Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 71 (2020)

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

Iftach Haitner *† Yonatan Karidi-Heller *

September 2, 2020

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 for many-turn protocols was left completely 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.

*School of Computer Science, Tel Aviv University. Emails: `iftachh@cs.tau.ac.il`,`karidiheller@mail.tau.ac.il`.
Research supported by ERC starting grant 638121, and Israel Science Foundation grant 666/19.

†Member of the Check Point Institute for Information Security.

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 hides (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 the parties' messages serve as the process's randomness. Thus, proving lower bound on coin-flipping protocols induces limitations on the stability on such random processes in general (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 flip is well understood, and almost matching upper (protocols) and lower (attackers) bounds are known, see Section 1.2. For adaptive adversaries, which are the focus of this work, much less is understood, and there are significant gaps 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 (of any round complexity). The works of Lichtenstein, Linial, and Saks [19], Goldwasser, Tauman Kalai, and Park [13] 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 for many-turn protocols was left completely 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, enforces the outcome of the protocol to be 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., the protocol's expected outcome (in an all-honest execution) is not $1/2$. 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

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* (a message is 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 [13] 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 [10] presented an *efficient* and optimal 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.

1.2.2 Data-Poisoning Attacks

Consider a learning algorithm that tries to learn an hypothesis from a training set comprised of samples taken from different sources. The random process corresponding to this learning task can be naturally viewed as (some form of) a coin-flipping protocol. 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

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 $(1 - 1/n)^{n/2} \approx \sqrt{1/e}$.

with a small number of sources. For this application, however, the attacker would better have the ability to force a predetermined output (rather than forcing some output, as our attack achieves). Hence, the attack on coin-flipping protocol we apply should be *bi-directional* (have to 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.

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. [11] facilitated their strongly adaptive attack on single-turn coin-flipping protocols (see Section 1.2) to obtain a strongly-adaptive data-poisoning attack.

Previously all known adversaries dealt only with *single-turn* coin-flipping protocols, translating into data-poisoning attacks which tamper some small set of samples. With the tools we present (see Theorem 1.1) it is now possible to discuss data-poisoning attacks that corrupts a small amount of *sources* (rather than tampering samples without consideration of the sources they are sampled from).

Open Questions & Future Work

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 [10].

A second question, motivated by the data-poisoning attacks described in Section 1.2.2, is what security models enable $o(n)$-corruption *bi-directional* attacks (the attacker can bias the protocol output to both directions). In a work in progress, [15], we show that in the strongly-adaptive corruption model, a $O(\sqrt{n})$-corruption *bi-directional* attack exists against any protocol.

Paper Organization

A rather elaborated 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.

2 Our Technique

In this section we give a rather elaborated description of our adaptive attack, and its analysis, 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 lager than $1 - \varepsilon$.

Similarly to previous adaptive attacks on full-information coin-flipping protocols, our attack exploits the "jumps" in the protocol 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 $\text{Msg} = (\text{Msg}_1, \ldots, \text{Msg}_\ell)$ denote the protocol transcript (i.e., parties' messages) in a random all-honest execution of Π. For $\text{msg} \in \text{Supp}(\text{Msg})$, let $\Pi(\text{msg})$ denote the final outcome of the execution described by msg, and for $\text{msg}_{\leq i} \in \text{Supp}(\text{Msg}_{\leq i} := (\text{Msg}_1, \ldots, \text{Msg}_i))$ let $\Pi(\text{msg}_{\leq i}) := \mathbb{E}\left[\Pi(\text{Msg}) \mid \text{Msg}_{\leq i} = \text{msg}_{\leq i}\right]$ be the expected outcome given a partial transcript. We refer to $\Pi(\text{Msg}_{\leq i}) - \Pi(\text{Msg}_{<i})$ (i.e., the change in the expected outcome induced by the i^{th} message) as the i^{th} jump in the protocol execution. Our attack exploits these gaps 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 it 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 with a certain probability, and when corrupting, only moderately changes the message distribution. Being gentle allows the attack to bypass the main obstacles in attacking many-turn protocols (see details in Section 2.2). Furthermore, in some setting of interest the attack can be made efficient. Efficient attacks where not even known for the single-turn, arbitrary message length case (the recursive attack of [27] is inherently inefficient). Finally, the gentleness of the attack make analyzing it a rather simple (and pleasurable!) task; the transcript of the gently attacked execution is not "too different" from the all-honest (unattacked) execution of the protocol. Consequently, the analysis requires one only to get 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] For correctness, we focus on robust protocols with respect to $b = 0$. That is, we assume that (for simplicity, with probability one)

$$\Pi(\text{Msg}_{\leq i}) \geq \Pi(\text{Msg}_{<i}) - 1/\sqrt{n} \tag{1}$$

We start with the case of single-turn robust protocols, and extend it to the many-turn (robust) setting in Section 2.2. The extension to arbitrary (non-robust) protocols is described in Section 2.3.

2.1 Attacking Robust Single-Turn Coin Flip

When corrupting a *single* message of the attacked protocol, we replace the original (honest) message distribution with the following parameterized variant.

Definition 2.1 (Biased). *For a distribution P, a constant $\alpha \geq 0$ and a function $f : \text{Supp}(P) \mapsto [-1/\alpha, \infty)$ such that $\mathbb{E}\left[f(P)\right] = 0$, let $\text{Biased}_\alpha^f(P)$ be the distribution defined by*

$$\mathbb{P}\left[\text{Biased}_\alpha^f(P) = x\right] := \mathbb{P}\left[P = x\right] \cdot (1 + \alpha \cdot f(x))$$

That is, the distribution P is "nudged" towards larger values of f, i.e., increasing the probability of positive elements (causing $\mathbb{E}\left[f\left(\text{Biased}_t^f(P)\right)\right] \geq \mathbb{E}\left[f(P)\right]$), and the larger α is the larger the bias.

[3] Gentle attacks, in the general sense that the attacker does not try to maximize the effect of the attack in each round, but rather keeps the attacked execution similar to the all-honest one, were found useful 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, 14, 6].

[4] An almost accurate example for a (bi-directional) 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 $1/\sqrt{n}$.

The function f to be considered by our attack is the protocol's expected output given the current message. While it is tempting to choose t as large as possible, and thus maximize a single round effect on the final outcome, our attack takes a more subtle approach.

The attack. Our attacker on an n-party, single-turn (n-round), robust protocol Π is defined below. In the following let $\operatorname{jump}(\mathrm{msg}_{\leq i}) := \Pi(\mathrm{msg}_{\leq i}) - \Pi(\mathrm{msg}_{<i})$, i.e., the difference in expected outcome induced by the last (i^{th}) message in the partial transcript.

Algorithm 2.2 (Single-turn 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, let $Q_i := \mathrm{Msg}_i|_{\mathrm{Msg}_{<i}=\mathrm{msg}_{<i}}$, let $\mathrm{jump}_i := \operatorname{jump}(\mathrm{msg}_{<i}, \cdot)$ and let $v_i := \operatorname{Var}[\mathrm{jump}_i(Q_i)]$.*

2. *If $v_i \geq 1/n$, corrupt the i^{th} party with probability $1/\varepsilon^3 \cdot \sqrt{v_i}$. If corrupted, instruct it to send the next message according to $\mathrm{Biased}^{\mathrm{jump}_i}_{1/\sqrt{v_i}}(Q_i)$.*

   Else, corrupt the i^{th} party with probability $1/\varepsilon^3 \cdot 1/\sqrt{n}$. If corrupted, instruct it to send the next message according to $\mathrm{Biased}^{\mathrm{jump}_i}_{\sqrt{n}}(Q_i)$. [5]

That is, a message (party) is corrupted with probability proportional to the (conditional) standard deviation it induces on the expected outcome of Π, or $1/\sqrt{n}$ if it is smaller. 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 (up to $\sqrt{n}$).[6] 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 $O(\sqrt{n}/\varepsilon^3)$. Thus, 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 bias, and thus the final 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. Having little liveliness according to our notion almost directly implies the success of the attack, with no need for additional tail inequalities as used by some of the previous works.

Let $\widehat{\mathrm{Msg}} = (\widehat{\mathrm{Msg}}_1, \ldots, \widehat{\mathrm{Msg}}_n)$ be the message distribution induced by the above attack, and consider the *sub*-martingale $S = (S_0, \ldots, S_n)$ with respect to $\widehat{\mathrm{Msg}}$ defined as $S_i := \Pi(\widehat{\mathrm{Msg}}_{\leq i})$, i.e., the expected honest outcome (if all messages were sampled honestly from now on) at every step of the biased execution. By definition, $S_0 = \mathbb{E}[\Pi(\mathrm{Msg})] = 1/2$ and $S_n \in \{0, 1\}$.

For $i \in [n]$, let Q_i be the value of Q_i in the attacked execution, determined by $\widehat{\mathrm{Msg}}_{i-1}$, and let $Y_i := \operatorname{jump}(\widehat{\mathrm{Msg}}_{<i}, q)$ for $q \leftarrow Q_i$. Since Q_i is the i^{th} honest message distribution (determined by $\widehat{\mathrm{Msg}}_{<i}$), it holds that $\mathbb{E}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}] = 0$. Since the Y_i's are also independent of each other conditioned on $\widehat{\mathrm{Msg}}$, the sequence $Y = (Y_1, \ldots, Y_n)$ is a *martingale difference sequence with respect*

[5] Since we assume Equation (1), in both cases $\mathrm{Biased}(Q_i)$ is indeed a valid probability distribution.

[6] Assuming 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 $\approx 1/2 \cdot (1 + \sqrt{n} \cdot 1/\sqrt{n}) = 1$.

to $(\widehat{\mathrm{Msg}}_i, Y_i)$, where the corresponding martingale is (somewhat of) an honest execution, coupled with the biased execution. The core of our analysis lies in the following lemma.

