, single-bit majority protocol (in which each party sends n bits). Typically, the change induced by any given message is of order $1/n$. Consequently, each v_i is of order $1/n^2$, and each party will be independently corrupted with probability $1/\varepsilon^3 \cdot 1/\sqrt{n}$. Thus, in expectation, the above attack corrupts $1/\varepsilon^3 \cdot \sqrt{n}$ parties. If corrupt, each of the single-bit messages the party sends is 1 with probability $\approx 1/2 \cdot (1 + \sqrt{n} \cdot 1/n) = 1/2 + 1/2\sqrt{n}$*

[7] In the main body of the paper, we transform (in the attacker's head) arbitrary protocols into *normal* protocols in which the overall influence of a party's small-jump messages is limited. Thus, each party needs to be tested for corruption only once.

Analysis. The analysis of the above attack is similar to the single-turn case. Let $\mathrm{Msg}^{\mathsf{A}} = (\mathrm{Msg}^{\mathsf{A}}_1, \ldots, \mathrm{Msg}^{\mathsf{A}}_\ell)$, $S = (S_0, \ldots, S_\ell)$ and $Y = (Y_1, \ldots, Y_\ell)$ be as in the single-turn case. Similarly to the single-turn case, the core of the proof lies in the following lemma.

Lemma 2.9. $\mathbb{E}\big[\sum_{i=1}^{\ell} \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}]\big] = O(\varepsilon^4)$.

The challenge in proving Lemma 2.9 is that, unlike the single-turn proof, it might be that the following does not hold:

$$\mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}\big] \geq {}^{1}\!/\!_{\varepsilon^4} \cdot \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}].$$

Indeed, let V_i be the value of the variables v_i in the execution of the attack described by $\mathrm{Msg}^{\mathsf{A}}$. Assume that conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}$, it holds that $V_i < {}^{\varepsilon^4}\!/\!_n$ and that a party P is about to send the i^{th} message. Unlike the single-turn case, the conditional probability that P is corrupted is no longer guaranteed to match the (non-conditional) probability that P is corrupted: the previous messages sent by P in $\mathrm{msg}_{<i}$ might *leak* whether P is corrupted or not. If the latter happens, then (by the same argument we used for proving the lemma in the single-turn case) it might be that $\mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}\big] < {}^{1}\!/\!_{\varepsilon^4} \cdot \mathrm{Var}[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}]$. Fortunately, since we only slightly modify each small-jump message and assume no party's small messages are too influential (which holds for normal protocols), a KL-divergence argument yields that on average for such messages it holds that $\mathbb{E}\big[S_i - S_{i-1} \mid \mathrm{Msg}^{\mathsf{A}}_{i-1} = \mathrm{msg}_{<i}\big] = \Omega({}^{1}\!/\!_{\varepsilon^4} \cdot \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{i-1} = \mathrm{msg}_{<i}\big])$, which suffices for the proof of the lemma to go through.

2.3 Attacking Non-Robust Coin Flip

The high-level idea of attacking a non-robust protocol (a protocol that has large jumps in both directions) is to attempt biasing the protocol towards zero in such a way that if this bias fails, the protocol will be robust for $b = 0$ (no large jumps downward)—prime for applying the attack on robust protocols. More formally, assume that with probability at least ${}^{1}\!/\!_{\log n}$, the protocol Π has a large negative jump, i.e., less than $-{}^{1}\!/\!\sqrt{n}$, and consider the following "one-shot" attacker on Π:[8]

Algorithm 2.10 (Negative jumps attacker)**.** *For $i = 1$ to n, do the following* before *the i^{th} message is sent:*

1. *Let $\mathrm{msg}_{<i}$ be the previously sent messages.*
2. *If there exists $m_i^- \in \mathrm{Supp}(\mathrm{Msg}_i \,|_{\mathrm{Msg}_{<i} = \mathrm{msg}_{<i}})$ such that $\Pi(\mathrm{msg}_{<i}, m_i^-) < \Pi(\mathrm{msg}_{<i}) - {}^{1}\!/\!\sqrt{n}$, and no party was corrupted yet, corrupt the party sending the i^{th} message and instruct it to send m_i^-.*

It is clear that above adversary biases the outcome of Π toward zero by at least ${}^{1}\!/\!_{\sqrt{n} \cdot \log(n)}$. Let Π_1 be the *protocol* induced by the above (deterministic) attack: all parties emulate the attacker in their head, and when it decides to (deterministically) corrupt a party, the corrupted party follows its (deterministic) instructions. If the protocol Π_1 has a large negative jump with probability larger than ${}^{1}\!/\!_{\log n}$, apply the above attack on Π_1 resulting in the protocol Π_2, and so on. Let $t \leq \sqrt{n} \cdot \log(n)$

[8] Assuming the next-message function of Π is efficiently samplable, e.g., Π is public-coin, the following attacker is the only reason for the inefficiency of our attack.

denote the number of times we applied the attack in this manner. If the expected outcome of Π_t is at most ε, then we are done: the implied t-adaptive adversary makes Π output 0 with probability $1-\varepsilon$. Otherwise, Π_t has the following property:

$$\mathbb{P}\left[\exists j \in [n] \colon \Pi_t(\widetilde{\mathrm{Msg}}_{\le j}) < \Pi_t(\widetilde{\mathrm{Msg}}_{< j}) - 1/\sqrt{n}\right] \le 1/\log(n) \tag{11}$$

letting $\widetilde{\mathrm{Msg}}$ be the messages of a random execution of Π_t. If the above happens, we apply the attack on robust protocols (Algorithm 2.7) on Π_t, instructing the adversary to halt if it encounters a large negative jump.

With careful analysis (and slight modifications to the attacker), one can show that, due to the *gentleness* of the attack, the property of encountering large negative jumps only with negligible probability is preserved. Hence, the attack carries as if there are no large negative jumps, meaning we have successfully biased the expected output of Π_t to $1-O(\varepsilon)$. Composing the attack that transforms Π into Π_t with the attack on robust protocols (on Π_t) yields the required attack on Π.[9]

3 Preliminaries

3.1 Notations

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values and functions. All logarithms considered here are base 2. For $n \in \mathbb{N}$, let $[n] := \{1,\ldots,n\}$. Given a Boolean statement S (e.g., $X \ge 5$), let $\mathbb{1}_S$ be the indicator function that outputs 1 if S is a true statement and 0 otherwise. For a distribution X, let $x \leftarrow X$ denote that x was sampled according to X.

3.2 Distributions and Random Variables

The support of a distribution P over a discrete set $\mathcal{X}$, denoted $\mathrm{Supp}(P)$, is defined by $\mathrm{Supp}(P) := \{x \in \mathcal{X} : P(x) > 0\}$. For random variables X and Y, let the random variable $\mathrm{Supp}(X \mid Y)$ denote the conditional support of X given Y. In addition, we define the random variables $\mathbb{E}\left[X \mid Y\right]$ and $\mathrm{Var}\left[X \mid Y\right]$ as (deterministic) functions of Y, by $\mathbb{E}\left[X \mid Y\right](y) := \mathbb{E}\left[X \mid Y = y\right]$ and $\mathrm{Var}\left[X \mid Y\right](y) := \mathrm{Var}\left[X \mid Y = y\right]$, respectively.

The statistical distance (also known as variation distance) of two distributions P and Q over a discrete domain $\mathcal{X}$ is defined by $\mathsf{SD}(P,Q) := \max_{\mathrm{S}\subseteq\mathcal{X}}|P(\mathrm{S}) - Q(\mathrm{S})| = \frac{1}{2}\sum_{x\in\mathrm{S}}|P(x) - Q(x)|$. Statistical distance enjoys a data-processing inequality.

Fact 3.1 (Data-processing inequality for statistical distance)**.** *For distributions P and Q and function f over a discrete domain $\mathcal{X}$, it holds that $\mathsf{SD}(f(P), f(Q)) \le \mathsf{SD}(P,Q)$.*

The KL-divergence (also known as Kullback-Leibler divergence and relative entropy) between two distributions P and Q over a discrete domain $\mathcal{X}$, is defined by

$$D_{\mathrm{KL}}\left(P \,\|\, Q\right) := \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)} = \mathbb{E}_{x\leftarrow P}\log\frac{P(x)}{Q(x)},$$

[9] In Section 2.1 we proved the quality of our attack for protocols with $\mathbb{E}\left[\Pi\right] = 1/2$. Still, the proof can be easily adapted to the caes of $\mathbb{E}\left[\Pi\right] \ge 1/\mathrm{polylog}(n)$.

where $0 \cdot \log \frac{0}{0} = 0$, and $D_{\mathrm{KL}}(P \,\|\, Q) := \infty$ if there exists $x \in \mathcal{X}$ such that $P(x) > 0$ but $Q(x) = 0$. KL-divergence is convex in the following sense:

Fact 3.2 (Convexity of KL-divergence)**.** *For finite distributions P_1, P_2, Q_1, Q_2, and $\lambda \in [0,1]$ it holds that $D_{\mathrm{KL}}(\lambda \cdot P_1 + (1-\lambda) \cdot P_2 \,\|\, \lambda Q_1 + (1-\lambda) \cdot Q_2) \leq \lambda \cdot D_{\mathrm{KL}}(P_1 \,\|\, Q_1) + (1-\lambda) \cdot D_{\mathrm{KL}}(P_2 \,\|\, Q_2)$.*

In addition, KL-divergence enjoys a chain rule.

Fact 3.3 (KL-divergence chain rule)**.** *For distributions $P(X,Y)$ and $Q(X,Y)$ for a pair of random variables X and Y, it holds that $D_{\mathrm{KL}}(P(X,Y) \,\|\, Q(X,Y)) = D_{\mathrm{KL}}(P(X) \,\|\, Q(X)) + \mathop{\mathbb{E}}_{x \leftarrow P(X)}\left[D_{\mathrm{KL}}(P(Y \mid X = x) \,\|\, Q(Y \mid X = x))\right]$.*

The following fact (see Fedotov et al. [13]) relates small KL-divergence to small statistical distance:

Fact 3.4 (Pinsker bound)**.** *For discrete distributions P and Q it holds that $\mathsf{SD}(P,Q) \leq \sqrt{\frac{1}{2} \cdot D_{\mathrm{KL}}(P \,\|\, Q)}$.*

3.3 Martingales

Martingales play an important role in our analysis.

Definition 3.5 (Martingales)**.** *A sequence of random variables $M = (M_1, \ldots, M_n)$ is a* martingale *with respect to a sequence of random variables $X_1, \ldots, X_n$, if $\mathbb{E}[M_{k+1} \mid X_{\leq k}] = M_k$ and M_k is determined by $X_{\leq k}$ (for every $k \in [n]$). The sequence M is a martingale if it is a martingale with respect to itself. The* increments *(also known as differences) sequence of M are the random variables $\{M_{k+1} - M_k\}_{k=1}^{n-1}$.*

In particular, we will be interested in the so-called Doob martingales.

Definition 3.6 (Doob martingales)**.** *The* Doob martingale *of the random variables $X = (X_1, \ldots, X_n)$ induced by the function $f \colon \mathrm{Supp}(X) \mapsto \mathbb{R}$, is the sequence $M_1, \ldots, M_n$ defined by $M_k := \mathbb{E}[f(X_1, \ldots, X_n) \mid X_{\leq k}]$.*

The proof of the following known facts is immediate.

Fact 3.7 (Martingale increments are orthogonal)**.** *Let $X_1, \ldots, X_n$ be a sequence of random variables. If there exist random variables $Z_1, \ldots, Z_n$ such that $\mathbb{E}[X_k \mid Z_{<k}] = 0$ and X_k is determined by $Z_{\leq k}$ (i.e., $\sum X_i$ is a martingale with respect to Z_k), then $\mathrm{Var}[\sum_{i=1}^n X_i] = \sum_{i=1}^n \mathrm{Var}[X_i]$.*

Fact 3.8 (Law of total expectation)**.** *For two random variables Y, X it holds that $\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y \mid X]]$.*

Fact 3.9 (Law of total variance)**.** *For two random variables Y, X it holds that $\mathrm{Var}[Y] = \mathbb{E}[\mathrm{Var}[Y \mid X]] + \mathrm{Var}[\mathbb{E}[Y \mid X]]$.*

Sub-martingales. We also use the related notion of sub-martingales.

Definition 3.10 (Sub-martingales)**.** *A sequence of random variables $S = (S_1, \ldots, S_n)$ is a* sub-martingale *with respect to a sequence of random variables $X_1, \ldots, X_n$, if $\mathbb{E}\big[S_{k+1} \,|\, X_{\leq k}\big] \geq S_k$ and S_k is determined by $X_{\leq k}$ (for every $k \in [n]$). The sequence S is a* sub-martingale *if it is a sub-martingale with respect to itself.*

In particular, we make use of the following known inequality.

Fact 3.11 (Doob's maximal inequality)**.** *Let $S_1, \ldots, S_n$ be a non-negative sub-martingale, then for any $c > 0$ it holds that $\mathbb{P}\big[\sup_k S_k \geq c\big] \leq {}^{\mathbb{E}[S_n]}/_{c}$.*

3.4 Full-Information Coin Flip

We start with the formal definition of full-information coin-flipping protocols.