Lemma 2.3. *$\mathbb{E}\big[\sum_{i=1}^n \mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}]\big] \leq \varepsilon^3$.*

The proof of Lemma 2.3 is sketched below, but first we use it for analyzing the quality of the attack. We first argue about the expected number of corruptions. By construction, the probability the attacker corrupts the i^{th} party is

$$\begin{aligned} {}^1\!/\!{}_{\varepsilon^3} \cdot \max\left\{\sqrt{\mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}]}, {}^1\!/\!{}_{\sqrt{n}}\right\} &\leq {}^1\!/\!{}_{\varepsilon^3} \cdot \max\left\{\sqrt{n} \cdot \mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}], {}^1\!/\!{}_{\sqrt{n}}\right\} \\ &\leq {}^1\!/\!{}_{\varepsilon^3} \cdot (\sqrt{n} \cdot \mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}] + {}^1\!/\!{}_{\sqrt{n}}). \end{aligned} \tag{2}$$

Hence by Lemma 2.3, the expected number of corruptions is bounded by

$${}^1\!/\!{}_{\varepsilon^3} \cdot \mathbb{E}\Big[\sum_{i=1}^n \Big(\sqrt{n} \cdot \mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}] + {}^1\!/\!{}_{\sqrt{n}}\Big)\Big] \leq \sqrt{n}(1 + {}^1\!/\!{}_{\varepsilon^3}).$$

We next argue about the bias induced by the attack. Since Y is a martingale difference sequence with respect to $(\widehat{\mathrm{Msg}}_i, Y_i)$, i.e., $\mathbb{E}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}, Y_{<i}\big] = 0$, it is easy to verify that

$$\mathbb{E}\Big[\Big(\sum_{i=1}^n Y_i\Big)^2\Big] = \sum_{i=1}^n \mathbb{E}\big[Y_i^2\big] = \mathbb{E}\Big[\sum_{i=1}^n \mathrm{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}]\Big] \tag{3}$$

Hence by Lemma 2.3 and Chebyshev's inequality, we deduce that

$$\mathbb{P}\Big[\big|\sum_{i=1}^n Y_i\big| \geq \varepsilon\Big] \leq \varepsilon \tag{4}$$

Furthermore, since S_i are the "biased towards one" variants of Q_i (and thus of Y_i), there exists a (rather) straightforward coupling between S and Y for which

$$S_i - S_{i-1} \geq Y_i \tag{5}$$

Since, by definition, $S_0 = 1/2$, it follows that $\mathbb{P}\big[S_n \leq 0\big] = \mathbb{P}\big[\sum_{i=1}^n Y_i \leq -{}^1\!/\!{}_2\big] \leq \varepsilon$, and since $S_n \in \{0,1\}$, we deduce that $\mathbb{P}\big[S_n = 1\big] \geq 1 - \varepsilon$. Namely, the output of the attacked protocol is 1 with probability at least $1-\varepsilon$. (The same argument works also if it is only guaranteed that $S_0 \geq \varepsilon$, as happens in Section 2.3)

Proving Lemma 2.3. We start with two simple observations. The first is that for any distribution P, constant $alpha \geq 0$ and function $f : \mathrm{Supp}(P) \mapsto [-{}^1\!/\!{}_\alpha, \infty)$ such that $\mathbb{E}\big[f(P)\big] = 0$, it holds that

$$\begin{aligned} \mathbb{E}\big[f\big(\mathrm{Biased}_\alpha^f(P)\big)\big] &= \sum_{x \in \mathrm{Supp}(P)} f(x) \cdot \mathbb{P}\big[\mathrm{Biased}_\alpha^f(P) = x\big] \\ &= \sum_{x \in \mathrm{Supp}(P)} f(x) \cdot \mathbb{P}\big[P = x\big] \cdot (1 + \alpha \cdot f(P)) \end{aligned} \tag{6}$$

$$= \mathbb{E}\big[f(P) \cdot (1 + \alpha \cdot f(P))\big] = \mathbb{E}\big[f\big] + \alpha \cdot \mathbb{E}\big[f^2(P)\big] = 0 + \alpha \cdot \operatorname{Var}\big[f(P)\big].$$

A second immediate observation is that for any $p \in [0,1]$:

$$\Big(p \cdot \mathrm{Biased}^f_\alpha\big(P\big) + (1-p) \cdot P\Big) \equiv \mathrm{Biased}^f_{p\cdot\alpha}\big(P\big) \tag{7}$$

Let V_i be the value of the variable v_i in the execution of the attack (determined by $\widehat{\mathrm{Msg}}_{<i}$), and let $V'_i := \max\{v_i, {}^1\!/\!{}_n\}$. For a partial transcript $\mathrm{msg}_{<i} \in \mathrm{Supp}(\widehat{\mathrm{Msg}}_{<i})$, applying the above observations with respect to $P := \mathrm{Msg}_i\big|_{\mathrm{Msg}_{<i}=\mathrm{msg}_{<i}}$, $p := {}^1\!/\!{}_{\varepsilon^3} \cdot \sqrt{V'_i\big|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i}}}$, $\alpha := 1/\sqrt{V'_i\big|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i}}}$ and $\mathrm{jump}_i := \mathrm{jump}(\mathrm{msg}_{<i}, \cdot)$, yields that

$$\begin{aligned}
\mathbb{E}\big[S_i - S_{i-1} \mid \widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}\big] &= \mathbb{E}\big[\mathrm{jump}_i\big(\mathrm{Biased}^{\mathrm{jump}_i}_{1/\varepsilon^3}\big(\mathrm{Msg}_i\big|_{\mathrm{Msg}_{<i}=\mathrm{msg}_{<i}}\big)\big)\big] \\
&= {}^1\!/\!{}_{\varepsilon^3} \cdot \operatorname{Var}\big[\mathrm{jump}(\mathrm{Msg}_{\leq i}) \mid \mathrm{Msg}_{<i} = \mathrm{msg}_{<i}\big] \\
&= {}^1\!/\!{}_{\varepsilon^3} \cdot \operatorname{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}\big].
\end{aligned} \tag{8}$$

It follows that

$$\mathbb{E}\big[S_n - S_0\big] = {}^1\!/\!{}_{\varepsilon^3} \cdot \mathbb{E}\big[\sum_{i=1}^{n} \operatorname{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}]\big] \tag{9}$$

and since both S_0 and S_n take values in $[0,1]$, we conclude that $\mathbb{E}\big[\sum_{i=1}^n \operatorname{Var}[Y_i \mid \widehat{\mathrm{Msg}}_{<i}]\big] \leq \varepsilon^3$.

2.2 Attacking Robust Many-Turn Coin Flip

Moving to many-turn protocols, one can no longer decide whether to attack a message *independently* of the other messages. There are simply too many massages, and whatever such strategy one takes, it either corrupts too many parties, or biases the protocol's outcome by too little. So rather, the strategy to consider is to decide whether to corrupt or not, *per party*, and not per message, with the exception of "highly influential" messages. Given this mandatory change, the main challenge is that once deciding to corrupt a party, we should corrupt its messages in a way that does not significantly reduce the the influence of its future messages. Otherwise, a corrupt party might never be useful for biasing the protocol outcome. For instance, consider the n-party n^2-round majority protocol, i.e., each party sends n bits, that is equipped with the following "punishing" 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. If the protocol is single turn (i.e., each party sends all its bis in a single message), then the effect of such punishing mechanism can be taken into consideration at the time of corrupting the party (single) message. But if the parties send their bits in turns (compared to all at once, like in the single-turn case), the task of attacking protocols with more (implicit) unpredictable mechanisms that determine message influence, is much more challenging. Here the gentle approach described in the previous section is extremely useful. Actually, being useful in such scenarios is what we had in mind when designing it in the first place.

A second challenge is that that attacker has to decide whether to corrupt a party *before* it is certain the party can be used to significantly bias the protocol outcome; the influence a party has on the final outcome might vary between different executions, and there is no way to tell in advance whether a certain party will be influential since other parties (which we do not corrupt) introduce randomness to the process. We overcome this obstacle by choosing the parties to corrupt not merely based on the influence of their next message, as we did in the single-turn case. Rather, in addition to the single-case mechanism, each party is corrupted with (fixed) probability $1/\sqrt{n}$, and it reenters the "lottery" every time the messages it sent had sufficient (potential) influence on the outcome. To keep the attack gentle, the messages of the chosen to be corrupted party are modified only to gain a moderate bias.

Normal protocols. With the realization that parties should be given multiple chances to become corrupted (the "lottery" mentioned above), the presentation of the attack is simplified by splitting the parties into pseudo-parties for which we decide *independently* whether to corrupt or not. Those pseudo-parties have relatively small *overall* influence on the protocol's outcome, except the unavoidable case of having a single very influential message.

Definition 2.4 (Normal protocols, informal)**.** *Let* $\mathsf{party}(\mathrm{msg}_{<i})$ *be the identity of the party about to send the* i^{th} *message as described by the partial transcript* $\mathrm{msg}_{<i}$. *A party* P *has* a large jump *in* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg})$, *if* $\mathsf{party}(\mathrm{msg}_{<i}) = \mathsf{P}$ *for some* $i \in [\ell]$ *such that*

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

An n*-party protocol is* normal *if the following hold:*[7]

1. *A large-jump party sends only a* single *message.*
2. *For a small-jumps party, a party that has no large jumps,* P *it holds that*

$$\sum_{i \in [\ell]\colon \mathsf{party}(\mathrm{Msg}_{<i}) = \mathsf{P}} \mathrm{Var}[\mathrm{jump}(\mathrm{Msg}_{\leq i}) \mid \mathrm{Msg}_{<i}] \leq 1/n$$

(I.e., the overall sum of conditional variances the party P *"has" is bounded.)*

We enforce normality by partitioning, if needed, the messages of the parties into sequences, viewing each such sequence as a separate pseudo-party. The advantage of considering protocols in their normal form is that our attack, described below, either corrupts *all* messages sent by a (pseudo-)party, or corrupts *none* of them.

The attack. Our attacker on an n-party, ℓ-round, robust, normal protocol Π is defined as follows:

Algorithm 2.5 (Many-turn attacker)**.**

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* $\mathrm{jump}_i := \mathrm{jump}(\mathrm{msg}_{<i}, \cdot)$*, and let* $v_i := \mathrm{Var}[\mathrm{jump}_i(Q_i)]$.

[7] Actually, we can only guarantee a relaxed variant of the normality condition.

2. *If $v_i \geq 1/n$, corrupt the party sending the i^{th} message with probability $1/\varepsilon^3 \cdot \sqrt{v_i}$. If corrupted, instructs it to send its next message according to* $\text{Biased}^{\text{jump}_i}_{1/\sqrt{v_i}}(Q_i)$.

   Else, if the i^{th} message is the first message to be sent by the party, corrupt this party with probability $1/\varepsilon^3 \cdot 1/\sqrt{n}$. If corrupted (now or in previous rounds), instruct it to send its next message according to $\text{Biased}^{\text{jump}_i}_{\sqrt{n}}(Q_i)$.

That is, a large-jump party is treated like in the single-turn case, whereas a small-jumps party is corrupted with probability proportional to $1/\sqrt{n}$ (again like in the single-turn case)—when corrupted, *all* messages of the small-jumps party are modified.[8] The analysis of the above attack is similar to the single-turn case. Let $\widehat{\text{Msg}} = (\widehat{\text{Msg}}_1, \ldots, \widehat{\text{Msg}}_\ell)$, $S = (S_0, \ldots, S_\ell)$ 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 2.6. $\mathbb{E}\left[\sum_{i=1}^{\ell} \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i}]\right] = O(\varepsilon^3)$.

The challenge in proving Lemma 2.6 is that unlike the single-turn proof, it might be that the following does not hold:[9]

$$\mathbb{E}\left[S_i - S_{i-1} \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}\right] \geq 1/\varepsilon^3 \cdot \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}]$$

Indeed, let V_i be the value of the variables v_i in the execution of the attack described by $\widehat{\text{Msg}}$. Assume that conditioned on $\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}$, it holds that $V_i < 1/n$ and that a small-jumps 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 be $1/\varepsilon^3 \cdot 1/\sqrt{n}$: the previous messages sent by P in $\text{msg}_{<i}$ 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 $\mathbb{E}\left[S_i - S_{i-1} \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}\right] < 1/\varepsilon^3 \cdot \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}]$. Fortunately, since we only slightly modify each message of a corrupted small-jumps party (proportionally to the conditional variance the message induces on the protocol's outcome), and since (due to the partitioning) the overall variance of the messages a small-jumps party sends is at most $1/n$, a KL-divergence argument yields that on average (in some sense) for a message sent by a small-jumps party it holds that $\mathbb{E}\left[S_i - S_{i-1} \mid \widehat{\text{Msg}}_{i-1} = \text{msg}_{<i}\right] = \Omega(1/\varepsilon^3 \cdot \text{Var}\left[Y_i \mid \widehat{\text{Msg}}_{i-1} = \text{msg}_{<i}\right])$, 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 (has large jumps to both directions) is trying to transform it into a (almost) robust protocol with respect to $b = 0$ (with no large jumps downward), with the assurance that if we fail transforming it into a robust protocol, it would

[8]Assuming Π is an n-party, n^2-round, single-bit majority protocol in which each party sends n bits, then (typically) the change induced by any given message is (absolute) order of $1/n$. Hence, each v_i is of order $1/n^2$, and each party will be independently corrupted with probability $1/\varepsilon^3 \cdot 1/\sqrt{n}$ (i.e., first *if* of Step (2) is never triggered). Thus, in expectation, the above attack corrupts $1/\varepsilon^3 \cdot \sqrt{n}$ parties. If corrupt, each of the n bit-messages the party sends is 1 with probability $\approx 1/2 \cdot (1 + \sqrt{n} \cdot 1/n) = 1/2 + 1/2\sqrt{n}$.

[9]Recall that the above equality allowed us to argue that $\mathbb{E}\left[\sum_{i=1}^{\ell} \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i}]\right] \leq \varepsilon^3 \cdot \mathbb{E}\left[S_\ell - S_0\right] \leq \varepsilon^3$.

already be completely biased (towards 0). If we succeeded transforming it into a (almost) robust protocol we apply our attack on robust protocols.

More formally, assume that with probability at least $1/\log n$, Π has a large negative jump, i.e., $-1/\sqrt{n}$, and consider the following "one-shot" attacker on Π:[10]

Algorithm 2.7 (Negative jumps attacker)**.**

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

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

It is clear that the above adversary biases the outcome of Π toward zero by at least $1/\sqrt{n}\log(n)$. 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 $t \leq \sqrt{n} \cdot \log(n)$ denote the number of these attack applications. If the expected outcome of Π_t is at most ε, then we are done: the t-adaptive adversary that runs these t attacks iteratively, makes Π output 0 with probability $1 - \varepsilon$. Otherwise, Π_t has the following property:

$$\mathbb{P}\left[\exists j \in [n] \colon \Pi_t(\widetilde{\text{Msg}}_{\leq j}) < \Pi_t(\widetilde{\text{Msg}}_{<j}) - 1//\sqrt{n}\right] \leq 1/\log(n) \tag{10}$$

letting $\widetilde{\text{Msg}}$ be the messages of a random execution of Π_t. If the above happens, then we apply the attack on robust protocols (Algorithm 2.5) on Π_t, instructing the adversary to halt if it encounters a large negative jump. With careful analysis (actually, we need to slightly refine the attack for that), one can show that the above attack on Π_t encounters large negative jumps with probability $O(\varepsilon)$ (recall that we set ε to $1/\text{loglog}(n)$). Hence, it successfully biases the expected output of Π_t to $1 - O(\varepsilon)$ (since with overwhelming probability the attack carries as if there are no large negative jumps). Composing the attack that transforms Π into Π_t with the attack on Π_t, yields the required attack on Π.

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 $[n] := \{1, \ldots, n\}$ and $(n) := \{0, \ldots, n\}$. Given a Boolean statement S (e.g., $X \geq 5$), let $\mathbb{1}_S$ be the indicator function that outputs 1 if S is a true statement and 0 otherwise.

[10]Interestingly, assuming the next-message function of Π is efficient, e.g., Π is public-coin, the following attacker is the only reason for the inefficiency of our attack.

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, 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 for statistical distance)**.** *For distributions P and Q and function f over a discrete domain $\mathcal{X}$, it holds that $\mathsf{SD}(f(P), f(Q)) \leq \mathsf{SD}(P, Q)$.*

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

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

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

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

The following fact (see Fedotov et al. [12]) relates small KL-divergence to small statistical distance:

Fact 3.3 (Pinsker bound)**.** *For discrete distributions P and Q it holds that $\mathsf{SD}(P, Q) \leq \sqrt{\frac{1}{2} \cdot D_{\mathrm{KL}}\left(P \,\|\, Q\right)}$.*

3.3 Martingales

Martingales play an important role in our analysis.

Definition 3.4 (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 $\mathbb{E}\left[M_{k+1} \mid X_{\leq k}\right] = M_k$ 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 3.5 (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 $M_k := \mathbb{E}\left[f(X_1, \ldots, X_n) \mid X_{\leq k}\right]$.*

The proof of the following known fact is immediate.

Fact 3.6 (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 $\mathbb{E}\left[X_k \mid Z_{<k}\right] = 0$ and X_k is determined by $Z_{\leq k}$ (i.e., $\sum X_i$ is a martingale with respect to Z_k), then* $\mathrm{Var}[\sum_{i=1}^n X_i] = \sum_{i=1}^n \mathrm{Var}[X_i]$.

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

Definition 3.7 (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 $\mathbb{E}\left[S_{k+1} \mid X_{\leq k}\right] \geq S_k$ 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.

Lemma 3.8 (Doob's maximal inequality)**.** *Let $S_1, \ldots, S_n$ be a non-negative sub-martingale, then for any $c > 0$ it holds that $\mathbb{P}\left[\sup_k S_k \geq c\right] \leq \mathbb{E}\left[S_n\right]/c$.*

3.4 Full-Information Coin Flip

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

Definition 3.9 (Full-information coin-flipping protocols)**.** *A protocol Π is a* full-information coin-flipping protocol *if it is* stateless,[11] *the parties keep no private state between the different communication rounds, and each turn consists of a* single *party broadcasting a string, and the parties common output is a deterministic Boolean function of the transcript.*

Remark 3.10 (Many messages per communication round)**.** *Note that we restrict that in each round only a single party broadcasts a message. Our attack readily applies for 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 on 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.*

We associate the the following notation with a full-information coin-flipping protocol.

Notation 3.11. *Let Π be an n-party, ℓ-party, full-information coin-flipping protocol.*

- *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 $\mathrm{msg}_{\leq i} \in \mathrm{Supp}\left(\mathrm{Msg}^\Pi_{\leq i}\right)$, let $\Pi\left(\mathrm{msg}_{\leq i}\right) := \mathbb{E}\left[\Pi\left(\mathrm{Msg}^\Pi\right) \mid \mathrm{Msg}^\Pi_{\leq i} = \mathrm{msg}_{\leq i}\right]$.*

  (I.e., the expected outcome of Π given $\mathrm{msg}_{\leq i}$)

  Let $\mathbb{E}\left[\Pi\right] := \Pi()$, and refer to this quantity as the expected outcome of Π.

[11]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, before it acts, samples its state *conditioned on the current public transcript.* It is easy to see that an attack on the stateless variant is also an attack, with exactly the same parameters, on the original (stateful) protocol.

- *For* $\mathrm{msg}_{<i} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi}_{<i})$, *let* $\mathsf{party}(\mathrm{msg}_{<i}) \in [n]$ *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* $\mathcal{I}dx_{\mathsf{P}}(\mathrm{msg}) := \{i \in [\ell] \colon \mathsf{party}(\mathrm{msg}_{<i}) = \mathsf{P}\}$.