Definition 3.12 (Full-information coin-flipping protocols)**.** *A protocol Π is a* full-information coin-flipping protocol *if it is* stateless *(i.e., the parties keep no private state between the different communication rounds),*[10] *single turn (i.e., each turn consists of a* single *party broadcasting a string), and the parties' common output is a deterministic Boolean function of the transcript.*

Remark 3.13 (Many messages per communication round)**.** *Our attack readily applies to the model in which many parties might broadcast a message in a single round, as long as the adversary controls the message arrival order in this round (as assumed in Tauman Kalai et al. [27]). The setting in which many messages per round are allowed, and the adversary has no control over the arrival order, is equivalent (at least under a natural formulation of this model) to the static adversary cases, in which we know that $\Theta(n/\log n)$ corruptions (n being the number of parties) are required.*

Notation 3.14. *We associate the the following notation with an n-party, ℓ-party, full-information coin-flipping protocol Π:*

- *Let $\mathrm{Msg}^{\Pi} = (\mathrm{Msg}^{\Pi}_1, \ldots, \mathrm{Msg}^{\Pi}_\ell)$ denote a random transcript (i.e., parties' messages) of Π.*
- *For partial transcript $\mathrm{msg}_{\leq i} \in \mathrm{Supp}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big)$, let $\Pi\big(\mathrm{msg}_{\leq i}\big) := \mathbb{E}\big[\Pi\big(\mathrm{Msg}^{\Pi}\big) \mid \mathrm{Msg}^{\Pi}_{\leq i} = \mathrm{msg}_{\leq i}\big]$. (I.e., the expected outcome of Π given $\mathrm{msg}_{\leq i}$.) We let $\mathbb{E}\big[\Pi\big] := \Pi()$, and refer to this quantity as the* expected outcome of Π.
- *For $\mathrm{msg}_{<i} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi}_{<i})$, let $\mathsf{NxtParty}(\mathrm{msg}_{<i}) \in [n]$ be the identity of the party to send the i^{th} message, as determined by $\mathrm{msg}_{<i}$.*
- *For a party $\mathsf{P} \in [n]$ and transcript $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$, let $\mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg}) := \{i \in [\ell] \colon \mathsf{NxtParty}(\mathrm{msg}_{<i}) = \mathsf{P}\}$.*
- *For $\mathrm{msg}_{\leq j} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi}_{<i})$ and $i < j$ let $\mathrm{Speaker}_i(\mathrm{msg}_{\leq j}) = \mathsf{NxtParty}(\mathrm{msg}_{<i})$.*
- *For $\mathrm{msg}_{\leq i} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi}_{\leq i})$, let $\mathrm{jump}^{\Pi}\big(\mathrm{msg}_{\leq i}\big) := \Pi\big(\mathrm{msg}_{\leq i}\big) - \Pi(\mathrm{msg}_{<i})$. (I.e., $\mathrm{jump}^{\Pi}\big(\mathrm{msg}_{\leq i}\big)$ is the increment in expectation caused by the i^{th} message.)*

[10] Since we consider attackers of *unbounded* computational power, this assumption is without loss of generality: given a stateful protocol we can apply our attack on its stateless variant in which each party, samples its state *conditioned on the current public transcript* before it acts. It is easy to see that an attack on the stateless variant implies an attack of the same quality on the original (stateful) protocol.

3.4.1 Adaptive Adversaries

Definition 3.15 (Adaptive adversary)**.** *A t-*$\mathsf{adaptive\ adversary}$ *for a full-information coin-flipping protocol in an* $\mathsf{unbounded}$ *algorithm that can take the following actions during the protocol execution.*

1. Before *each communication round, it can decide to add the next to speak party to the corrupted party list, as long as the size of this list does not exceed t.*
2. *In a communication round where a corrupted party is speaking, the adversary has* $\mathsf{full\ control}$ *over the message it sends but is bound to send a valid message (i.e., in the protocol message space support).*

We make use of the following definitions and properties for such adversaries.

The attacked protocol.

Definition 3.16 (The attacked protocol)**.** *Given a full-information coin-flipping protocol Π and a deterministic (adaptive) adversary A attacking it, let Π_{A} be the full-information coin-flipping protocol induced by this attack: the parties act according to Π while emulating A. Once a party realizes it is corrupted, it acts according to the instruction of (the emulated) A. For non-deterministic A, let Π_{A} be the distribution over protocols induced by the randomness of A.*

Derandomization.

Proposition 3.17 (Attacker derandomization)**.** *For an adversary A acting on a full-information coin-flipping protocol Π there exist* $\mathsf{deterministic}$ *adversaries A^+ and A^- such that $\mathbb{E}\left[\Pi_{\mathsf{A}^+}\right] \geq \mathbb{E}\left[\Pi_{\mathsf{A}}\right]$ and $\mathbb{E}\left[\Pi_{\mathsf{A}^-}\right] \leq \mathbb{E}\left[\Pi_{\mathsf{A}}\right]$.*

Proof. By simple expectation arguments over the randomness of A. □

Composition of adaptive adversaries.

Definition 3.18 (Adaptive adversary composition)**.** *Let Π be a coin-flipping protocol, let A be an adaptive adversary for Π, and let B be an adaptive adversary for Π_{A}. The adversary $\mathsf{B} \circ \mathsf{A}$ on Π is defined as follows:*

Algorithm 3.19 (Adversary $\mathsf{B} \circ \mathsf{A}$ on Π)**.**

```
For $i := 1$ to NumMsgs($\Pi$):
    If $\mathsf{C} \in \{\mathsf{A}, \mathsf{B}\}$ would like to modify the $i^{\text{th}}$ message, corrupt the current party (if not already corrupted), and alter its message according to $\mathsf{C}$ (giving priority to $\mathsf{B}$ over $\mathsf{A}$).
```

It is clear that if A is k_{A}-adaptive and B is k_{B}-adaptive, then $\mathsf{B} \circ \mathsf{A}$ is $(k_{\mathsf{A}} + k_{\mathsf{B}})$-adaptive.

Proposition 3.20. *Let Π, A and B be as in Definition 3.18, then $\mathbb{E}\left[\Pi_{\mathsf{B} \circ \mathsf{A}}\right] = \mathbb{E}\left[(\Pi_{\mathsf{A}})_{\mathsf{B}}\right]$.*

Proof. It is clear that $(\Pi_{\mathsf{A}})_{\mathsf{B}}$ and $\Pi_{\mathsf{B} \circ \mathsf{A}}$ induce the same distribution on the protocol tree of Π, and thus induce the same output distribution. □

Composition of strongly adaptive adversaries.

Definition 3.21 (Strongly adaptive adversary composition). *Let* Π *be a coin-flipping protocol, let* A *be a strongly adaptive adversary for* Π*, and let* B *be a strongly adaptive adversary for* Π_{A}*. The adversary* $\mathsf{B} \circ \mathsf{A}$ *on* Π *is defined as follows:*

Algorithm 3.22 (Adversary $\mathsf{B} \circ \mathsf{A}$ on Π).

```
For $i := 1$ to $\mathrm{NumMsgs}(\Pi)$:
    1. Let $\mathrm{msg}_i$ be the message sent in the $i^{\mathrm{th}}$ round.
    2. Emulate the $i^{\mathrm{th}}$ round of $\mathsf{A}$, with input $\mathrm{msg}_i$. If $\mathsf{A}$ wishes to alter $\mathrm{msg}_i$ to $\widehat{\mathrm{msg}}$, corrupt the current party (if not already corrupted), alter the sent message to $\widehat{\mathrm{msg}}$, and set $\mathrm{msg}_i \leftarrow \widehat{\mathrm{msg}}$.
    3. Emulate the $i^{\mathrm{th}}$ round of $\mathsf{B}$, with input $\mathrm{msg}_i$. If $\mathsf{B}$ wishes to alter $\mathrm{msg}_i$ to $\widehat{\mathrm{msg}}$, corrupt the current party (if not already corrupted) and alter the sent message to $\widehat{\mathrm{msg}}$.
```

It is clear that if A is k_{A}-strongly adaptive and B is k_{B}-strongly adaptive, then $\mathsf{B} \circ \mathsf{A}$ is $(k_{\mathsf{A}} + k_{\mathsf{B}})$-strongly adaptive.

Proposition 3.23. *Let* Π*,* A *and* B *be as in Definition 3.21, then* $\mathbb{E}\big[\Pi_{\mathsf{B}\circ\mathsf{A}}\big] = \mathbb{E}\big[(\Pi_{\mathsf{A}})_{\mathsf{B}}\big]$.

Proof. It is clear that $(\Pi_{\mathsf{A}})_{\mathsf{B}}$ and $\Pi_{\mathsf{B}\circ\mathsf{A}}$ induce the same distribution on the protocol tree of Π, and thus induce the same output distribution. □

3.5 Useful Inequalities

We use the following standard inequalities.

Fact 3.24. *For* $-\frac{1}{2} \leq x$ *it holds that* $x \log(1+x) \leq 2x^2$.

Fact 3.25. *For* $0 \leq x \leq 1$ *it holds that* $x \log x \geq -1$.

4 Biasing Robust Coin Flip

In this section, we present an attack for biasing robust coin-flipping protocols. To simplify notation, we focus on robustness towards 0; see below. In the following, n typically represents the number of parties of the robust ("non-normal") protocol. We make use of the following notation.

Notation 4.1. *For* $n \in \mathbb{N}$*, let* $\varepsilon_n := 1/\sqrt[50]{\log\log n}$*,* $\lambda_n := 100/\varepsilon_n^5 = 100 \cdot \sqrt[10]{\log\log n}$ *and* $\delta_n := 1/\log^2 n$.

The main result of this section is stated below.

Definition 4.2 (Robust coin-flipping protocols). *An* ℓ*-round, full-information, coin-flipping protocol* Π *is* (α,β)-robust*, if* $\mathbb{P}\big[\exists i \in [\ell] \colon \min\big(\mathrm{Supp}\big(\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i}\big)\big) \leq -\alpha\big] \leq \beta$.

Theorem 4.3 (Biasing robust coin-flipping protocols). *Let* Π *be an* n*-party,* $\big(1/(\lambda_n \cdot \sqrt{n}), \delta_n\big)$*-robust, full-information coin-flipping protocol such that* $\mathbb{E}\big[\Pi\big] \geq \varepsilon_n$*. Then there exists an* $O(\sqrt{n} \cdot \log n)$*-adaptive adversary* A *such that* $\mathbb{E}\big[\Pi_{\mathsf{A}}\big] \geq 1 - \varepsilon_n$.

We start, Section 4.1, by proving a variant of Theorem 4.3 for "normal" coin-flipping protocols. Informally, in a *normal* coin-flipping protocol, parties participating in multiple rounds don't have "too large of an influence" on the protocol's outcome, though a party sending a single message may greatly influence the outcome. In Section 4.2, we leverage this attack for proving Theorem 4.3 by transforming any given protocol to a normal protocol, and show that the guaranteed attack on the latter protocol yields an attack of essentially the same quality on the original protocol.[11]

4.1 Biasing Normal Robust Coin Flip

Normal coin-flipping protocols are coin-flipping protocols of a concrete message-ownership structure. In Section 4.2, we show that an arbitrary coin-flipping protocol can be viewed, with some parameter adaptation, as a normal coin-flipping protocol.

Notation 4.4 (Parties classification)**.**

Large-jump parties. *A party* P *has* a large jump in msg*, if* $\exists i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})$ *s.t.*

$\mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big] \geq 1/(n\lambda_n)$.

Small-jump parties. *A party* P *has* small jumps in msg *if it participates (sends at least one message) but has no large jumps in* msg*.*

Unfulfilled parties. *A small-jumps party is* unfulfilled in msg *if*

$\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big] < 1/(n\lambda_n)$.

Similarly, we refer to jumps as small or big jumps, based on their variance, i.e., large jumps are jumps of variance $\geq 1/(n\lambda_n)$ and small jumps are jumps of variance $< 1/(n\lambda_n)$.

Definition 4.5 (Normal coin-flipping protocols)**.** *Let* Π *be a* t*-party,* ℓ*-round, full-information coin-flipping protocol and let* $n \in \mathbb{N}$*. We say* Π *is* n-normal*, if the following hold:*

Large-jump parties send a single message. $|\mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})| = 1$*, for every large-jump party* P *in* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$*.*

Small-jump party has bounded overall variance. $\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big] \leq 2/(n\lambda_n)$*, for every small-jumps party* P *in* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$*.*

Bounded number of unfulfilled parties. *In every transcript* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *there are at most* n unfulfilled *(participating) parties.*

At most one non-robust party. *There exists at most one party* $\mathsf{NonRbst} \in [t]$ *such that for every transcript* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *and* $i \in [\ell] \setminus \mathrm{SentBy}_{\mathsf{NonRbst}}(\mathrm{msg})$*:*

$$\min\big(\mathrm{Supp}\big(\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big)\big) > -1/(\lambda_n\sqrt{n})$$

(Namely, only NonRbst *can cause large negative jumps.)*

[11] It is worth mentioning that the shift from attacking arbitrary protocols to normal protocols is merely done for notational convince, and nothing exciting is hidden under the hood of the transformation above.

Intuitively, a protocol is n-normal if no small-jump parties have too large influence (as a function of n). In Section 4.2, we show how to turn any n-party protocol into a t-party, n-normal protocol (for some $t > n$).

We present an attack on normal protocols that either corrupts (essentially) *all* messages sent by a party or corrupts *none* of them.

Theorem 4.6 (Biasing normal robust coin-flipping protocols)**.** *For every n-normal, ℓ-round, $({}^{1}\!/\!_{(\lambda_n\sqrt{n})}, \delta_n)$-robust, full-information coin-flipping protocol Π with $\mathbb{E}\big[\Pi\big] \geq \varepsilon_n$, there exists $O(\sqrt{n} \cdot \log n)$-adaptive adversary A with $\mathbb{E}\big[\Pi_{\mathsf{A}}\big] \geq 1 - \varepsilon_n$.*

That is, if Π is n-normal and $({}^{1}\!/\!_{(\lambda_n\sqrt{n})}, \delta_n)$-robust, it can be biased by an $O(\sqrt{n} \cdot \log n)$-adaptive adversary.

4.1.1 Gently Biasing a Distribution

We begin by introducing a method of biasing distributions to increase their expectation under some utility function. Our adversary will then use this technique to modify the messages of the corrupted parties, with the utility function being the change induced in the protocol's expected outcome.

Definition 4.7 (Biased distribution)**.** *Let P be a distribution, let $\alpha > 0$ and let $f\colon \operatorname{Supp}(P) \mapsto [-{}^{1}\!/\!_{\alpha}, \infty)$ be such that $\mathbb{E}\big[f(P)\big] = 0$. The distribution $\operatorname{Biased}^{f}_{\alpha}\big(P\big)$ is defined by $\mathbb{P}\big[\operatorname{Biased}^{f}_{\alpha}\big(P\big) = e\big] = \mathbb{P}\big[P = e\big] \cdot (1 + \alpha \cdot f(e))$.*

It is easy to verify this is indeed a distribution. If f is the identity function, we sometimes omit it from the above notation. We use the following properties of the Biased distribution (proven in Section 4.3).

Lemma 4.8 (Properties of the Biased distribution)**.** *For any P, α and f as in Definition 4.7, it holds that*

1. $\mathbb{E}\big[f\big(\operatorname{Biased}^{f}_{\alpha}\big(P\big)\big)\big] = \alpha \cdot \operatorname{Var}\big[f(P)\big]$.
2. $D_{\mathrm{KL}}\big(\operatorname{Biased}^{f}_{\alpha}\big(P\big) \,\|\, P\big) \leq 2\alpha^2 \cdot \operatorname{Var}\big[f(P)\big]$.
3. $\big(p \cdot \operatorname{Biased}^{f}_{\alpha}\big(P\big) + (1-p) \cdot P\big) \equiv \operatorname{Biased}^{f}_{p\cdot\alpha}\big(P\big)$ *for any* $p \in [0,1]$.
4. *There exist a distribution (A, B) which couples P and $\operatorname{Biased}^{f}_{\alpha}\big(P\big)$, i.e., $A \equiv P$ and $B \equiv \operatorname{Biased}^{f}_{\alpha}\big(P\big)$, such that for any $(a, b) \leftarrow (A, B)$ it holds that $f(B) \geq f(A)$.*

4.1.2 The Attack

Using the above tool, we define our attack on normal coin-flipping protocols. Fix an n-normal, ℓ-round, full-information coin-flipping protocol Π, such that $\mathbb{E}\big[\Pi\big] \geq \varepsilon_n$ and $\mathbb{P}\big[\operatorname{SentBy}_{\mathsf{NonRbst}}(\operatorname{Msg}^{\Pi}) \neq \emptyset\big] \leq \delta_n$. When clear from the context, we omit the subscript n from the notations $\varepsilon_n, \lambda_n, \delta_n$. The $O(\sqrt{n} \cdot \log n)$-adaptive attacker A on Π is defined as follows.

Algorithm 4.9 (The attacker A)**.**

For $i := 1$ **to** ℓ, *do the following* before *the* i^{th} *message is sent:*

1. *Let* P *be the party about to send the* i^{th} *message. If* $\mathsf{P} = \mathsf{NonRbst}$*, abort.*
2. *Let* $\mathrm{msg}_{<i}$ *denote the messages sent in the previous rounds. Let* Q_i *be the distribution* $\mathrm{Msg}^{\Pi}_i\big|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}$*, let* $\mathrm{jump}_i(x) := \mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, x)$ *and let* $v_i := \mathrm{Var}\big[\mathrm{jump}_i(Q_i)\big]$.
3. **If** *this is the first message sent by* P*, corrupt* P *according to the following method:*
   (a) **If** P *is a large-jump party, i.e.,* $v_i \geq 1/\lambda n$*, corrupt it with probability* $\lambda^2 \cdot \sqrt{v_i}$*.* [12]
   (b) **Else** *(*P *is a small-jumps party), corrupt* P *with probability* $\lambda^2/\sqrt{n}$*.*
4. **If** P *is in the corrupted parties pool:*
   (a) **If** P *is a large-jump party, instruct* P *to broadcast its next message according to* $\mathrm{Biased}^{\mathrm{jump}_i}_{1/\sqrt{v_i}}\big(Q_i\big)$.
   (b) **Else***:*
      i. **If** $\mathbb{P}\big[\mathsf{P}$ *is corrupted by* $\mathsf{A} \mid \mathrm{Msg}^{\Pi_{\mathsf{A}}}_{<i} = \mathrm{msg}_{<i}\big] \leq 16\lambda^2/\sqrt{n}$*,*[13] *instruct* P *to broadcast its next message according to* $\mathrm{Biased}^{\mathrm{jump}_i}_{\sqrt{n}}\big(Q_i\big)$.
      ii. **Else***, instruct* P *to sample its next message* honestly *(i.e., according to* Q_i*).*

The main difference between the above attacker and its simplified variant presented in Section 2, is that the above attacker might decide not to modify a message of an already corrupted party (see Step 4b). This change enables us to easily bound the KL-divergence between the attacked and all-honest distributions, a bound that plays a critical role in our analysis.[14] In the rest of this section, we analyze the expected outcome of Π_{A} and the number of parties A corrupts. The proof makes use of the following random variables associated with a random execution of Π_{A}.

Notation 4.10 (Random variables associated with a random execution of Π_{A})**.**

- $\mathrm{Msg}^{\mathsf{A}} = (\mathrm{Msg}^{\mathsf{A}}_1, \ldots, \mathrm{Msg}^{\mathsf{A}}_\ell) := \mathrm{Msg}^{\Pi_{\mathsf{A}}}$.
- $S_k := \Pi\big(\mathrm{Msg}^{\mathsf{A}}_{\leq k}\big)$.

  (Note that $S_0, \ldots, S_\ell$ *is* sub-martingale *with respect to* $\mathrm{Msg}^{\mathsf{A}}$*.)*
- $X_k := S_k - S_{k-1}$.

  (Note that $X_1, \ldots, X_\ell$ *are the jumps induced by the attacked execution, i.e.,* $X_i = \mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\mathsf{A}}_{\leq k}\big)$*. Also note that* $S_0 = \mathbb{E}\big[\Pi\big] \geq \varepsilon$ *and* $S_\ell = S_0 + \sum_{i=1}^{\ell} X_i$*.)*
- CorruptedParties*: the set of parties corrupted in this execution of* A *on* Π.

  (Note that CorruptedParties *is* not *determined by* $\mathrm{Msg}^{\mathsf{A}}$*, as there is additional randomness involved.)*

[12] $\lambda^2 \cdot \sqrt{v_i}$ is indeed in the interval $[0,1]$: $\mathsf{P} \neq \mathsf{NonRbst}$ implies that $v_i \leq 1/\lambda\sqrt{n}$.

[13] Recall that Π_{A} is the protocol induced by the attack of A on Π, and or $\mathrm{Msg}^{\Pi_{\mathsf{A}}}_{<i}$ are the first $i-1$ messages in its execution. Hence, we are defining the strategy of A in the i^{th} round using its strategy in the first $i-1$ rounds, so this self-reference is well defined.

[14] We are not sure whether this change is mandatory for the attack to go through or merely an artifact of our proof technique that bounds the KL divergence between the attacked and honest execution (see Claim 4.12).

- *$Q_1, \ldots, Q_\ell$: the value of these variables as computed by A.*

We prove Theorem 4.6 via three key observations. The first observation, proved in Section 4.1.3, guarantees a per-round coupling between the change in expected outcome induced by the attack and what would have been the change in an honest execution (conditioned on previous messages).

Claim 4.11 (Coupling honest and attacked conditional distributions)**.** *There exists a random variable $Y = (Y_1, \ldots, Y_\ell)$ jointly distributed with $\mathrm{Msg}^{\mathsf{A}}$, such that for every $i \in [\ell]$:*

1. *$X_i \ge Y_i$.*
2. *Conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i}$: Y_i is distributed like $\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\le i}\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{Msg}^{\mathsf{A}}_{<i}}\right)$, and is independent of $Y_{<i}$ and $\left\{\mathrm{Speaker}_i(\mathrm{Msg}^{\mathsf{A}}_{<i}) \in \mathrm{CorruptedParties}\right\}$.*

That is, Y_i is distributed like the (conditional) change in expected outcome induced by the i^{th} step *if it were carried out honestly*, and is never larger than the (conditional) change induced by the i^{th} step of the attacked execution. It is easy to verify that $\mathbb{E}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}, Y_{<k}\right] = 0$, i.e., $\sum_{i=1}^{k} Y_i$ is a martingale difference sequence with respect to $(\mathrm{Msg}^{\mathsf{A}}_k, Y_k)$. For the rest of this section, let Y be the random variable guaranteed by Claim 4.11.

Next, we consider the set of indices corresponding to robust jumps, defined by

$$\mathrm{RobustJumps} := [\ell] \setminus \mathrm{SentBy}_{\mathsf{NonRbst}}(\mathrm{Msg}^{\mathsf{A}}) \tag{12}$$

Note that RobustJumps is a random set, determined by $\mathrm{Msg}^{\mathsf{A}}$. The following observation (proved in Section 4.1.4) states that the overall conditional variance of Y contributed by the robust jumps is small, which implies that the variance of $\sum Y_i$ is small. It follows that $\sum Y_i$ is typically not "too small", and since $X_i \ge Y_i$, that $\sum X_i$ is typically not too small.

Claim 4.12 (Bounding Y's conditional variance)**.** $\mathbb{E}\left[\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] < {}^{2}\!/\!_{\lambda}$.

Finally, in Section 4.1.5 we prove that attacked execution does not deviate too much, in KL-divergence terms, from the honest execution. This implies that, with overwhelming probability, $\mathsf{NonRbst}$ does not participate in the protocol (since it participated in the original protocol with very small probability).

Claim 4.13 (Bounding KL-Divergence between attacked and honest executions)**.** $D_{\mathrm{KL}}\left(\mathrm{Msg}^{\mathsf{A}} \,\|\, \mathrm{Msg}^{\Pi}\right) \le 16^3 \lambda^3$.

Equipped with Claims 4.11 to 4.13, we are ready to prove Theorem 4.6.

Proof of Theorem 4.6.

Expected outcome. We start by analyzing the expected bias induced by A. Note that

$$\mathrm{Var}\left[\textstyle\sum_{i \in \mathrm{RobustJumps}} Y_i\right] = \textstyle\sum_{i=1}^{n} \mathrm{Var}\left[Y_i \cdot \mathbb{1}_{i \in \mathrm{RobustJumps}}\right] = \mathbb{E}\left[\textstyle\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] \le {}^{2}\!/\!_{\lambda} \tag{13}$$

The first and second equalities holds by Facts 3.7 and 3.9 respectively, since $\mathbb{E}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}, Y_{<k}\right] = 0$ and $\mathbb{1}_{i\in\mathrm{RobustJumps}}$ is determined by $\mathrm{Msg}^{\mathsf{A}}_{<k}$. The inequality holds by Claim 4.12.