- *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.)*

3.4.1 Adaptive Adversaries

Definition 3.12 (Adaptive adversary)**.** *A* t-adaptive adversary *for a full-information coin-flipping protocol in an* 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 in which a corrupted party is speaking, the adversary has* full control *over the message it sends, but bounded to send a valid message (i.e., in the protocol message space support).*

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

The attacked protocol.

Definition 3.13 (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.

Derandomizing.

Proposition 3.14 (Attack derandomization)**.** *For an adversary* A *acting on a full-information coin-flipping protocol* Π *there exist* deterministic *adversaries* A^{+} *and* A^{-} *such that* $\mathbb{E}\big[\Pi_{\mathsf{A}^{+}}\big] \geq \mathbb{E}\big[\Pi_{\mathsf{A}}\big]$ *and* $\mathbb{E}\big[\Pi_{\mathsf{A}^{-}}\big] \leq \mathbb{E}\big[\Pi_{\mathsf{A}}\big]$.

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

Composition.

Definition 3.15 (Composing adaptive adversaries)**.** *Let* Π *be a coin-flipping protocol, let* A *be a deterministic* k_{A}*-adaptive adversary for* Π, *and let* B *be a* k_{B}*-adaptive adversary for* Π_{A}. *The* $(k_{\mathsf{A}} + k_{\mathsf{B}})$*-adaptive adversary* $\mathsf{B} \circ \mathsf{A}$ *on* Π *is defined as follows:*

Algorithm 3.16 (Adversary $\mathsf{B} \circ \mathsf{A}$ on Π)**.**

```
For $i := 1$ to NumMsgs($\Pi$):
    If $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$ would like to modify the $i^{\mathrm{th}}$ message, alter it according to $\mathsf{P}$.
```

Proposition 3.17. *Let* Π, A *and* B *be as in Definition 3.15, then* $\mathbb{E}\left[\Pi_{\mathsf{B}\circ\mathsf{A}}\right] = \mathbb{E}\left[(\Pi_{\mathsf{A}})_{\mathsf{B}}\right]$.

Proof. Since A is deterministic, B never modifies a message in Π_{A} that is to be modified by A. Indeed, since B is a valid attacker it never sends a message out of the support of Π_{A}, and a message to be corrupted by A is fixed. It follows 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 3.18. *For* $-\frac{1}{2} \leq x$ *it holds that* $x\log(1+x) \leq 2x^2$.

Fact 3.19. *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, fix $n \in \mathbb{N}$ (the number of parties of the robust protocol), and make use of the following constants.

Notation 4.1. *Let* $\varepsilon := 1/\sqrt[50]{\log\log n}$, $\lambda := 100/\varepsilon^5$ *and* $\delta := 1/\log^2 n$.

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* n-robust, *if* $\mathbb{P}\left[\exists i \in [\ell] \colon \operatorname{Supp}\left(\operatorname{jump}^{\Pi}\left(\operatorname{Msg}^{\Pi}_{\leq i}\right) \mid \operatorname{Msg}^{\Pi}_{<i}\right) \cap (-\infty, -1/\lambda\cdot\sqrt{n}] \neq \emptyset\right] \leq \delta$.

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

We start, Section 4.1, by proving a variant of Theorem 4.3 for "normal" coin-flipping protocol. Informally, in a *normal* coin-flipping protocol no party influences the protocol "by much" over multiple rounds, though a party might have large influence on the outcome over a single round. In Section 4.2, we leverage this attack for proving Theorem 4.3, by transforming the given protocol into a normal protocol, and showing that the guaranteed attack on the latter protocol yields an attack of essentially the same quality on the original protocol.[12]

4.1 Biasing Normal Robust Coin Flip

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

Definition 4.4 (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:*

[12]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 aforementioned transformation.

Single non-robust party: *There exists a party* $\mathrm{NonRobust} \in [t]$ *such that for every transcript* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *and* $i \in [\ell]$*:*
$\left(\mathrm{Supp}\left(\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\leq i}\right) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\right) \cap (-\infty, -1/\lambda\cdot\sqrt{n}]\right) \neq \emptyset \iff i \in \mathcal{I}dx_{\mathrm{NonRobust}}(\mathrm{msg})$.

Large-jump party sends a single message: *For every large-jump party* P *in* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *it holds that* $|\mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})| = 1$*,*
where a party $\mathsf{P} \in [t] \setminus \{\mathrm{NonRobust}\}$ *has* large jump in msg*, if* $\exists i \in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})$ *s.t.* $\mathrm{Var}\left[\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\leq i}\right) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\right] \geq 1/\lambda n$.

Small-jumps party has bounded overall variance: *For every small-jumps party* P *in* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *it holds that:*
$\sum_{i\in\mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} \mathrm{Var}\left[\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\leq i}\right) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\right] \leq 2 \cdot 1/\lambda n$*,*
where a party $\mathsf{P} \in [t] \setminus \{\mathrm{NonRobust}\}$ *has* small jumps in msg*, if it participates (sends a message) but has no large jumps in* msg*.*

At most n unfulfilled parties: *In every transcript* $\mathrm{msg} \in \mathrm{Supp}(\mathrm{Msg}^{\Pi})$ *there are at most* n unfulfilled *parties,*
where a small-jumps party is unfulfilled in msg *if*
$\sum_{i\in\mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} \mathrm{Var}\left[\mathrm{jump}^{\Pi}\left(\mathrm{Msg}^{\Pi}_{\leq i}\right) \mid \mathrm{Msg}^{\Pi}_{<i} = \mathrm{msg}_{<i}\right] < 1/\lambda n$.

In the following, We say that a party P is a large-jump party with respect to msg if it has a large jump in msg, and is a small-jumps party with respect to msg if it has small-jumps in msg.

The advantage of considering normal protocols is that for such protocols our attack either corrupts (essentially) *all* message sent by a party, or corrupts *none* of them. Our attack can be easily adapted for arbitrary (non-normal) protocols, and then it only corrupts, if at all, a (typically small, non continuous) subset of the a party's messages.

Our attack on normal coin-flipping protocols is stated below.

Lemma 4.5 (Biasing normal coin-flipping protocols)**.** *Let* Π *be an* n*-normal,* ℓ*-round, full-information coin-flipping protocol. If* $\mathbb{E}\left[\Pi\right] \geq \varepsilon$ *and* $\mathbb{P}\left[\mathcal{I}dx_{\mathrm{NonRobust}}(\mathrm{Msg}^{\Pi}) \neq \emptyset\right] \leq \delta$*, then there exists an* $O(\sqrt{n} \cdot \log n)$*-adaptive adversary* A *such that* $\mathbb{E}\left[\Pi_{\mathsf{A}}\right] \geq 1 - O(\varepsilon)$*.*

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

We begin by introducing a method of biasing distributions in order to increase their expectation under some utility function. Jumping ahead, our adversary will use this technique in order to modify the messages of the corrupted parties, with the utility function being the change it induces on the protocol's expectation.

Definition 4.6 (Biased distribution)**.** *Let* X *be a distribution,* $\alpha > 0$ *and* $f\colon \mathrm{Supp}(X) \mapsto [-1/\alpha, \infty)$ *such that* $\mathbb{E}\left[f(X)\right] = 0$*. We define the distribution* $\mathrm{Biased}^{f}_{\alpha}\left(X\right)$ *as follows:* $\mathbb{P}\left[\mathrm{Biased}^{f}_{\alpha}\left(X\right) = x\right] = \mathbb{P}\left[X = x\right] \cdot (1 + \alpha f(x))$*.*

It is easy to verify this is indeed a distribution. If f is the identity function, we sometimes omit it from the above notation.

Lemma 4.7 (Properties of the Biased distribution)**.** *Let* X *be a distribution,* $\alpha > 0$ *and* f : $\mathrm{Supp}(X) \mapsto [-1/2\alpha, \infty)$ *such that* $\mathbb{E}\left[f(X)\right] = 0$*. The following hold:*

1. $\mathbb{E}\left[f\left(\mathrm{Biased}^{f}_{\alpha}\left(X\right)\right)\right] = \alpha \cdot \mathrm{Var}\left[f(X)\right]$.

2. *$D_{\mathrm{KL}}\big(\mathrm{Biased}^f_\alpha(X) \,\|\, X\big) \leq 2\alpha^2 \cdot \mathrm{Var}\big[f(X)\big]$.*

3. *For any $0 \leq p \leq 1$ it holds that $\big(p \cdot \mathrm{Biased}^f_\alpha(X) + (1-p) \cdot X\big) \equiv \mathrm{Biased}^f_{p\alpha}(X)$.*

4. *There exists a pair of random variables (A, B), i.e., a coupling, such that $A \equiv X$, $B \equiv \mathrm{Biased}^f_\alpha(X)$, and $f(B) \geq f(A)$.*

Proof.

Item 1:

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

Item 2:

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

The last inequality follows by Jensen's inequality and Fact 3.18.

Item 3:

$$\begin{aligned}
&\mathbb{P}\Big[\Big(p \cdot \mathrm{Biased}^f_\alpha(X) + (1-p) \cdot X\Big) = x\Big] \\
&= p \cdot \mathbb{P}\big[\mathrm{Biased}^f_\alpha(X) = x\big] + (1-p) \cdot \mathbb{P}\big[X = x\big] \\
&= p \cdot \mathbb{P}\big[X = x\big] \cdot (1 + \alpha f(X)) + (1-p) \cdot \mathbb{P}\big[X = x\big] \\
&= \mathbb{P}\big[X = x\big] \cdot (1 + p\alpha f(X)).
\end{aligned}$$

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

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

By construction $f(b) \geq f(a)$, and it is not hard to verify that the marginal distributions of a and b are that of X and $\mathrm{Biased}^f_\alpha(X)$, respectively.