By Equation (13) and Chebyshev's inequality, (remember that $\mathbb{E}\left[\sum_{i\in\mathrm{RobustJumps}} Y_i\right] = 0$)

$$\mathbb{P}\left[\sum_{i\in\mathrm{RobustJumps}} Y_i \le -{\varepsilon}/{2}\right] \le \mathbb{P}\left[\left|\sum_{i\in\mathrm{RobustJumps}} Y_i\right| \ge {\varepsilon}/{2}\right] \le \left({}^{\varepsilon/2}\big/\sqrt{2/\lambda}\right)^{-2} = {8}/{\lambda\varepsilon^2} = \tfrac{8}{\varepsilon^{2}.100/\varepsilon^5} \le \varepsilon/4 \tag{14}$$

We next show that with overwhelming probability $\mathrm{RobustJumps} = [\ell]$, namely $\mathsf{NonRbst}$ does not participate in the execution. Let Bad^{Π} be the event $\left\{\mathrm{SentBy}_{\mathsf{NonRbst}}(\mathrm{Msg}^{\Pi}) \ne \emptyset\right\}$, and let Bad^{A} be the event $\left\{\mathrm{SentBy}_{\mathsf{NonRbst}}(\mathrm{Msg}^{\mathsf{A}}) \ne \emptyset\right\}$. By assumption, $\mathbb{P}\left[\mathrm{Bad}^{\Pi}\right] \le \delta$, and by Claim 4.13 and data-processing of KL-Divergence,

$$D_{\mathrm{KL}}\left(\mathbb{1}_{\mathrm{Bad}^A} \,\|\, \mathbb{1}_{\mathrm{Bad}^{\Pi}}\right) \le D_{\mathrm{KL}}\left(\mathrm{Msg}^{\mathsf{A}} \,\|\, \mathrm{Msg}^{\Pi}\right) \le 16^3\lambda^3 \tag{15}$$

We also note that,

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathbb{1}_{\mathrm{Bad}^A} \,\|\, \mathbb{1}_{\mathrm{Bad}^{\Pi}}\right) &= \mathbb{P}\left[\mathrm{Bad}^A\right] \cdot \log\left(\frac{\mathbb{P}\left[\mathrm{Bad}^A\right]}{\mathbb{P}\left[\mathrm{Bad}^{\Pi}\right]}\right) + \left(1 - \mathbb{P}\left[\mathrm{Bad}^A\right]\right) \cdot \log\left(\frac{1 - \mathbb{P}\left[\mathrm{Bad}^A\right]}{1 - \mathbb{P}\left[\mathrm{Bad}^{\Pi}\right]}\right) \qquad (16) \\ &= \mathbb{P}\left[\mathrm{Bad}^A\right] \cdot \log\left(\frac{\mathbb{P}\left[\mathrm{Bad}^A\right]}{\mathbb{P}\left[\mathrm{Bad}^{\Pi}\right]}\right) + \left(1 - \mathbb{P}\left[\mathrm{Bad}^A\right]\right) \cdot \left(\log\left(1 - \mathbb{P}\left[\mathrm{Bad}^A\right]\right) - \log\left(1 - \mathbb{P}\left[\mathrm{Bad}^{\Pi}\right]\right)\right) \\ &\ge \mathbb{P}\left[\mathrm{Bad}^A\right] \cdot \log\left(\frac{\mathbb{P}\left[\mathrm{Bad}^A\right]}{\mathbb{P}\left[\mathrm{Bad}^{\Pi}\right]}\right) + (-1 + 0) \ge \mathbb{P}\left[\mathrm{Bad}^A\right] \cdot \log\left(\frac{\mathbb{P}\left[\mathrm{Bad}^A\right]}{\delta}\right) - 1. \end{aligned}$$

where the penultimate inequality follows by Fact 3.25. We now show that

$$\mathbb{P}\left[\mathrm{Bad}^A\right] \le {\varepsilon}/{4} \tag{17}$$

Indeed, assuming Equation (17) does not hold, then (for sufficiently large n),

$$\sqrt{\log\log n} \le \frac{\log\left({\log^2 n}/{4\sqrt[50]{\log\log n}}\right)}{4\sqrt[50]{\log\log n}} - 1 \le D_{\mathrm{KL}}\left(\mathbb{1}_{\mathrm{Bad}^A} \,\|\, \mathbb{1}_{\mathrm{Bad}^{\Pi}}\right) \tag{18}$$

$$D_{\mathrm{KL}}\left(\mathbb{1}_{\mathrm{Bad}^A} \,\|\, \mathbb{1}_{\mathrm{Bad}^{\Pi}}\right) \le 16^3\lambda^3 < \sqrt{\log\log n} \tag{19}$$

Inequality (18) follows by Equation (16), and Inequality (19) by Equation (15). Overall—yielding a contraction. Combining Equations (14) and (17) yields,

$$\begin{aligned} \mathbb{E}\left[\Pi_{\mathsf{A}}\right] &= \mathbb{P}\left[S_\ell = 1\right] = \mathbb{P}\left[S_\ell > 0\right] = \mathbb{P}\left[S_0 + \sum_{i=1}^{\ell} X_i > 0\right] = \mathbb{P}\left[\sum_{i=1}^{\ell} X_i > -S_0\right] \ge \mathbb{P}\left[\sum_{i=1}^{\ell} X_i > -\varepsilon\right] \\ &\ge \mathbb{P}\left[\sum_{i\in\mathrm{RobustJumps}} Y_i > -\varepsilon\right] - \mathbb{P}\left[\mathrm{SentBy}_{\mathsf{NonRbst}}(\mathrm{Msg}^{\mathsf{A}}) \ne \emptyset\right] \ge (1 - {\varepsilon}/{4}) - {\varepsilon}/{4} \ge 1 - {\varepsilon}/{2}. \end{aligned}$$

The second inequality follows by Claim 4.11. The penultimate inequality follows by Equations (14) and (17).

Number of corruptions. It is left to argue that A does not perform too many corruptions. We calculate the *expected* number of corruptions, and bound the *actual* number of corruptions using Markov's inequality. We introduce several additional notations. Let SmallParties and LargeParties be the (random) sets of small-jumps and large-jump parties (that participate in the execution) with respect to $\mathrm{Msg}^{\mathsf{A}}$, respectively. Let $\mathrm{SmallJumps} := \left\{k \in [\ell] \colon \mathrm{Speaker}_k(\mathrm{Msg}^{\mathsf{A}}) \in \mathrm{SmallParties}\right\}$ be the set of small jumps, and let $\mathrm{LargeJumps} := \left\{k \in [\ell] \colon \mathrm{Speaker}_k(\mathrm{Msg}^{\mathsf{A}}) \in \mathrm{LargeParties}\right\}$ be the set of large jumps. Note that all of the above random sets are determined by $\mathrm{Msg}^{\mathsf{A}}$. We first notice that since a small-jumps party is corrupted with probability ${}^{\lambda^2}/_{\sqrt{n}}$, it holds that

$$\mathbb{E}\left[|\mathrm{SmallParties} \cap \mathrm{CorruptedParties}|\right] = {}^{\lambda^2}/_{\sqrt{n}} \cdot \mathbb{E}\left[|\mathrm{SmallParties}|\right] \tag{20}$$

In addition, the definition of n-normal protocols stipulates that for any transcript of Π, there are at most n unfulfilled parties. Since each fulfilled (not unfulfilled) party contributes at least ${}^{1}/_{\lambda n}$ to the sum of variances, which is small by Claim 4.12, we deduce that

$$\mathbb{E}\left[|\mathrm{SmallParties}|\right] \leq 3n \tag{21}$$

Combining the above two observations yields the following bound on the number of corrupted small-jump parties:

$$\mathbb{E}\left[|\mathrm{SmallParties} \cap \mathrm{CorruptedParties}|\right] \leq {}^{\lambda^2}/_{\sqrt{n}} \cdot 3n = 3\lambda^2\sqrt{n}$$

As for large-jump parties, for any $k \in [\ell]$, partial transcript $t = \mathrm{msg}_{<k}$ and large-jump party P sending the k^{th} message, P is corrupted with probability $\lambda^2 \cdot \sqrt{\mathrm{Var}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k} = \mathrm{msg}_{<k}\right]} \leq \lambda^2 \cdot \sqrt{\lambda n} \cdot \mathrm{Var}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k} = \mathrm{msg}_{<k}\right]$. Thus, we have that

$$\begin{aligned} &\mathbb{E}\left[|\mathrm{LargeParties} \cap \mathrm{CorruptedParties}|\right] \\ &= \mathbb{E}\left[\textstyle\sum_{i \in \mathrm{LargeJumps}} \lambda^2 \cdot \sqrt{\mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]}\right] \leq \mathbb{E}\left[\textstyle\sum_{i \in \mathrm{LargeJumps}} \lambda^2 \cdot \sqrt{\lambda n} \cdot \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] \\ &\leq \lambda^3 \cdot \sqrt{n} \cdot \mathbb{E}\left[\textstyle\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] \leq \lambda^3 \cdot \sqrt{n} \cdot {}^{2}/_{\lambda} = 2\lambda^2\sqrt{n}. \end{aligned} \tag{22}$$

The first inequality follows by the definition of a large jump, i.e., $\mathrm{Var}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k} = \mathrm{msg}_{<k}\right] \geq {}^{1}/_{\lambda n}$, and last inequality by Claim 4.12. Therefore, the expected amount of corruptions is at most $5\lambda^2\sqrt{n}$. Hence, by Markov's inequality, with probability at least $1 - {}^{\varepsilon}/_{2}$ the amount of corruptions made by A is at most ${}^{10\lambda^3\sqrt{n}}/_{\varepsilon} < 10\lambda^4\sqrt{n}$.

Putting it together. Consider the adversary A' that acts just as A, but aborts (letting players continue the execution honestly) once the amount of corruptions surpasses $10\lambda^4\sqrt{n} = O(\sqrt{n} \cdot \log n)$. It holds that

$$\begin{aligned} \mathbb{E}\left[\Pi_{\mathsf{A}'}\right] &= \mathbb{P}\left[\Pi_{\mathsf{A}'} = 1\right] \geq \mathbb{P}\left[\Pi_{\mathsf{A}} = 1 \wedge |\mathrm{CorruptedParties}| < 10\lambda^4\sqrt{n}\right] \\ &\geq \mathbb{P}\left[\Pi_{\mathsf{A}} = 1\right] - \mathbb{P}\left[|\mathrm{CorruptedParties}| \geq 10\lambda^4\sqrt{n}\right] \geq 1 - {}^{\varepsilon}/_{2} - {}^{\varepsilon}/_{2} = 1 - \varepsilon, \end{aligned}$$

which concludes the proof of the theorem. □

4.1.3 Coupling X_i and Q_i, Proving Claim 4.11

Claim 4.14 (Restatement of Claim 4.11)**.** *There exists a random variable $Y = (Y_1, \ldots, Y_\ell)$ jointly distributed with $\mathrm{Msg}^{\mathsf{A}}$, such that for every $i \in [\ell]$:*

1. *$X_i \geq Y_i$.*
2. *Conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i}$: Y_i is distributed like $\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\leq i} \,|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{Msg}^{\mathsf{A}}_{<i}}\right)$, and is independent of $Y_{<i}$ and $\left\{\mathrm{Speaker}_i(\mathrm{Msg}^{\mathsf{A}}_{<i}) \in \mathrm{CorruptedParties}\right\}$.*

Proof. Fix $i \in [\ell]$ and denote $\mathsf{P} = \mathrm{Speaker}_i(\mathrm{Msg}^{\mathsf{A}})$.

Let C be the event $\{\mathsf{P} \in \mathrm{CorruptedParties}\}$. Let C_L be the event $\{C \wedge \mathsf{P} \in \mathrm{LargeParties}\}$. Also let C_S be the event $\{(C \wedge \mathsf{P} \in \mathrm{SmallParties}) \wedge (\mathbb{P}\left[C \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}\right] \leq {}^{16\lambda^2}/\!_{\sqrt{n}})\}$, i.e., P is small-jump corrupted party and A instructs it to alter the current message (see Step 4(b)ii of Algorithm 4.9). Finally, define the following random variable (determined by $\mathrm{Msg}^{\mathsf{A}}_{<i}$ and $\mathbb{1}_C$):

$$\alpha = \begin{cases} {}^{1}/_{\mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]} & \text{if } \mathbb{1}_{C_L} = 1 \\ \sqrt{n} & \text{if } \mathbb{1}_{C_S} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Now consider the (random) distribution (A, B) guaranteed by Lemma 4.8(4) with respect to $P = Q_i$, $f = \mathrm{jump}_i$ and α.[15] It is easy to verify that by construction (i.e., Algorithm 4.9): conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i}$ and $\mathbb{1}_C$, B distributes like $\mathrm{Msg}^{\mathsf{A}}_i$.

Now, conditioned on on $\mathrm{Msg}^{\mathsf{A}}_{\leq i}$ and $\mathbb{1}_C$ we sample $\widetilde{\mathrm{Msg}}_i \leftarrow A\big|_{B = \mathrm{Msg}^{\mathsf{A}}_i}$, independently of $\widetilde{\mathrm{Msg}}_{<i}$.

Finally, we set $Y_i = \mathrm{jump}_i(\widetilde{\mathrm{Msg}}_i)$, and from the previous observation it immediately follows that Y_i is distributed like $\mathrm{jump}_i\left(\mathrm{Msg}^{\Pi}_{\leq i} \,|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{Msg}^{\mathsf{A}}_{<i}}\right)$.

It is clear that conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i}$, $\widetilde{\mathrm{Msg}}_i$ is distributed like Q_i. In addition, it is independent of $\mathbb{1}_C$ because it is distributed the same no matter the value of $\mathbb{1}_C$—it is even distributed the same conditioned on $\mathbb{1}_C, \mathbb{1}_{C_S}, \mathbb{1}_{C_L}$. And so the same is true for Y_i. As for the independence from $Y_{<i}$, it follows immediately by the independence from $\widetilde{\mathrm{Msg}}_{<i}$.

All that is left to show is that $X_i \geq Y_i$.

$$Y_i = \mathrm{jump}_i(\widetilde{\mathrm{Msg}}_i) \equiv \mathrm{jump}_i(A\big|_{B = \mathrm{Msg}^{\mathsf{A}}_i}) \leq \mathrm{jump}_i(\mathrm{Msg}^{\mathsf{A}}_i) = X_i$$

where the inequality follows by the property $f(B) \geq f(A)$ guaranteed by Lemma 4.8(4). □

4.1.4 Bounding Y's Conditional Variance, Proving Claim 4.12

Claim 4.15 (Restatement of Claim 4.12)**.** $\mathbb{E}\left[\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] < {}^{2}/_{\lambda}$.

Proof. Immediately follows by Claims 4.17 and 4.18, given below. Claim 4.17 states that $\mathbb{E}\left[\sum_i \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] \leq {}^{1}/_{\lambda}$ when i ranges over LargeJumps, and Claim 4.18 state the same when i ranges over SmallJumps. Hence, $\mathbb{E}\left[\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\right]\right] < {}^{2}/_{\lambda}$. □

[15] $(\mathrm{jump}_i(\cdot) = \mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, \cdot))$

In the following, we use a conditional variant of the biased distribution.

Definition 4.16 (Conditional variant of Biased)**.** *Let $\mathcal{Y}, \mathcal{Z}$ and η be jointly distributed random variables, and let $f \colon \operatorname{Supp}(\mathcal{Y}) \mapsto \mathbb{R}$ be a function, such that (1) η is determined by $\mathcal{Z}$; (2) $f(\mathcal{Y}) \geq -1/\eta$; (3) $\mathbb{E}\big[f(\mathcal{Y}) \mid \mathcal{Z}\big] = 0$. Define the random variable* $\mathrm{Biased}_{\eta}^{f}\big(\mathcal{Y} \mid \mathcal{Z}\big)$*, jointly distributed with $\mathcal{Z}$, by sampling* $\mathrm{Biased}_{\eta}^{f}\big(\mathcal{Y} \mid \mathcal{Z}\big) \leftarrow \mathrm{Biased}_{\eta}^{f}\big(\mathcal{Y}|_{\mathcal{Z}=\mathcal{Z}}\big)$.[16]

We now move to proving Claims 4.17 and 4.18.

Large jumps.

Claim 4.17. $\mathbb{E}\big[\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] < 1/\lambda$.

Proof. Let L_k be the event $\{k \in \mathrm{LargeJumps}\}$. Note that $\mathbb{1}_{L_k}$ is determined by $\mathrm{Msg}^{\mathsf{A}}_{<k}$. Compute,

$$\begin{aligned}
&\mathbb{1}_{L_k} \cdot \mathbb{E}\big[X_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big] && (23)\\
&= \mathbb{1}_{L_k} \cdot \Bigg(\lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big]} \cdot \mathbb{E}\big[\mathrm{Biased}_{1/\sqrt{\mathrm{Var}\left[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\right]}}\big(Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big) \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big] && (24)\\
&\qquad + \Big(1 - \lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big]}\Big) \cdot \mathbb{E}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big]\Bigg)\\
&= \mathbb{1}_{L_k} \cdot \lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big]} \cdot 1/\sqrt{\mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big]} \cdot \mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big] + 0 && (25)\\
&= \mathbb{1}_{L_k} \cdot \lambda^2 \cdot \mathrm{Var}\big[Y_k \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big].
\end{aligned}$$

Equality (24) follows by construction (see Step 3a and Step 4a of Algorithm 4.9) and Claim 4.11, and Equality (25) follows by Lemma 4.8(1). Hence,

$$\begin{aligned}
&\mathbb{E}\big[\textstyle\sum_{i=1}^{\ell} X_i\big] = \sum_{i=1}^{\ell} \mathbb{E}\big[X_i\big] = \sum_{i=1}^{\ell} \mathbb{E}\big[\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] = \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] && (26)\\
&\geq \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{1}_{L_k} \cdot \mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] && (27)\\
&= \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{1}_{L_k} \cdot \lambda^2 \cdot \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] && (28)\\
&= \lambda^2 \cdot \mathbb{E}\big[\textstyle\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big].
\end{aligned}$$

Equality (26) follows by the law of total expectation (Fact 3.8), Inequality (27) holds since $\mathbb{1}_{L_i}$ is determined by $\mathrm{Msg}^{\mathsf{A}}_{<i}$ and $\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] \geq 0$, and Equality (28) follows by Equation (23).

Thus,

$$\lambda^2 \cdot \mathbb{E}\big[\textstyle\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \leq \mathbb{E}\big[\sum_{i=1}^{\ell} X_i\big] \leq 1$$

and so $\mathbb{E}\big[\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \leq 1/\lambda^2 < 1/\lambda$. □

[16] Here $\mathcal{Y}|_{\mathcal{Z}=\mathcal{Z}}$ means we consider the distribution induced by conditioning $\mathcal{Y}$ on the current value of $\mathcal{Z}$.

Small jumps.

Claim 4.18. $\mathbb{E}\big[\sum_{i\in\mathrm{SmallJumps}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] < {}^{1}\!/\!_{\lambda}$.

For some party P denote by $\mathcal{I}_{\mathsf{P}} = \mathrm{SentBy}_{\mathsf{P}}\big(\mathrm{Msg}^{\mathsf{A}}\big)$, i.e., the indices in which P is the speaker. Also, consider the following definition for measuring the contribution of a small-jumps party.

Definition 4.19 (Contributional parties)**.** *A party* P *is said to be* contributional *if*

$$\mathbb{P}\big[\sum_{i\in\mathcal{I}_{\mathsf{P}}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] > \tfrac{1}{8\lambda n} \mid \mathsf{P}\in\mathrm{SmallParties}\big] \geq {}^{1}\!/\!_{8}.$$

Let ContribParties *be the set of contributional parties, and let* SmallContribParties := SmallParties ∩ ContribParties*.*

Note that being a contributional party is determined by the *protocol*, i.e., ContribParties does not depend on the transcript. In contrast, SmallContribParties, i.e., the set of contributional small-jumps parties, does depend on the transcript—because SmallParties is transcript-dependant. We make use of the following claim.

Claim 4.20. *For any contributional party* P*, it holds that*

$$\mathbb{E}\big[\sum_{i\in\mathcal{I}_{\mathsf{P}}} X_i \,|\, \mathsf{P}\in\mathrm{CorruptedParties}\cap\mathrm{SmallContribParties}\big] \geq {}^{1}\!/\!_{(256\lambda\cdot\sqrt{n})}.$$

We prove Claim 4.20 below, but first use it for proving Claim 4.18.

Proof of Claim 4.18. We start by using Claim 4.20 to lower-bound $\mathbb{E}\big[\sum_{i=1}^{\ell} X_i\big]$.

$$\begin{aligned}
\mathbb{E}\big[\textstyle\sum_{i=1}^{\ell} X_i\big] &= \textstyle\sum_{i=1}^{\ell}\mathbb{E}\big[X_i\big] = \sum_{i=1}^{\ell}\mathbb{E}\big[\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] = \mathbb{E}\big[\sum_{i=1}^{\ell}\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] && (29)\\
&\geq \mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\sum_{i\in\mathcal{I}_{\mathsf{P}}}\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] && (30)\\
&= \mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\sum_{i\in\mathcal{I}_{\mathsf{P}}} X_i\big] && (31)\\
&= \mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{ContribParties}}\mathbb{E}\big[\sum_{i\in\mathcal{I}_{\mathsf{P}}} X_i \,|\, \mathsf{P}\in\mathrm{CorruptedParties}\cap\mathrm{SmallContribParties}\big] && (32)\\
&\qquad\qquad\qquad \cdot\,\mathbb{P}\big[\mathsf{P}\in\mathrm{CorruptedParties}\cap\mathrm{SmallContribParties}\big]\big]\\
&\geq \mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{ContribParties}} {}^{1}\!/\!_{256\lambda\sqrt{n}}\cdot\mathbb{P}\big[\mathsf{P}\in\mathrm{CorruptedParties}\cap\mathrm{SmallContribParties}\big]\big] && (33)\\
&= \mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{ContribParties}} {}^{1}\!/\!_{256\lambda\sqrt{n}}\cdot\big(\mathbb{P}\big[\mathsf{P}\in\mathrm{SmallContribParties}\big]\cdot{}^{\lambda^2}\!/\!_{\sqrt{n}}\big)\big] && (34)\\
&= {}^{\lambda}\!/\!_{256n}\cdot\mathbb{E}\big[|\mathrm{SmallContribParties}|\big].
\end{aligned}$$

Inequality (29) follows by the law of total expectation (Fact 3.8). Inequality (30) holds since $\mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] \geq 0$. Equality (30) follows by the law of total expectation, since $\big\{\mathrm{Speaker}_i(\mathrm{Msg}^{\mathsf{A}})\in\mathrm{SmallContribParties}\big\}$ is determined by $\mathrm{Msg}^{\mathsf{A}}_{<i}$. Equality (32) also follows by the law of total expectation, combined with the fact that $\mathbb{E}\big[\sum_{i\in\mathcal{I}_{\mathsf{P}}} X_i \,|\, \mathsf{P}\notin\mathrm{CorruptedParties}\wedge\mathsf{P}\in\mathrm{SmallContribParties}\big] = 0$. Inequality (33) by Claim 4.20. Equality (34) by construction (see Step 3b of Algorithm 4.9).

Moving over from contributional parties, for any *non-contributional* party P, it holds that

$$\mathbb{E}\big[\sum_{i\in\mathcal{I}_{\mathsf{P}}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] \,|\, \mathsf{P}\in\mathrm{SmallParties}\big] < {}^{1}\!/\!_{8}\cdot{}^{2}\!/\!_{\lambda n} + {}^{1}\!/\!_{8\lambda n} < {}^{3}\!/\!_{8\lambda n}.$$

The first inequality follows by definition of a small jumps party and that of a non-contributional party, respectively. Finally, by the above inequality,

$$\mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{SmallParties}\setminus\mathrm{ContribParties}}\sum_{i\in\mathcal{I}_\mathsf{P}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \le \mathbb{E}\big[|\mathrm{SmallParties}|\big]\cdot {}^{3}/_{8\lambda n} \tag{35}$$

Denote $\gamma = \mathbb{E}\big[\sum_{i\in\mathrm{SmallJumps}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big]$. Each fulfilled (not unfulfilled) party contributes at least ${}^{1}/_{\lambda n}$ to the sum of conditional variances, and so there are at most $\gamma\lambda n$ such parties. Recalling that there are at most n unfulfilled parties, we deduce that

$$\mathbb{E}\big[|\mathrm{SmallParties}|\big] \le n + n\gamma\lambda \tag{36}$$

Assume towards a contradiction that $\gamma \ge {}^{1}/_{\lambda}$. By Equation (36), $\mathbb{E}\big[|\mathrm{SmallParties}|\big] \le 2n\gamma\lambda$, and thus by Equation (35)

$$\mathbb{E}\big[\textstyle\sum_{\mathsf{P}\in\mathrm{SmallParties}\setminus\mathrm{ContribParties}}\sum_{i\in\mathcal{I}_\mathsf{P}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \le 3\gamma/4 \tag{37}$$

We conclude that $\mathbb{E}\big[\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\sum_{i\in\mathcal{I}_\mathsf{P}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \ge \gamma/4$, and since by definition, every small jumps party contributes at most ${}^{2}/_{\lambda n}$ to the sum of conditional variances, we deduce that

$$\mathbb{E}\big[|\mathrm{SmallContribParties}|\big] \ge \gamma\lambda n/8 \tag{38}$$

Combining Equations (29) and (38), yields that

$$\mathbb{E}\Big[\sum_{i=1}^{\ell} X_i\Big] \ge {}^{\gamma\lambda^2}/_{2048} \tag{39}$$

It follows that $\mathbb{E}\big[S_\ell\big] = \mathbb{E}\big[S_0 + \sum_{i=1}^{\ell} X_i\big] \ge \varepsilon + {}^{\gamma\lambda^2}/_{2048}$, which is larger than 1 for sufficiently large n, yielding a contradiction. Hence, $\mathbb{E}\big[\sum_{i\in\mathrm{SmallJumps}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] = \gamma < {}^{1}/_{\lambda}$, concluding the proof. □

Proving Claim 4.20.

Proof of Claim 4.20. Fix a contributional party P, and consider the following events (jointly distributed with $\mathrm{Msg}^{\mathsf{A}}$):

- $C = \{\mathsf{P} \in \mathrm{CorruptedParties}\}$.
- $S = \{\mathsf{P} \in \mathrm{SmallParties}\}$.
- $L = \Big\{\sum_{i\in\mathcal{I}_\mathsf{P}}\mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] > {}^{1}/_{8\lambda n}\Big\}$, i.e.,, P has large conditional variance.
- $H = \big\{\forall k \in \mathcal{I}_\mathsf{P} \colon \mathbb{P}\big[\mathsf{P} \in \mathrm{CorruptedParties} \mid \mathrm{Msg}^{\mathsf{A}}_{<k}\big] < 16 \cdot {}^{\lambda^2}/_{\sqrt{n}}\big\}$, i.e., Step 4(b)ii of Algorithm 4.9 never happens for P.

We start by proving that $\mathbb{P}\big[H \wedge L \mid S\big]$ is large, and then use a KL-divergence argument to deduce that $\mathbb{P}\big[H \wedge L \mid S \wedge C\big]$ is large, i.e., P encounters large conditional variance even when it is a corrupted small-jumps party. This will imply that the change in expectation P induces (when a corrupted small-jumps party) is large.