□

In the rest of this subsection we fix an n-normal, ℓ-round, full-information coin-flipping protocol Π such that $\mathbb{E}\left[\Pi\right] \geq \varepsilon$ and $\mathbb{P}\left[\mathcal{I}dx_{\text{NonRobust}}(\text{Msg}) \neq \emptyset\right] \leq \delta$, for $\text{Msg} \equiv \text{Msg}^{\Pi}$. The attacker for Π is defined as follows.

Algorithm 4.8 (Adversary A)**.**

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

1. *Let* P *be the the party about to send the* i^{th} *message. If* $\mathsf{P} = \text{NonRobust}$, *do not intervene in the current round.*

2. *Let* $\text{msg}_{<i}$ *denote the messages sent in the previous rounds. Let* Q_i *be the distribution* $\text{Msg}_i^{\Pi}\big|_{\text{Msg}^{\Pi}_{<i}=\text{msg}_{<i}}$, *let* $\text{jump}_i := \text{jump}^{\Pi}(\text{msg}_{<i}, \cdot)$ *and let* $v_i := \text{Var}\left[\text{jump}_i(Q_i)\right]$.

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* $\lambda^2 \cdot \sqrt{v_i}$. [13]

   (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* $\text{Biased}^{\text{jump}_i}_{\sqrt{v_i}}\left(Q_i\right)$.

   (b) **Else***:*

      i. **If** $\mathbb{P}\left[\mathsf{P} \text{ is corrupted by } \mathsf{A} \mid \text{Msg}^{\mathsf{A}}_{<i} = \text{msg}_{<i}\right] \leq {}^{16\lambda^2}\!/\!_{\sqrt{n}}$, *for* $\text{Msg}^{\mathsf{A}}_{<i}$ *which is the distribution of messages sent in a random execution of* Π *in the presence of* A *for the first* $i-1$ *rounds,*[14] *instruct* P *to broadcast its next message according to* $\text{Biased}^{\text{jump}_i}_{\sqrt{n}}\left(Q_i\right)$.

      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.[15]

In the rest of this section we analyze the expected outcome of Π_{A} and the number of parties A corrupts. Let $\widehat{\text{Msg}} = (\widehat{\text{Msg}}_1, \ldots, \widehat{\text{Msg}}_\ell)$ denote the messages sent in a random execution of A on Π. Let $Q_1 \ldots, Q_\ell$ be the value of these variables computed by A, and let CorruptedParties be the set of parties corrupted in this execution of A on Π (all variables are jointly distributed with $\widehat{\text{Msg}}$). Note that CorruptedParties is *not* determined by $\widehat{\text{Msg}}$ (as there is additional randomness involved). Let $S_0, \ldots, S_\ell$ be the *sub-martingale* $S_k := \Pi\left(\widehat{\text{Msg}}_{\leq k}\right)$ with respect to $\widehat{\text{Msg}}$. Let $X_k := S_k - S_{k-1} = \text{jump}^{\Pi}\left(\widehat{\text{Msg}}_{\leq k}\right)$ be the jumps induced by the attacked execution. Note that $S_0 = \mathbb{E}\left[\Pi\right] \geq \varepsilon$ and $S_\ell = S_0 + \sum_{i=1}^{\ell} X_i$.

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

[14] Since we are defining the strategy of A in the i^{th} round using its strategy in the first $i-1$ rounds, this self-reference is well defined.

[15] We are not certain 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 4.10).

We prove Lemma 4.5 using the following three observations. The first observation, proved in Section 4.1.1, guarantees a per-round coupling between the change in expected outcome induced by the attack, and what that would have been the change in an honest execution (conditioned on previous messages).

Claim 4.9 (Coupling honest and attacked conditional distributions)**.** *There exists a random variable $Y = (Y_1, \ldots, Y_\ell)$ jointly distributed with $\widehat{\mathrm{Msg}}$, such that the following holds for every $i \in [\ell]$, $\mathrm{msg}_{<i} \in \mathrm{Supp}\left(\widehat{\mathrm{Msg}}_{<i}\right)$ and $c \in \{0,1\}$:*

1. $X_i \geq Y_i$,
2. $Y_i|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i}} \equiv \mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, Q_i)$ *where* $\mathsf{P} = \mathsf{party}(\mathrm{msg}_{<i})$*. Furthermore, conditioned on* $\widehat{\mathrm{Msg}}_{<i}$*,* Y_i *is independent of* $Y_{<i}$ *and* $\{\mathsf{P} \in \mathrm{CorruptedParties}\}$*.*

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 $\mathbb{E}\left[Y_k \mid \widehat{\mathrm{Msg}}_{<k}, Y_{<k}\right] = 0$, i.e., $\sum_{i=1}^{k} Y_i$ is a martingale with respect to $(\widehat{\mathrm{Msg}}_k, Y_k)$. For the rest of this section, let Y be the random variable guaranteed by Claim 4.9.

Next, we consider the set of robust messages with respect to $\widehat{\mathrm{Msg}}$, defined by

$$\mathrm{RobustJumps} := [\ell] \setminus \mathcal{I}dx_{\mathrm{NonRobust}}(\widehat{\mathrm{Msg}}) \tag{11}$$

Note that RobustJumps is a random set (determined by $\widehat{\mathrm{Msg}}$). The following observation (proved in Section 4.1.2) states that the overall conditional variance of Y contributed by the robust messages 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 $X_i \geq Y_i$, that $\sum X_i$ is typically not too small.

Claim 4.10 (Bounding Y's conditional variance)**.** $\mathbb{E}\left[\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\left[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\right]\right] < {}^{2}/_{\lambda}$.

Finally, in Section 4.1.3 we prove that attacked execution does not deviate too much, in KL-divergence terms, from the honest execution. This implies that, with overwhelming probability, NonRobust does not participate in the protocol (since it participated in the original protocol with very small probability).

Claim 4.11 (Bounding KL-Divergence between attacked and honest executions)**.** *It holds that* $D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}} \,\|\, \mathrm{Msg}^{\Pi}\right) \leq 16^3 \lambda^3$*.*

Equipped with Claims 4.9 to 4.11, we are ready to prove Lemma 4.5.

Proving Lemma 4.5

Proof of Lemma 4.5.

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

$$\operatorname{Var}\Big[\sum_{i \in \text{RobustJumps}} Y_i\Big] = \sum_{i=1}^{n} \operatorname{Var}\big[Y_i \cdot \mathbb{1}_{i \in \text{RobustJumps}}\big] = \mathbb{E}\Big[\sum_{i \in \text{RobustJumps}} \operatorname{Var}\big[Y_i \mid \widehat{\text{Msg}}_{<i}\big]\Big] \leq {}^{2}/_{\lambda} \quad (12)$$

The first equality holds by Fact 3.6 (since $\mathbb{E}\big[Y_k \mid \widehat{\text{Msg}}_{<k}, Y_{<k}\big] = 0$ and $\mathbb{1}_{i \in \text{RobustJumps}}$ is determined by $\widehat{\text{Msg}}_{<k}$). The inequality holds by Claim 4.10. Since $\mathbb{E}\big[\sum_{i \in \text{RobustJumps}} Y_i\big] = 0$. Thus, by Chebyshev's inequality

$$\mathbb{P}\Big[\sum_{i \in \text{RobustJumps}} Y_i \leq -{}^{\varepsilon}/_{2}\Big] \leq \mathbb{P}\Big[\Big|\sum_{i \in \text{RobustJumps}} Y_i\Big| \geq {}^{\varepsilon}/_{2}\Big] < {}^{\varepsilon}/_{4} \quad (13)$$

Next we show that with overwhelming probability RobustJumps $= [\ell]$, namely NonRobust does not participate in the execution. Let Bad^{Π} be the event $\{\mathcal{I}dx_{\text{NonRobust}}(\text{Msg}) \neq \emptyset\}$. By assumption, $\mathbb{P}\big[\text{Bad}^{\Pi}\big] \leq \delta$. Let Bad be the event $\Big\{\mathcal{I}dx_{\text{NonRobust}}(\widehat{\text{Msg}}) \neq \emptyset\Big\}$. Note that

$$D_{\text{KL}}\big(\mathbb{1}_{\text{Bad}} \,\|\, \mathbb{1}_{\text{Bad}^{\Pi}}\big) \leq D_{\text{KL}}\big(\widehat{\text{Msg}} \,\|\, \text{Msg}^{\Pi}\big) \leq 16^3 \lambda^3 \quad (14)$$

The first inequality follows by data-processing of KL Divergence, and the last inequality follows by Claim 4.11. It follows that

$$\begin{aligned} D_{\text{KL}}\big(\mathbb{1}_{\text{Bad}} \,\|\, \mathbb{1}_{\text{Bad}^{\Pi}}\big) &= \mathbb{P}\big[\text{Bad}\big] \cdot \log\left(\frac{\mathbb{P}\big[\text{Bad}\big]}{\mathbb{P}\big[\text{Bad}^{\Pi}\big]}\right) + \big(1 - \mathbb{P}\big[\text{Bad}\big]\big) \cdot \log\left(\frac{1 - \mathbb{P}\big[\text{Bad}\big]}{1 - \mathbb{P}\big[\text{Bad}^{\Pi}\big]}\right) \quad (15) \\ &= \mathbb{P}\big[\text{Bad}\big] \cdot \log\left(\frac{\mathbb{P}\big[\text{Bad}\big]}{\mathbb{P}\big[\text{Bad}^{\Pi}\big]}\right) + \big(1 - \mathbb{P}\big[\text{Bad}\big]\big) \cdot \big(\log\big(1 - \mathbb{P}\big[\text{Bad}\big]\big) - \log\big(1 - \mathbb{P}\big[\text{Bad}^{\Pi}\big]\big)\big) \\ &\geq \mathbb{P}\big[\text{Bad}\big] \cdot \log\left(\frac{\mathbb{P}\big[\text{Bad}\big]}{\mathbb{P}\big[\text{Bad}^{\Pi}\big]}\right) + (-1 + 0) \geq \mathbb{P}\big[\text{Bad}\big] \cdot \log\left(\frac{\mathbb{P}\big[\text{Bad}\big]}{\delta}\right) - 1. \end{aligned}$$