To prove that $\mathbb{P}\big[H \wedge L \mid S\big]$ is large, we move to the conditional probability space where S occurs (i.e., P participates in the protocol as a small-jumps party). Consider the martingale $C_0, \ldots, C_\ell$ defined by $C_k := \mathbb{E}\big[\mathbb{1}_C \mid \mathrm{Msg}^{\mathsf{A}}_{\leq k}\big]$ (that is, C_k is the projection of the event C on the information held by $\mathrm{Msg}^{\mathsf{A}}_{\leq k}$). Under the conditioning P is a small-jumps party, therefore, the adversary corrupts P with probability $\lambda^2/\sqrt{n}$, i.e., $\mathbb{E}\big[\mathbb{1}_C\big] = \mathbb{P}\big[C\big] = \lambda^2/\sqrt{n}$. Thus by Doob's maximal inequality (see Fact 3.11), it holds that

$$\mathbb{P}\big[\neg H\big] = \mathbb{P}\big[\sup\{C_0, \ldots, C_\ell\} \geq 16 \cdot \lambda^2/\sqrt{n}\big] \leq 1/16 \tag{40}$$

Back to the non-conditional probability space, we deduce that

$$\mathbb{P}\big[L \wedge H \mid S\big] \geq \mathbb{P}\big[L \mid S\big] - \mathbb{P}\big[\neg H \mid S\big] \geq 1/8 - 1/16 = 1/16 \tag{41}$$

where $\mathbb{P}\big[L \mid S\big] \geq 1/8$ holds since P is contributional. We next bound $D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}\,\big|_{S \wedge C} \parallel \mathrm{Msg}^{\mathsf{A}}\,\big|_{S}\big)$. Letting $\widetilde{\mathrm{Msg}}$ denote the distribution $\mathrm{Msg}^{\mathsf{A}}\,\big|_{S \wedge C}$ compute,

$$\begin{align}
&D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}\,\big|_{S \wedge C} \parallel \mathrm{Msg}^{\mathsf{A}}\,\big|_{S}\big) = D_{\mathrm{KL}}\big(\widetilde{\mathrm{Msg}} \parallel \mathrm{Msg}^{\mathsf{A}}\,\big|_{S}\big) \tag{42}\\
&= \sum_{i=1}^{\ell} \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i} \wedge S \wedge C} \parallel \mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i} \wedge S}\big)\big] \tag{43}\\
&= \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i} \wedge C} \parallel \mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i}}\big)\big] \tag{44}\\
&\leq \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i} \wedge C} \parallel \mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i} \wedge \neg C}\big)\big] \tag{45}\\
&\leq \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\big(\mathrm{Biased}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, \cdot)}_{\sqrt{n}}\big(\mathrm{Msg}^{\Pi}_i\,\big|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big) \parallel \mathrm{Msg}^{\Pi}_i\,\big|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big)\big] \tag{46}\\
&\leq \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} 2n \cdot \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big]\big] \tag{47}\\
&= 2n \cdot \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big]\big] \tag{48}\\
&\leq 2n \cdot \mathbb{E}\big[2/\lambda n\big] \leq 4/\lambda. \notag
\end{align}$$

Equality (43) follows by chain-rule of KL Divergence (see Fact 3.3). Equality (44) follows by the fact that the conditional distribution of messages not sent by P is not affected by conditioning on C, and we can drop the conditioning on S since it is determined by $\mathrm{msg}_{<i}$. Inequality (45) follows by convexity of KL-Divergence (Fact 3.2) ($\mathrm{Msg}^{\mathsf{A}}_i$ is a convex combination of $\mathrm{Msg}^{\mathsf{A}}_i\,\big|_{C}$ and $\mathrm{Msg}^{\mathsf{A}}_i\,\big|_{\neg C}$). Inequality (46) follows by construction (see Step 4b of Algorithm 4.9) and the convexity of KL-Divergence—the altered messages are a convex combination of honest messages and biased messages (caused by Step 4(b)ii of Algorithm 4.9). Inequality (47) follows from Lemma 4.8(2), and the penultimate inequality holds since, by assumption, the protocol Π is n-normal.

By Equation (42) and the Pinsker bound (see Fact 3.4), it holds that $\mathsf{SD}\big(\mathrm{Msg}^{\mathsf{A}}\,\big|_{S \wedge C}, \mathrm{Msg}^{\mathsf{A}}\,\big|_{S}\big) \leq 2/\sqrt{\lambda}$. Consequently, by Equation (41) and the data-processing inequality of statistical distance

(Fact 3.1), it holds that (for sufficiently large n)

$$\mathbb{P}\big[H \wedge L \,|\, S \wedge C\big] \geq \mathbb{P}\big[H \wedge L \,|\, S\big] - {}^{2}\!/\!\sqrt{\lambda} = {}^{1}\!/\!{}_{16} - {}^{2}\!/\!\sqrt{\lambda} > {}^{1}\!/\!{}_{32} \tag{49}$$

In other words, even when P is a corrupted small-jumps party, it still encounters large conditional variance and biases all jumps it encounters. Therefore, all that is left to do is analyze the expectancy of P's increments under this conditioning. For $\mathrm{msg} \in \mathrm{Supp}\big(\mathrm{Msg}^{\mathsf{A}}\big)$ let $\mathbb{1}_H(\mathrm{msg})$ be *the value* of $\mathbb{1}_H$ as determined by $\mathrm{Msg}^{\mathsf{A}} = \mathrm{msg}$. Compute,

$$\begin{aligned}
\mathbb{E}\big[\textstyle\sum_{i \in \mathcal{I}_{\mathsf{P}}} X_i \,|\, S \wedge C\big] &= \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} \mathbb{E}\big[X_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i} \wedge C\big]\big] \\
&\geq \mathbb{E}_{\mathrm{msg} \leftarrow \widetilde{\mathrm{Msg}}}\big[\mathbb{1}_H(\mathrm{msg}) \cdot \textstyle\sum_{i \in \mathrm{SentBy}_{\mathsf{P}}(\mathrm{msg})} \mathbb{E}\big[\mathrm{Biased}_{\sqrt{n}}\big(Y_i|_{\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}}\big)\big]\big] &\quad (50) \\
&= \mathbb{E}\big[\mathbb{1}_H \cdot \textstyle\sum_{i \in \mathcal{I}_{\mathsf{P}}} \sqrt{n} \cdot \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big] \,|\, S \wedge C\big] &\quad (51) \\
&\geq \mathbb{E}\big[\mathbb{1}_H \cdot \sqrt{n} \cdot \mathbb{1}_L \cdot {}^{1}\!/\!{}_{8\lambda n} \,|\, S \wedge C\big] &\quad (52) \\
&= {}^{1}\!/\!{}_{8\lambda\sqrt{n}} \cdot \mathbb{P}\big[H \wedge L \,|\, S \wedge C\big] \geq {}^{1}\!/\!{}_{8\lambda\sqrt{n}} \cdot {}^{1}\!/\!{}_{32} = {}^{1}\!/\!{}_{256\lambda\sqrt{n}}.
\end{aligned}$$

Inequality (50) follows by the definition of A (see Step 4b of Algorithm 4.9) and Claim 4.11 (Y_i is independent of C conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i}$). Equality (51) follows from Lemma 4.8(1). Inequality (52) follows by a point-wise inequality, and the last inequality follows by Equation (49). □

4.1.5 Bounding KL-Divergence between Attacked and Honest Executions, Proving Claim 4.13

Claim 4.21 (Restatement of Claim 4.13)**.** *$D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}} \,\|\, \mathrm{Msg}^{\Pi}\big) \leq 16^3\lambda^3$.*

The core of the proof relies on Lemma 4.8(3) that states that corrupting some party with probability p and then biasing its message according to $\mathrm{Biased}^{f}_{\alpha}$, is equivalent to biasing this message according to $\mathrm{Biased}^{f}_{p\alpha}$. This fact yields the following observation:

Claim 4.22. *For any $i \in [\ell]$ and $\mathrm{msg}_{<i} \in \mathrm{Supp}(\mathrm{Msg}^{\mathsf{A}}_{<i})$, it holds that*

$$D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i|_{\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}} \,\|\, \mathrm{Msg}^{\Pi}_i|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}}\big) \leq 16^3\lambda^4 \cdot \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big].$$

Proof. Let $\mathsf{P} := \mathsf{NxtParty}(\mathrm{msg}_{<i})$ (i.e., be the party sending the i^{th} message). If P is $\mathsf{NonRbst}$, we are done since its messages are unchanged (never corrupted). Otherwise, we separately deal with the case that P is a small-jumps party and a large-jump party (as determined by the partial transcript $\mathrm{msg}_{<i}$).

P is a small-jumps party. Conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}$, the i^{th} message is altered from its honest (conditional) distribution according to Π, with probability $p \leq {}^{16\lambda^2}\!/\!\sqrt{n}$ (see Step 4b of Algorithm 4.9). If the i^{th} message is altered, it is sampled according to $\mathrm{Biased}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i},\,\cdot)}_{\sqrt{n}}\big(\mathrm{Msg}^{\Pi}_i|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}}\big)$. By Lemma 4.8(3), $\mathrm{Msg}^{\mathsf{A}}_i|_{\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}}$ is distributed like $\mathrm{Biased}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i},\,\cdot)}_{p\sqrt{n}}\big(\mathrm{Msg}^{\Pi}_i|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}}\big)$. Hence, by Lemma 4.8(2)

$$\begin{aligned}
&D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i|_{\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}} \,\|\, \mathrm{Msg}^{\Pi}_i|_{\mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}}\big) \\
&\leq 2 \cdot \big(p\sqrt{n}\big)^2 \cdot \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big] \\
&\leq 2 \cdot 16^2\lambda^4 \cdot \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\leq i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big].
\end{aligned}$$

P is a large-jump party. Conditioned on $\mathrm{Msg}^{\mathsf{A}}_{<i} = \mathrm{msg}_{<i}$, the i^{th} message is altered from its honest (conditional) distribution according to Π, with probability $\lambda^2\sqrt{v}$ where $v := \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\le i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big]$. If the i^{th} message is altered, it is sampled according to $\mathrm{Biased}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i},\,\cdot)}_{1/\sqrt{v}}\big(\mathrm{Msg}^{\Pi}_i\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big)$. Hence, by Lemma 4.8(3), $\mathrm{Msg}^{\mathsf{A}}_i\,|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i}}$ is distributed like $\mathrm{Biased}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i},\,\cdot)}_{\lambda^2}\big(\mathrm{Msg}^{\Pi}_i\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big)$. By Lemma 4.8(2), we conclude that

$$D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i}} \;\|\; \mathrm{Msg}^{\Pi}_i\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big) \le 2\cdot\lambda^4\cdot\mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\le i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big].$$

□

Proof of Claim 4.13. Let the set RobustJumps(msg) denote the value of RobustJumps determined by $\mathrm{Msg}^{\mathsf{A}} = \mathrm{msg}$. Compute,

$$\begin{align}
&D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_{\le \ell} \;\|\; \mathrm{Msg}^{\Pi}_{\le \ell}\big) \notag\\
&= \sum_{i=1}^{\ell} \mathbb{E}_{\mathrm{msg}\leftarrow \mathrm{Msg}^{\mathsf{A}}}\big[D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i}} \;\|\; \mathrm{Msg}^{\Pi}_i\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big)\big] \tag{53}\\
&= \mathbb{E}_{\mathrm{msg}\leftarrow \mathrm{Msg}^{\mathsf{A}}}\big[\textstyle\sum_{i\in \mathrm{RobustJumps(msg)}} D_{\mathrm{KL}}\big(\mathrm{Msg}^{\mathsf{A}}_i\,|_{\mathrm{Msg}^{\mathsf{A}}_{<i}=\mathrm{msg}_{<i}} \;\|\; \mathrm{Msg}^{\Pi}_i\,|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}\big)\big] \tag{54}\\
&\le \mathbb{E}_{\mathrm{msg}\leftarrow \mathrm{Msg}^{\mathsf{A}}}\big[\textstyle\sum_{i\in \mathrm{RobustJumps(msg)}} 16^3\lambda^4\cdot \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\le i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\big]\big] \tag{55}\\
&= 16^3\lambda^4\cdot \mathbb{E}\big[\textstyle\sum_{i\in \mathrm{RobustJumps}} \mathrm{Var}\big[\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\le i}\big) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \notag\\
&= 16^3\lambda^4\cdot \mathbb{E}\big[\textstyle\sum_{i\in \mathrm{RobustJumps}} \mathrm{Var}\big[Y_i \mid \mathrm{Msg}^{\mathsf{A}}_{<i}\big]\big] \tag{56}\\
&\le 16^3\lambda^3. \tag{57}
\end{align}$$

Equality (53) follows by chain rule of KL Divergence (see Fact 3.3). Equality (54) follows since non-RobustJumps are not corrupted. Inequality (55) follows by Claim 4.22. Inequality (56) follows by definition of Y_i (i.e., its conditional distribution is $\mathrm{jump}^{\Pi}\big(\mathrm{Msg}^{\Pi}_{\le i}\big)$). And finally, Inequality (57) follows by Claim 4.12. □

4.2 Biasing Robust Coin Flip

In this section, we use the attack on normal robust protocols proved to exist in Section 4.1, for attacking *arbitrary* robust protocols. We do that by transforming an arbitrary robust protocol into a related normal coin-flipping protocol and proving that the attack on the latter normal protocol stated in Theorem 4.6, yields an attack of essentially the same quality on the original (non-normal) protocol, thus proving Theorem 4.3.