The first inequality follows by Fact 3.19. Assume by the way of contradiction that $\mathbb{P}\big[\text{Bad}\big] \geq {}^{\varepsilon}/_{4}$. Thus (for large enough n),

$$\sqrt{\log\log n} > 16^3 \lambda^3 \geq D_{\text{KL}}\big(\mathbb{1}_{\text{Bad}} \,\|\, \mathbb{1}_{\text{Bad}^{\Pi}}\big) \geq \frac{\log\big(\log^2 n/4 \sqrt[50]{\log\log n}\big)}{4\sqrt[50]{\log\log n}} - 1 \geq \sqrt{\log\log n},$$

yielding the contraction $\sqrt{\log\log n} > \sqrt{\log\log n}$ (for large enough n). We conclude that $\mathbb{P}\big[\mathcal{I}dx_{\text{NonRobust}}(\widehat{\text{Msg}}) \neq \emptyset\big] = \mathbb{P}\big[\text{Bad}\big] \leq {}^{\varepsilon}/_{4}$. Combining the above observations, we conclude that

$$\begin{aligned} \mathbb{E}\big[\Pi_{\mathsf{A}}\big] &= \mathbb{P}\big[S_\ell = 1\big] = \mathbb{P}\big[S_\ell > 0\big] = \mathbb{P}\Big[S_0 + \sum_{i=1}^{\ell} X_i > 0\Big] = \mathbb{P}\Big[\sum_{i=1}^{\ell} X_i > -S_0 \geq -\varepsilon\Big] \quad (16) \\ &\geq \mathbb{P}\Big[\sum_{i \in \text{RobustJumps}} Y_i > -\varepsilon\Big] - \mathbb{P}\big[\mathcal{I}dx_{\text{NonRobust}}(\widehat{\text{Msg}}) \neq \emptyset\big] \geq (1 - {}^{\varepsilon}/_{4}) - {}^{\varepsilon}/_{4} \geq 1 - {}^{\varepsilon}/_{2}. \end{aligned}$$

Number of corruptions. So it is left to prove that A does not perform too many corruptions. We do that by calculating the *expected* number of corruptions, and use a Markov bound. 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 $\widehat{\mathrm{Msg}}$, respectively. Let $\mathrm{SmallJumps} := \left\{k \in [\ell] \colon \mathsf{party}(\widehat{\mathrm{Msg}}_{<k}) \in \mathrm{SmallParties}\right\}$ be the set of small jumps, and let $\mathrm{LargeJumps} := \left\{k \in [\ell] \colon \mathsf{party}(\widehat{\mathrm{Msg}}_{<k}) \in \mathrm{LargeParties}\right\}$ be the set of large jumps. Note that all the above random sets are determined by $\widehat{\mathrm{Msg}}$.

We first notice that since a small-jumps party is corrupted with probability ${\lambda^2}/{\sqrt{n}}$, it holds that

$$\mathbb{E}\big[|\mathrm{SmallParties} \cap \mathrm{CorruptedParties}|\big] = {\lambda^2}/{\sqrt{n}} \cdot \mathbb{E}\big[|\mathrm{SmallParties}|\big] \tag{17}$$

In addition, the definition of an n-normal protocols stipulates that for any transcript of Π there are at most n unfulfilled parties. Since each non-unfulfilled party contributes at least ${1}/{\lambda n}$ to the sum of variances, which is small by Claim 4.10, we deduce that

$$\mathbb{E}\big[|\mathrm{SmallParties}|\big] \le 3n \tag{18}$$

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

$$\mathbb{E}\big[|\mathrm{SmallParties} \cap \mathrm{CorruptedParties}|\big] \le {\lambda^2}/{\sqrt{n}} \cdot 3n = 3\lambda^2\sqrt{n} \tag{19}$$

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

$$\begin{aligned}
&\mathbb{E}\big[|\mathrm{LargeParties} \cap \mathrm{CorruptedParties}|\big] \qquad (20)\\
&= \mathbb{E}\Big[\sum_{i \in \mathrm{LargeJumps}} \lambda^2 \cdot \sqrt{\mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]}\Big] \le \mathbb{E}\Big[\sum_{i \in \mathrm{LargeJumps}} \lambda^2 \cdot \sqrt{\lambda n} \cdot \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\Big]\\
&\le \lambda^3 \cdot \sqrt{n} \cdot \mathbb{E}\Big[\sum_{i \in \mathrm{RobustJumps}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\Big] \le \lambda^3 \cdot \sqrt{n} \cdot {2}/{\lambda} = 2\lambda^2\sqrt{n}.
\end{aligned}$$

The last inequality follows by Claim 4.10. Thus in total, the expected amount of corruptions is at most $5\lambda^2\sqrt{n}$. Hence by Markov 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

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

which concludes the proof of the lemma. □

4.1.1 Coupling X_i and Q_i, Proving Claim 4.9

Proof of Claim 4.9. Fix $i \in [\ell]$ and let C be the event $\left\{\mathsf{party}(\widehat{\mathrm{Msg}}_{<i}) \in \mathrm{CorruptedParties}\right\}$. We show how to sample Y_i, jointly with $\mathbb{1}_C$ and $\mathrm{Msg}_{\le i}$, such that $Y_i|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i}}$ has the stated distribution for every $\mathrm{msg}_{<i} \in \mathrm{Supp}\left(\widehat{\mathrm{Msg}}_{<i}\right)$.

To that end, we define a random variable Z_i, jointly distributed with $\widehat{\mathrm{Msg}}_{\le i}$ and independent of $Y_{<i}$ and C. Fix $\mathrm{msg}_{\le i} \in \widehat{\mathrm{Msg}}_{\le i}$, to sample $Z_i|_{\widehat{\mathrm{Msg}}_{\le i}=\mathrm{msg}_{\le i}}$ we make use of Lemma 4.7(4): let Q_i, v_i, and jump_i be the values of these variables (in the execution of Algorithm 4.8), determined by $\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}$. If the condition of Step 4b does not hold (i.e., $\mathsf{party}(\mathrm{msg}_{<i})$ is a small-jumps party but $\mathbb{P}\left[C \mid \widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}\right] > {}^{16\lambda^2}/_{\sqrt{n}}$), let $\alpha = 0$. Else, let $\alpha = \min\{{}^1/_{\sqrt{v_i}}, {}^1/_{\sqrt{n}}\}$. Let (A, B) be the random variables guaranteed by Lemma 4.7(4) with respect to $X := Q_i$, $f := \mathrm{jump}_i$ and α. Sample $Z_i|_{\widehat{\mathrm{Msg}}_{\le i}=\mathrm{msg}_{\le i}}$ (independently of $Y_{<i}$ and C) according to $\mathrm{jump}_i(A|_{B=\mathrm{msg}_i})$.

By Lemma 4.7(4), $\mathrm{jump}_i(B) \ge \mathrm{jump}_i(A)$ and thus $X_i|_{\widehat{\mathrm{Msg}}_{\le i}=\mathrm{msg}_{\le i}} \equiv \mathrm{jump}_i(\mathrm{msg}_i) \ge \mathrm{jump}_i(A|_{B=\mathrm{msg}_i}) \equiv Z_i$. In addition, $Z_i|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i} \wedge C} \equiv \mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, Q_i)$ for any $\mathrm{msg}_{<i} \in \mathrm{Supp}(\widehat{\mathrm{Msg}}_{<i})$. This holds since $B|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i} \wedge C} \equiv \mathrm{Biased}^f_\alpha(Q_i) \equiv \widehat{\mathrm{Msg}}_i|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i} \wedge C}$, and thus $A|_{\widehat{\mathrm{Msg}}_{<i}=\mathrm{msg}_{<i} \wedge B=\widehat{\mathrm{Msg}}_i} \equiv Q_i$.

To conclude the proof, set $Y_i = \begin{cases} X_i & \mathbb{1}_C = 0 \\ Z_i & \mathbb{1}_C = 1 \end{cases}$. It is clear that $X_i \ge Y_i$, that the conditional distributions are as required, and that Y_i is independent of $Y_{<i}$ and C (conditioned on $\widehat{\mathrm{Msg}}_{<i}$). $\square$

4.1.2 Bounding Y's Conditional Variance, Proving Claim 4.10

Proof of Claim 4.10. Immediately follows by Claims 4.13 and 4.14, stated below, that handle large and small jumps, respectively. $\square$

Recall that $\mathrm{RobustJumps} = [\ell] \setminus \mathcal{I}dx_{\mathrm{NonRobust}}(\widehat{\mathrm{Msg}}) = \mathrm{SmallJumps} \cup \mathrm{LargeJumps}$.

In addition, we make use of the following conditional variant of the biased distribution,

Definition 4.12 (Conditional variant of Biased)**.** *Let X, Z be jointly distributed random variables, $\alpha : \mathrm{Supp}(Z) \mapsto \mathbb{R}_+$ and $f : \mathrm{Supp}(X) \mapsto \mathbb{R}$ such that $\forall z \in \mathrm{Supp}(Z)$: $f\left(X|_{Z=z}\right) \ge -{}^1/_{\alpha(z)}$, and $\mathbb{E}\left[f(X) \mid Z\right] = 0$. We define a random variable $\mathrm{Biased}^f_\alpha(X \mid Z)$ jointly distributed with Z, by $\mathrm{Biased}^f_\alpha(X \mid Z)|_{Z=z} \equiv \mathrm{Biased}^f_{\alpha(z)}(X|_{Z=z})$, for any $z \in \mathrm{Supp}(Z)$.*

Large jumps.