We start by defining the normal form variant of a coin-flipping protocol. Let Π be an n-party, ℓ-round, full-information coin-flipping protocol. Letting $t = 2\ell n + 1$, the t-party, ℓ-round, n-normal variant of Π, is defined as follows:

Protocol 4.23 (n-normal protocol $\widetilde{\Pi}$)**.**

1. *For each party* P *of the protocol* Π*, the protocol* $\widetilde{\Pi}$ *has* 2ℓ *parties* $\mathsf{P}^{\mathsf{small}}_1, \ldots, \mathsf{P}^{\mathsf{small}}_{\ell}$ *and* $\mathsf{P}^{\mathsf{large}}_1, \ldots, \mathsf{P}^{\mathsf{large}}_{\ell}$*. In addition,* $\widetilde{\Pi}$ *has a special party named* $\mathsf{NonRbst}$*.*

2. *For each party* P *of* Π*, start three counters* $L_{\mathsf{P}} = S_{\mathsf{P}} = 1$*, and* $A_{\mathsf{P}} = 0$*.*

3. In rounds $i = 1$ to ℓ, the protocol is defined as follows.

(a) Let $\mathrm{msg}_{<i}$ denote the messages sent in the previous rounds, and let P be the party that would have sent the i^{th} message in Π given this transcript.

(b) Let Q_i be the distribution $\mathrm{Msg}_i^{\Pi}\big|_{\mathrm{Msg}^{\Pi}_{<i}=\mathrm{msg}_{<i}}$ and let $v_i := \mathrm{Var}\big[\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, Q_i)\big]$.

(c) Set P' (the "active" party) as follows:

```
i. If $\min\big(\mathrm{Supp}\big(\mathrm{jump}^{\Pi}\big(\mathrm{msg}_{<i,}\, Q_i\big)\big)\big) \le -1/\lambda_n\sqrt{n}$, set $\mathsf{P}'$ to $\mathsf{NonRbst}$.
ii. Else, If $v_i \ge 1/\lambda_n n$, set $\mathsf{P}'$ to $\mathsf{P}^{\mathsf{large}}_{L_{\mathsf{P}}}$, and update $L_{\mathsf{P}} = L_{\mathsf{P}} + 1$.
iii. Else, If $v_i < 1/\lambda_n n$:
        Set $\mathsf{P}'$ to $\mathsf{P}^{\mathsf{small}}_{S_{\mathsf{P}}}$ and update $A_{\mathsf{P}} = A_{\mathsf{P}} + v_i$
        If $A_{\mathsf{P}} > 1/\lambda_n n$:
            • Set $S_{\mathsf{P}} = S_{\mathsf{P}} + 1$.
            • Set $A_{\mathsf{P}} = 0$.
```

(d) P' sends the i^{th} message, as P would in Π given the partial transcript $\mathrm{msg}_{<i}$.

Claim 4.24. *Assume Π is a n-party full-information coin-flipping protocol, then $\widetilde{\Pi}$ is a n-normal full-information coin-flipping protocol.*

Proof. We handle each of the conditions independently,

Single non-robust party: Step 3(c)i properly handles jumps that should belong to $\mathsf{NonRbst}$.

Large-jump party sends a single message: Clearly Step 3(c)ii associates parties of the form $\mathsf{P}^{\mathsf{large}}_k$ with at most one jump, and it is clear that only parties of this form might have large jumps.

Small-jumps party has bounded overall variance: Step 3(c)iii assures that once $A_{\mathsf{P}} > 1/\lambda_n n$, namely the active party has a sum of conditional variances which is larger than $1/\lambda_n n$, it is never associated with another jump further along the execution. Thus, since A_{P} increases by at most $1/\lambda_n n$ at a time, it never surpasses $2 \cdot 1/\lambda_n n$.

At most n unfulfilled parties: Note that parties which have a sum of conditional variances which is at most $1/\lambda_n n$ must be parties of the form $\mathsf{P}^{\mathsf{small}}_k$, and the only parties of this form that participate in the protocol (namely, unfulfilled parties) are $\mathsf{P}^{\mathsf{small}}_{S^f_{\mathsf{P}}}$ where P is some party (at most n) and S^f_{P} is the final value of S_{P}. Therefore, at most n unfulfilled parties exist for any transcript.

□

Proving Theorem 4.3. Given the above tool and Theorem 4.6, the proof of Theorem 4.3 is immediate.

Proof of Theorem 4.3. Let $\widetilde{\Pi}$ be the n-normal variant of Π defined by Protocol 4.23. By Theorem 4.6, there exists a $O(\sqrt{n} \cdot \log n)$-adaptive adversary $\widetilde{\mathsf{A}}$ for $\widetilde{\Pi}$ such that $\mathbb{E}\big[\widetilde{\Pi}_{\widetilde{\mathsf{A}}}\big] \ge 1 - \varepsilon_n$. Consider the adversary A on Π that emulates $\widetilde{\mathsf{A}}$ while transforming corruptions of the parties of $\widetilde{\Pi}$ to parties of Π according to the mapping implicitly defined in Protocol 4.23. It is clear that

$\mathbb{E}\left[\Pi_{\mathsf{A}}\right] = \mathbb{E}\left[\widetilde{\Pi}_{\widetilde{\mathsf{A}}}\right] \geq 1-\varepsilon_n$. In addition, corrupting k parties in $\widetilde{\Pi}$ is translated to corrupting at most k parties of Π, since by construction the parties in $\widetilde{\Pi}$ are refinements of the parties in Π. We conclude that A is the desired $O(\sqrt{n}\cdot\log n)$-adaptive □

4.3 Proving Lemma 4.8

In this section, we prove Lemma 4.8.

Lemma 4.25 (Restatement of Lemma 4.8)**.** *For any P, α and f as in Definition 4.7, it holds that*

1. $\mathbb{E}\left[f\left(\mathrm{Biased}^f_\alpha(P)\right)\right] = \alpha\cdot\mathrm{Var}\left[f(P)\right]$.
2. $D_{\mathrm{KL}}\left(\mathrm{Biased}^f_\alpha(P)\parallel P\right) \leq 2\alpha^2\cdot\mathrm{Var}\left[f(P)\right]$.
3. $\left(p\cdot\mathrm{Biased}^f_\alpha(P) + (1-p)\cdot P\right) \equiv \mathrm{Biased}^f_{p\cdot\alpha}(P)$ *for any* $p\in[0,1]$.
4. *There exist a distribution* (A,B) *which couples* P *and* $\mathrm{Biased}^f_\alpha(P)$*, i.e.,* $A\equiv P$ *and* $B\equiv \mathrm{Biased}^f_\alpha(P)$*, such that for any* $(a,b)\leftarrow(A,B)$ *it holds that* $f(B)\geq f(A)$*.*

Proof of Lemma 4.8.

Item 1:

$$\begin{aligned}
\mathbb{E}\left[f\left(\mathrm{Biased}^f_\alpha(P)\right)\right] &= \sum_{e\in\mathrm{Supp}(P)} f(e)\cdot\mathbb{P}\left[\mathrm{Biased}^f_\alpha(P)=e\right]\\
&= \sum_{e\in\mathrm{Supp}(P)} f(e)\cdot\mathbb{P}\left[P=e\right]\cdot(1+\alpha f(e))\\
&= \mathbb{E}\left[f(P)\cdot(1+\alpha f(P))\right] = \mathbb{E}\left[f(P)\right]+\alpha\cdot\mathbb{E}\left[f^2(P)\right] = \alpha\cdot\mathrm{Var}\left[f(P)\right].
\end{aligned}$$

Item 2:

$$\begin{aligned}
D_{\mathrm{KL}}\left(\mathrm{Biased}^f_\alpha(P)\parallel P\right) &= \sum_{e\in\mathrm{Supp}(P)} \mathbb{P}\left[\mathrm{Biased}^f_\alpha(P)=e\right]\cdot\log\left(\frac{\mathbb{P}\left[\mathrm{Biased}^f_\alpha(P)=e\right]}{\mathbb{P}\left[P=e\right]}\right)\\
&= \sum_{e\in\mathrm{Supp}(P)} \mathbb{P}\left[P=e\right]\cdot(1+\alpha f(e))\cdot\log(1+\alpha f(e))\\
&= \mathbb{E}\left[(1+\alpha f(P))\cdot\log(1+\alpha f(P))\right] = \mathbb{E}\left[\log(1+\alpha f(P))\right]+\mathbb{E}\left[\alpha f(P)\cdot\log(1+\alpha f(P))\right]\\
&\leq \log\left(1+\mathbb{E}\left[\alpha f(P)\right]\right)+\mathbb{E}\left[2\alpha^2 f^2(P)\right] = 2\alpha^2\cdot\mathrm{Var}\left[f(P)\right].
\end{aligned}$$

The last inequality follows by Jensen's inequality and Fact 3.24.

Item 3:

$$\begin{aligned}
&\mathbb{P}\left[\left(p\cdot\mathrm{Biased}^f_\alpha(P)+(1-p)\cdot P\right)=e\right]\\
&= p\cdot\mathbb{P}\left[\mathrm{Biased}^f_\alpha(P)=e\right]+(1-p)\cdot\mathbb{P}\left[P=e\right]\\
&= p\cdot\mathbb{P}\left[P=e\right]\cdot(1+\alpha f(P))+(1-p)\cdot\mathbb{P}\left[P=e\right]\\
&= \mathbb{P}\left[P=e\right]\cdot(1+p\alpha f(P)).
\end{aligned}$$

Item 4: Consider the following random process: Sample $a \leftarrow P$. If $f(a) \geq 0$, set $b = a$. If $f(a) < 0$ with probability $1 + \alpha f(a)$ set $b = a$, otherwise sample $b \leftarrow P_f^+$ for

$$P_f^+ \equiv \begin{cases} e \text{ with probability } \frac{\mathbb{P}\left[P=e\right]\cdot f(e)}{\mathbb{E}\left[|f(P)|\right]} & \text{for } e \in \operatorname{Supp}(P) \text{ with } f(e) > 0 \end{cases}$$

By construction $f(b) \geq f(a)$. In addition, it is not hard to verify that the marginal distributions of a and b are that of P and $\operatorname{Biased}_\alpha^f\left(P\right)$, respectively.

□

5 Biasing Arbitrary Coin Flip

In this section, we use the attack on robust protocols, described in Section 4, to prove our main result: an adaptive attack on any full-information coin-flipping protocols. The main result of our paper is given below. Recalling our notations,

Notation 5.1 (Restatement of Notation 4.1)**.** *For $n \in \mathbb{N}$, let $\varepsilon_n := {}^1\!/\!\sqrt[50]{\log\log n}$, $\lambda_n := {}^{100}\!/\varepsilon_n^5 = 100 \cdot \sqrt[10]{\log\log n}$ and $\delta_n := {}^1\!/\!\log^2 n$.*

Theorem 5.2 (Biasing full-information coin flips)**.** *For any n-party, full-information coin-flipping protocol Π, there exists a $O\left(\sqrt{n}\cdot\log^3 n\right)$-adaptive adversary A, such that $\mathbb{E}\left[\Pi_{\mathsf{A}}\right] \leq \varepsilon_n$ or $\mathbb{E}\left[\Pi_{\mathsf{A}}\right] \geq 1 - \varepsilon_n$.*

Our proof makes use of the following deterministic one-shot (modifies at most a single message) adversary attacking an n-party full-information coin-flipping protocol Γ. The adversary takes advantage of large negative jumps in order to bias the Γ's output towards 0,

Algorithm 5.3 (One-shot adaptive adversary B on Γ)**.**

For $i = 1$ **to** $\operatorname{NumMsgs}(\Gamma)$*:*

1. *Let $\operatorname{msg}_{<i}$ be the messages sent in the previous rounds, and let P be the party about to send the i^{th} message.*
2. *Denote $\mathcal{M}_i := \operatorname{Supp}\left(\operatorname{Msg}_{\leq i}^{\Gamma} \mid \operatorname{Msg}_{<i}^{\Gamma} = \operatorname{msg}_{<i}\right)$,*

   If *no message was corrupted before and $\exists m \in \mathcal{M}_i$: $\operatorname{jump}^{\Gamma}(\operatorname{msg}_{<i}, m) \leq -{}^1\!/\!\lambda_n\sqrt{n}$, corrupt and instruct P to broadcast such a message m as it next message.*

The proof of the following fact is immediate.