Claim 4.13. $\mathbb{E}\left[\sum_{i\in\mathrm{LargeJumps}} \mathrm{Var}\left[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\right]\right] < {}^1/_\lambda$.

Proof. We assume towards a contradiction that $\mathbb{E}\left[\sum_{i\in\mathrm{LargeJumps}} \mathrm{Var}\left[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\right]\right] \ge {}^1/_\lambda$, prove that this implies that $\mathbb{E}\left[S_\ell\right] > 1$, and derive a contradiction to the fact that $S_\ell \in \{0, 1\}$. For $k \in [\ell]$, let $E_k := \{k \in \mathrm{LargeJumps}\}$, and note that $\mathbb{1}_{E_k}$ is determined by $\widehat{\mathrm{Msg}}_{<k}$. Compute,

$$\mathbb{1}_{E_k} \cdot \mathbb{E}\left[X_k \mid \widehat{\mathrm{Msg}}_{<k}\right] \tag{21}$$

$$= \mathbb{1}_{E_k} \cdot \lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big]} \cdot \mathbb{E}\big[\mathrm{Biased}_{1/\sqrt{\mathrm{Var}\left[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\right]}}\big(Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big) \mid \widehat{\mathrm{Msg}}_{<k}\big] \tag{22}$$

$$+ \mathbb{1}_{E_k} \cdot \left(1 - \lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big]}\right) \cdot \mathbb{E}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big]$$

$$= \mathbb{1}_{E_k} \cdot \lambda^2 \sqrt{\mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big]} \cdot 1/\sqrt{\mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big]} \cdot \mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big] + 0 \tag{23}$$

$$= \mathbb{1}_{E_k} \cdot \lambda^2 \cdot \mathrm{Var}\big[Y_k \mid \widehat{\mathrm{Msg}}_{<k}\big].$$

Equality (22) follows by construction (see Step 3a and Step 4a of Algorithm 4.8) and Claim 4.9, and Equality (23) follows by Lemma 4.7(1). Hence (for large enough n),

$$\mathbb{E}\big[S_\ell\big] = \mathbb{E}\big[S_0 + \sum_{i=1}^{\ell} X_i\big] = \mathbb{E}\big[S_0\big] + \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \tag{24}$$

$$\geq \varepsilon + \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{1}_{\{i \in \mathrm{LargeJumps}\}} \cdot \mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \tag{25}$$

$$= \varepsilon + \mathbb{E}\big[\sum_{i=1}^{\ell} \mathbb{1}_{\{i \in \mathrm{LargeJumps}\}} \cdot \lambda^2 \cdot \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \tag{26}$$

$$= \varepsilon + \lambda^2 \cdot \mathbb{E}\big[\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \geq \varepsilon + \lambda^2/\lambda > 1.$$

Inequality (25) holds since $\mathbb{E}\big[S_0\big] \geq \varepsilon$ and $\mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big] \geq 0$, Equality (26) follows by Equation (21), and the penultimate inequality follows by assumption. The last inequality holds for sufficiently large n since λ is super-constant in n. Thus Equation (24) is in contradiction to the fact that $S_\ell \in \{0, 1\}$, and we conclude that $\mathbb{E}\big[\sum_{i \in \mathrm{LargeJumps}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] < 1/\lambda$. □

Small jumps.

Claim 4.14. $\mathbb{E}\big[\sum_{i \in \mathrm{SmallJumps}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] < 1/\lambda$.

In the following for a party P, let I_{P} be the random variable $\mathcal{I}dx_{\mathsf{P}}\Big(\widehat{\mathrm{Msg}}\Big)$.

Proof. We assume towards a contradiction that $\gamma := \mathbb{E}\big[\sum_{i \in \mathrm{SmallJumps}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \geq 1/\lambda$, and prove this implies that $\mathbb{E}\big[S_\ell\big] > 1$, which contradicts the fact that $S_\ell \in \{0, 1\}$. By definition of an n-normal protocol, for any transcript of Π there are at most n unfulfilled parties. Since all non-unfulfilled parties contribute at least $1/\lambda n$ to the sum of conditional variances, it holds that

$$\mathbb{E}\big[|\mathrm{SmallParties}|\big] \leq \gamma\lambda n + n \leq 2\gamma\lambda n \tag{27}$$

We say a party P is *contributional* if

$$\mathbb{P}\Bigg[\sum_{i \in I_{\mathsf{P}}} \mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big] > \frac{1}{8} \cdot \frac{1}{\lambda n} \mid \mathsf{P} \in \mathrm{SmallParties}\Bigg] \geq 1/8 \tag{28}$$

Note that being contributional is a function of the protocol itself, and not of a given transcript. Let ContribParties denote the set of contributional parties. In addition, let SmallContribParties be the (random) set of contributional small-jumps parties according to $\widehat{\mathrm{Msg}}$. By definition, all *non-contributional* small-jumps parties contribute at most ${}^{3\gamma}/_{4}$ to $\mathrm{Var}\big[\sum_{i\in \mathrm{SmallJumps}} Y_i\big]$. Hence, $\mathbb{E}\big[\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\sum_{i\in I_{\mathsf{P}}}\mathrm{Var}\big[Y_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \geq {}^{\gamma}/_{4}$. Since a small-jumps party has sum of conditional variances at most ${}^{2}/_{\lambda n}$, we conclude that

$$\mathbb{E}\big[|\mathrm{SmallContribParties}|\big] \geq {}^{\gamma\lambda n}/_{8} \tag{29}$$

We conclude the proof using the following claim (proven below).

Claim 4.15. *For any contributional party* P *it holds that*

$$\mathbb{E}\big[\sum_{i\in I_{\mathsf{P}}} X_i \mid \mathsf{P}\in \mathrm{CorruptedParties} \wedge \mathsf{P}\in\mathrm{SmallContribParties}\big] \geq {}^{1}/_{256\lambda\sqrt{n}}.$$

Given the above claim, compute

$$\begin{aligned}
\mathbb{E}\big[\sum_{i=1}^{\ell} X_i\big] &= \mathbb{E}\big[\sum_{i=1}^{\ell}\mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] \geq \mathbb{E}\big[\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\ \sum_{i\in I_{\mathsf{P}}}\mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big]\big] && (30)\\
&= \mathbb{E}\big[\sum_{\mathsf{P}\in\mathrm{SmallContribParties}}\ \sum_{i\in I_{\mathsf{P}}} X_i\big]\\
&= \sum_{\mathsf{P}\in\mathrm{ContribParties}} \mathbb{E}\big[\sum_{i\in I_{\mathsf{P}}} X_i \mid \mathsf{P}\in\mathrm{CorruptedParties}\wedge \mathsf{P}\in\mathrm{SmallContribParties}\big] && (31)\\
&\qquad\qquad \cdot\, \mathbb{P}\big[\mathsf{P}\in\mathrm{CorruptedParties}\wedge\mathsf{P}\in\mathrm{SmallContribParties}\big]\\
&\geq \sum_{\mathsf{P}\in\mathrm{ContribParties}} {}^{1}/_{256\lambda\sqrt{n}}\cdot \mathbb{P}\big[\mathsf{P}\in\mathrm{CorruptedParties}\wedge\mathsf{P}\in\mathrm{SmallContribParties}\big] && (32)\\
&= \sum_{\mathsf{P}\in\mathrm{ContribParties}} {}^{1}/_{256\lambda\sqrt{n}}\cdot \big(\mathbb{P}\big[\mathsf{P}\in\mathrm{SmallContribParties}\big]\cdot {}^{\lambda^2}/_{\sqrt{n}}\big) && (33)\\
&= {}^{\lambda}/_{256n}\cdot\mathbb{E}\big[|\mathrm{SmallContribParties}|\big] \geq {}^{\gamma\lambda^2}/_{2048}. && (34)
\end{aligned}$$

Inequality (30) holds since $\mathbb{E}\big[X_i \mid \widehat{\mathrm{Msg}}_{<i}\big] \geq 0$. Equality (31) holds since $\mathbb{E}\big[\sum_{i\in I_{\mathsf{P}}} X_i \mid \mathsf{P}\notin\mathrm{CorruptedParties}\wedge\mathsf{P}\in\mathrm{SmallContribParties}\big] = 0$. Inequality (32) by Claim 4.15. Equality (33) by construction (see Step 3b of Algorithm 4.8). Finally, Inequality (34) follows by Equation (29).

In total, $\mathbb{E}\big[S_\ell\big] = \mathbb{E}\big[S_0 + \sum_{i=1}^{\ell} X_i\big] \geq \varepsilon + {}^{\gamma\lambda^2}/_{2048}$. Thus, since λ is super constant in n, for large enough n it holds that $\mathbb{E}\big[\sum_{i=1}^{\ell} X_i\big] > 1$. This stands in contradiction to the fact that $S_\ell \in \{0,1\}$.

□

Proving Claim 4.15.

Proof of Claim 4.15. Fix a contributional party P, and consider the following events (jointly distributed with $\widehat{\mathrm{Msg}}$), let $C := \{\mathsf{P} \in \mathrm{CorruptedParties}\}$, let $S := \{\mathsf{P} \in \mathrm{SmallParties}\}$, let $L := \left\{\sum_{i \in I_{\mathsf{P}}} \mathrm{Var}\left[Y_i \,|\, \widehat{\mathrm{Msg}}_{<i}\right] > {}^{1}/{}_{8\lambda n}\right\}$, i.e., P has large conditional variance, and let

$$H := \left\{\forall k \in I_{\mathsf{P}}\colon \mathbb{P}\left[\mathsf{P} \in \mathrm{CorruptedParties} \,|\, \widehat{\mathrm{Msg}}_{<k}\right] < 16 \cdot {}^{\lambda^2}/{}_{\sqrt{n}}\right\},$$

i.e., Step 4(b)ii never happens for P. We start by proving that $\mathbb{P}\left[H \wedge L \,|\, S\right]$ is large, and then use a KL-divergence argument show that $\mathbb{P}\left[H \wedge L \,|\, S \wedge C\right]$ is large, i.e., P encounters large conditional variance even when it is a corrupted small-jumps party. Finally, this implies that the change in expectation P induces (when a corrupted small-jumps party) is large.