Claim 5.4. *Let Γ be an n-party full-information coin-flipping protocol. Then:*

$$\mathbb{E}\left[\Gamma_{\mathsf{B}}\right] \geq \mathbb{E}\left[\Gamma\right] + {}^1\!/\!\lambda_n\sqrt{n} \cdot \mathbb{P}\left[\exists i\colon \min\left(\operatorname{Supp}\left(\operatorname{jump}^{\Gamma}\left(\operatorname{Msg}_{\leq i}^{\Gamma}\right) \mid \operatorname{Msg}_{<i}^{\Gamma}\right)\right) \leq -{}^1\!/\!\lambda_n\sqrt{n}\right].$$

Equipped with the above tool and the attack presented in Section 4, we are ready to prove our main result.

Proof of Theorem 5.2. Denote $t = \sqrt{n}/\lambda\delta = O\big(\sqrt{n} \cdot \log^3 n\big)$, consider the protocols $\Pi^0, \ldots, \Pi^t$ recursively defined by $\Pi^0 := \Pi$ and $\Pi^{i+1} := {\Pi^i}_{\mathsf{B}}$. If $\mathbb{E}\big[\Pi^t\big] < \varepsilon_n$, then by Proposition 3.20 there exists a t-adaptive adversary that biases Π's output to less than ε_n (the composition of all intermediate adversaries), and we are done. Else, by Claim 5.4 there exists $k \in [t]$ such that for $\Psi = \Pi^k$ it holds that

$$\mathbb{P}\big[\exists i\colon \min\big(\mathrm{Supp}\big(\mathrm{jump}^{\Psi}\big(\mathrm{Msg}^{\Psi}_{\leq i}\big) \mid \mathrm{Msg}^{\Psi}_{<i}\big)\big) \leq -1/\lambda_n\sqrt{n}\big] \leq \delta.$$

Hence, by Theorem 4.3, there exists an $O(\sqrt{n} \cdot \log n)$-adaptive adversary A such that

$$\mathbb{E}\big[\Psi_{\mathsf{A}}\big] \geq 1 - \varepsilon_n.$$

Denote by C the t-adaptive adversary according to Definition 3.18 (the composition of all intermediate adversaries) such that $\Pi_{\mathsf{C}} \equiv \Pi^t$. Let $\mathsf{A} \circ \mathsf{C}$ be the $O\big(\sqrt{n} \cdot \log^3 n\big)$-adaptive adversary according to Definition 3.18, by Proposition 3.20 it holds that $\mathbb{E}\big[\Pi_{\mathsf{A}\circ\mathsf{C}}\big] = \mathbb{E}\big[\Psi_{\mathsf{A}}\big] \geq 1 - \varepsilon_n$, concluding the proof. □

6 Strongly Adaptive, Bidirectional Adversaries

In this section, we use strongly adaptive adversaries to make the attacker described in the previous sections bidirectional (able to bias the protocol's outcome to both zero and one). Formally (using the notations of Section 5), we prove the following result.

Notation 6.1 (Restatement of Notation 4.1)**.** *For $n \in \mathbb{N}$, let $\varepsilon_n := 1/\sqrt[50]{\log\log n}$, $\lambda_n := 100/\varepsilon_n^5 = 100 \cdot \sqrt[10]{\log\log n}$ and $\delta_n := 1/\log^2 n$.*

Theorem 6.2 (Forcing full-information coin-flipping protocols)**.** *For any n-party, full-information coin-flipping protocol Π such that $\mathbb{E}\big[\Pi\big] \geq \varepsilon_n$, there exists a $O\big(\sqrt{n} \cdot \log^3 n\big)$-strongly adaptive adversary A, such that $\mathbb{E}\big[\Pi_{\mathsf{A}}\big] \geq 1 - \varepsilon_n$.*

Note that this is indeed a bidirectional attack capable of biasing protocols in which both values are significant enough. If one wants to bias a protocol towards 0, simply apply the attack on the flipped protocol (exactly the same protocol, besides the output function, which returns the opposite from the original output function).

The attack. The attack follows the same lines as the one described in Section 5, except in the immunization phase (Algorithm 5.3) that turns the protocol to be robust. Using strongly adaptive corruptions, the adversary can always immunize the protocol so that the (non-strong) adaptive attack described in Section 4 is applicable. The strongly adaptive immunization deals with *non-robust jumps* much better; instead of preparing for the worst-case scenario and corrupting every such jump, the attacker deals with them only if the unfavorable outcome is taken.

Algorithm 6.3 (One-shot strongly adaptive adversary B on Γ)**.**

```
For $i = 1$ to NumMsgs($\Gamma$):
    Let $\mathrm{msg}_{\leq i}$ be the messages sent in the previous (and current) rounds, and let $\mathsf{P}$ be the party
    that sent the $i^{\text{th}}$ message. Let $\mathcal{M}_i := \mathrm{Supp}\big(\mathrm{Msg}^{\Gamma}_{\leq i} \mid \mathrm{Msg}^{\Gamma}_{<i} = \mathrm{msg}_{<i}\big)$.
```

If no message was corrupted before and $\mathrm{jump}^{\Gamma}\left(\mathrm{msg}_{\leq i}\right) \leq -1/\lambda_n\sqrt{n}$*, corrupt* P *and instruct it send a message* $m \in \mathcal{M}_i$ *with* $\mathrm{jump}^{\Gamma}(\mathrm{msg}_{<i}, m) \geq 0$.

Similarly to Section 5, consider the following (immediate) claim.

Claim 6.4. *Let* Γ *be an* n*-party, full-information coin-flipping protocol* Γ*. Then*

$$\mathbb{E}\left[\Gamma_{\mathsf{B}}\right] \geq \mathbb{E}\left[\Gamma\right] + \mathbb{E}\left[\#\mathrm{Corruptions}\right] \cdot 1/\lambda_n\sqrt{n}.$$

Proof. Immediate. □

Similarly to the immunization phase presented in Section 5, we iteratively apply B until

$$\mathbb{P}\left[\exists i\colon \min\left(\mathrm{Supp}\left(\mathrm{jump}^{\Gamma}\left(\mathrm{Msg}^{\Gamma}_{\leq i}\right) \mid \mathrm{Msg}^{\Gamma}_{<i}\right)\right) \leq -1/\lambda_n\sqrt{n}\right] < \delta_n/2 \tag{58}$$

Denote the number of such applications by t. Let $\Pi^0 := \Pi$ and $\Pi^{i+1} := \Pi^i{}_{\mathsf{B}}$. By Claim 6.4, it holds that $\mathbb{E}\left[\Pi^{i+1}\right] \geq \mathbb{E}\left[\Pi^i\right] + 1/\lambda_n\sqrt{n} \cdot \mathbb{E}\left[\#\mathrm{Corruptions\ in\ }\Pi^i{}_{\mathsf{B}}\right]$. Reorganizing the terms, $\mathbb{E}\left[\#\mathrm{Corruptions\ in\ }\Pi^i{}_{\mathsf{B}}\right] \leq (\mathbb{E}\left[\Pi^{i+1}\right] - \mathbb{E}\left[\Pi^i\right]) \cdot \lambda_n\sqrt{n}$. In total,

$$\sum_{i=0}^{t-1} \mathbb{E}\left[\#\mathrm{Corruptions\ in\ }\Pi^i{}_{\mathsf{B}}\right] = \lambda_n\sqrt{n} \cdot \sum_{i=0}^{t-1}(\mathbb{E}\left[\Pi^i\right] - \mathbb{E}\left[\Pi^{i-1}\right]) = \lambda_n\sqrt{n} \cdot (\mathbb{E}\left[\Pi^t\right] - \mathbb{E}\left[\Pi^0\right]) \leq \lambda_n\sqrt{n} \tag{59}$$

Let C be the composition of all intermediate adversaries, according to Definition 3.21, such that $\Pi_{\mathsf{C}} \equiv \Pi^t$. By the above inequality, the expected amount of strongly adaptive corruptions C performs is at most $\lambda_n\sqrt{n}$. Let C' be the variant of C that aborts once it reaches $2\lambda_n\sqrt{n}/\delta_n$ corruptions, and denote $\Psi = \Pi_{\mathsf{C}'}$. By Markov's inequality, C' aborts with probability at most $\delta_n/2$. Combined with our stopping condition (i.e., Equation (58)) for applying B, it follows that that

$$\mathbb{P}\left[\exists i\colon \min\left(\mathrm{Supp}\left(\mathrm{jump}^{\Psi}\left(\mathrm{Msg}^{\Psi}_{\leq i}\right) \mid \mathrm{Msg}^{\Psi}_{<i}\right)\right) \leq -1/\lambda_n\sqrt{n}\right] \leq \delta_n.$$

In addition, $\mathbb{E}\left[\Psi\right] \geq \mathbb{E}\left[\Pi\right] \geq \varepsilon_n$. Hence, by Theorem 4.3, there exists an $O(\sqrt{n} \cdot \log n)$-strongly adaptive adversary A such that

$$\mathbb{E}\left[\Psi_{\mathsf{A}}\right] \geq 1 - \varepsilon_n.$$

Consider the attacker $\mathsf{A} \circ \mathsf{C}'$, the composed $O\left(\sqrt{n} \cdot \log^3 n\right)$-strongly-adaptive adversary. By Proposition 3.23, $\mathbb{E}\left[\Pi_{\mathsf{A}\circ\mathsf{C}'}\right] = \mathbb{E}\left[\Psi_{\mathsf{A}}\right] \geq 1 - \varepsilon_n$, concluding the proof.

Acknowledgment

We are grateful to Raz Landau, Nikolaos Makriyannis, Eran Omri, and Eliad Tsfadia for very helpful discussions. The first author is thankful to Michael Ben-Or for encouraging him to tackle this beautiful question.

References

[1] M. Ajtai and N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[2] N. Alon and M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM Journal on Computing*, 22(2):403–417, 1993.

[3] A. Beimel, I. Haitner, N. Makriyannis, and E. Omri. Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. In *Proceedings of the 59th Annual Symposium on Foundations of Computer Science (FOCS).*, 2018.

[4] M. Ben-Or and N. Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 408–416, 1985.

[5] I. Berman, I. Haitner, and A. Tentes. Coin flipping of any constant bias implies one-way functions. *Journal of the ACM*, 65(3):14, 2018.

[6] I. Berman, I. Haitner, and E. Tsfadia. A tight parallel-repetition theorem for random-terminating interactive arguments. In *Annual International Cryptology Conference (CRYPTO)*, 2020.

[7] M. Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1983.

[8] R. B. Boppana and B. O. Narayanan. Perfect-information leader election with optimal resilience. *SIAM Journal on Computing*, 29(4):1304–1320, 2000.

[9] Y. Dodis. New imperfect random source with applications to coin-flipping. In *Automata, Languages and Programming, 24th International Colloquium (ICALP)*, pages 297–309, 2001.

[10] Y. Dodis. Fault-tolerant leader election and collective coin-flipping in the full information model. `https://cs.nyu.edu/~dodis/ps/cf-survey.pdf`, 2006.

[11] O. Etesami, S. Mahloujifar, and M. Mahmoody. Computational concentration of measure: Optimal bounds, reductions, and more. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–363, 2020.

[12] O. Etesami, S. Mahloujifar, and M. Mahmoody. Computational concentration of measure: Optimal bounds, reductions, and more. In *Symposium on Discrete Algorithms (SODA)*, pages 345–363, 2020.

[13] A. A. Fedotov, P. Harremoes, and F. Topsoe. Refinements of pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.

[14] S. Goldwasser, Y. Tauman Kalai, and S. Park. Adaptively secure coin-flipping, revisited. In *Automata, Languages and Programming, 24th International Colloquium (ICALP)*, pages 663–674, 2015.

[15] I. Haitner. A parallel repetition theorem for any interactive argument. *SIAM Journal on Computing*, 42(6):2487–2501, 2013.

[16] I. Haitner and E. Omri. Coin Flipping with Constant Bias Implies One-Way Functions. *SIAM Journal on Computing*, pages 389—409, 2014. Preliminary version in *FOCS'11*.

[17] J. Håstad, R. Pass, D. Wikström, and K. Pietrzak. An efficient parallel repetition theorem. In *Theory of Cryptography (TCC)*, pages 1–18, 2010.

[18] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80, 1988.

[19] D. Lichtenstein, N. Linial, and M. Saks. Some extremal problems arising from discrete control processes. *Combinatorica*, 9(3):269–287, 1989.

[20] S. Mahloujifar and M. Mahmoody. Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In *Theory of Cryptography (TCC)*, pages 245–279, 2017.

[21] S. Mahloujifar and M. Mahmoody. Can adversarially robust learning leveragecomputational hardness? In *Algorithmic Learning Theory*, pages 581–609, 2019.

[22] S. Mahloujifar, M. Mahmoody, and A. Mohammed. Multi-party poisoning through generalized p-tampering. Technical Report 1809.03474, arXiv, 2018.

[23] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *AAAI Conference on Artificial Intelligence*, pages 4536–4543, 2019.

[24] H. K. Maji, M. Prabhakaran, and A. Sahai. On the Computational Complexity of Coin Flipping. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 613–622, 2010.

[25] A. Russell, M. Saks, and D. Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. *SIAM Journal on Computing*, 31(6):1645–1662, 2002.

[26] M. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM Journal on Discrete Mathematics*, 2(2):240–244, 1989.

[27] Y. Tauman Kalai, I. Komargodski, and R. Raz. A lower bound for adaptively-secure collective coin-flipping protocols. In *International Symposium on Distributed Computing (DISC)*, 2018.