To prove that $\mathbb{P}\left[H \wedge L \,|\, S\right]$ is large, we momentarily 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 $C_k := \mathbb{E}\left[\mathbb{1}_C \,|\, \widehat{\mathrm{Msg}}_{\leq k}\right]$ (that is, C_k is the projection of the event C on the information held by $\widehat{\mathrm{Msg}}_{\leq k}$). Since, under the conditioning, P is a small-jumps party, it holds that the adversary corrupts P with probability ${}^{\lambda^2}/{}_{\sqrt{n}}$, i.e., $\mathbb{E}\left[\mathbb{1}_C\right] = \mathbb{P}\left[C\right] = {}^{\lambda^2}/{}_{\sqrt{n}}$. Thus by Doob's maximal inequality (see Lemma 3.8), it holds that

$$\mathbb{P}\left[\neg H\right] = \mathbb{P}\left[\sup_k C_k \geq 16 \cdot {}^{\lambda^2}/{}_{\sqrt{n}}\right] \leq {}^{1}/{}_{16} \tag{35}$$

Back to the regular probability space, we deduce that

$$\mathbb{P}\left[L \wedge H \,|\, S\right] \geq \mathbb{P}\left[L \,|\, S\right] - \mathbb{P}\left[\neg H \,|\, S\right] \geq {}^{1}/{}_{8} - {}^{1}/{}_{16} = {}^{1}/{}_{16} \tag{36}$$

where $\mathbb{P}\left[L \,|\, \mathsf{P} \in \mathrm{SmallParties}\right] \geq {}^{1}/{}_{8}$ holds by assumption. We next bound $D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}}\big|_{S \wedge C} \,\|\, \widehat{\mathrm{Msg}}\big|_{S}\right)$. Compute,

$$\begin{aligned}
& D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}}\big|_{S \wedge C} \,\|\, \widehat{\mathrm{Msg}}\big|_{S}\right) && (37)\\
&= \sum_{i=1}^{\ell} \mathop{\mathbb{E}}_{\mathrm{msg}_{<i} \leftarrow \widehat{\mathrm{Msg}}_{<i}|_{S \wedge C}} \left[D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge C} \,\|\, \widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}}\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathrm{msg} \leftarrow \widehat{\mathrm{Msg}}|_{S \wedge C}} \Big[\sum_{i \in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge C} \,\|\, \widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i}}\right)\Big] && (38)\\
&\leq \mathop{\mathbb{E}}_{\mathrm{msg} \leftarrow \widehat{\mathrm{Msg}}|_{S \wedge C}} \Big[\sum_{i \in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\left(\widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge C} \,\|\, \widehat{\mathrm{Msg}}_i\big|_{\widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge \neg C}\right)\Big] && (39)\\
&\leq \mathop{\mathbb{E}}_{\mathrm{msg} \leftarrow \widehat{\mathrm{Msg}}|_{S \wedge C}} \Big[\sum_{i \in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} D_{\mathrm{KL}}\left(\mathrm{Biased}_{\sqrt{n}}^{\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i},\,\cdot)}\left(\mathrm{Msg}_i^{\Pi}\,\big|_{\mathrm{Msg}_{<i}^{\Pi} = \mathrm{msg}_{<i}}\right) \,\|\, \mathrm{Msg}_i^{\Pi}\,\big|_{\mathrm{Msg}_{<i}^{\Pi} = \mathrm{msg}_{<i}}\right)\Big] \\
& && (40)\\
&\leq \mathop{\mathbb{E}}_{\mathrm{msg} \leftarrow \widehat{\mathrm{Msg}}|_{S \wedge C}} \Big[\sum_{i \in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} 2n \cdot \mathrm{Var}\left[\mathrm{jump}^{\Pi}\left(\mathrm{Msg}_{\leq i}^{\Pi}\right) \mid \mathrm{Msg}_{<i}^{\Pi} = \mathrm{msg}_{<i}\right]\Big] && (41)\\
&\leq 2n \cdot \mathbb{E}\left[{}^{2}/{}_{\lambda n}\right] \leq {}^{4}/{}_{\lambda}.
\end{aligned}$$

The first equality follows by chain-rule of KL Divergence. Equality (38) follows by the fact that the distribution of messages not sent by P is not affected by conditioning on C. Inequality (39) follows by Fact 3.2 ($\widehat{\mathrm{Msg}}_i$ is a convex combination of $\widehat{\mathrm{Msg}}_i\big|_C$ and $\widehat{\mathrm{Msg}}_i\big|_{\neg C}$). Inequality (40) follows by construction (see Step 4b of Algorithm 4.8), Inequality (41) follows from Lemma 4.7(2), and the penultimate inequality hold since, by assumption, the protocol Π is n-normal.

By Equation (37) and the Pinsker bound (see Fact 3.3), it holds that $\mathsf{SD}\Big(\widehat{\mathrm{Msg}}\big|_{S\wedge C}, \widehat{\mathrm{Msg}}\big|_S\Big) \leq {}^2\!/\!\sqrt{\lambda}$. Hence, by Equation (36) and the data-processing inequality of statistical distance (Fact 3.1), it holds that (for large enough n)

$$\mathbb{P}\big[H \wedge L \,|\, S \wedge C\big] \geq {}^1\!/_{32} \tag{42}$$

In other words, even when P is a corrupted small-jumps party, it still encounters large conditional variance and biases all jumps it encounters. Thus, all that is left to do is analyze the expectancy of P's increments under this conditioning. Let $\widetilde{\mathrm{Msg}}$ denote the distribution $\widehat{\mathrm{Msg}}\big|_{S\wedge C}$, and for $\mathrm{msg} \in \mathrm{Supp}\Big(\widehat{\mathrm{Msg}}\Big)$ let $\mathbb{1}_H(\mathrm{msg})$ be *the value* of $\mathbb{1}_H$ determined by $\widehat{\mathrm{Msg}} = \mathrm{msg}$. Compute,

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{i\in I_{\mathsf{P}}} X_i \,|\, S \wedge C\Big] \\
&= \mathbb{E}\Big[\sum_{i\in I_{\mathsf{P}}} X_i \,|\, S \wedge C\Big] = \mathop{\mathbb{E}}_{\mathrm{msg}\leftarrow\widetilde{\mathrm{Msg}}}\Big[\sum_{i\in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} \mathbb{E}\big[X_i \,|\, \widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge C\big]\Big] \\
&\geq \mathop{\mathbb{E}}_{\mathrm{msg}\leftarrow\widetilde{\mathrm{Msg}}}\Big[\mathbb{1}_H(\mathrm{msg}) \cdot \sum_{i\in \mathcal{I}dx_{\mathsf{P}}(\mathrm{msg})} \mathbb{E}\big[\mathrm{Biased}_{\sqrt{n}}\big(Y_i \,|\, \widehat{\mathrm{Msg}}_{<i}\big) \,|\, \widehat{\mathrm{Msg}}_{<i} = \mathrm{msg}_{<i} \wedge C\big]\Big] &(43)\\
&= \mathbb{E}\Big[\mathbb{1}_H \cdot \sum_{i\in I_{\mathsf{P}}} \sqrt{n} \cdot \mathrm{Var}\big[Y_i \,|\, \widehat{\mathrm{Msg}}_{<i}\big] \,|\, S \wedge C\Big] \geq \sqrt{n} \cdot \mathbb{E}\big[\mathbb{1}_H \cdot {}^1\!/_{8\lambda n} \cdot \mathbb{1}_L \,|\, S \wedge C\big] &(44)\\
&= {}^1\!/_{8\lambda\sqrt{n}} \cdot \mathbb{P}\big[H \wedge L \,|\, S \wedge C\big] \geq {}^1\!/_{8\lambda\sqrt{n}} \cdot {}^1\!/_{32} = {}^1\!/_{256\lambda\sqrt{n}}. &(45)
\end{aligned}$$

Inequality (43) follows by the definition of A (see Step 4b of Algorithm 4.8) and Claim 4.9 (Y_i is independent of C conditioned on $\widehat{\mathrm{Msg}}_{<i}$). Equality (44) follows from Lemma 4.7(1). The penultimate inequality follows by a point-wise inequality, and Inequality (45) follows by Equation (42). □

4.1.3 Bounding KL-Divergence between Attacked and Honest Executions, Proving Claim 4.11

In this section we show that the KL Divergence between $\widehat{\mathrm{Msg}}$ and Msg^{Π} is small. In particular we prove the following claim,

Claim 4.16 (Restatement of Claim 4.11). $D_{\mathrm{KL}}\big(\widehat{\mathrm{Msg}} \,\|\, \mathrm{Msg}^{\Pi}\big) \leq 16^3\lambda^3$.

The core of the proof relies on Lemma 4.7(3), which states that corrupting some party with probability p and then biasing its message according to Biased^f_α, is equivalent to biasing this message according to $\mathrm{Biased}^f_{p\alpha}$. This fact yields the following observation:

Claim 4.17. *For any $i \in [\ell]$ and partial transcript $\mathrm{msg}_{<i} \in \mathrm{Supp}(\widehat{\mathrm{Msg}}_{<i})$, it holds that*

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

Proof. Let $\mathsf{P} := \mathsf{party}(\mathrm{msg}_{<i})$ be the party sending the i^{th} message. If P is NonRobust, we are done since its messages are left unchanged (it is never corrupted). Else, we separately deal with the case that P is a small-jumps party and a large-jump party (with respect to the partial transcript $\mathrm{msg}_{<i}$, which suffices to determine the type of the party).

P **is a small-jumps party.** Conditioned on $\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}$, the i^{th} message is altered from its honest (conditional) distribution according Π, with probability $p \leq {}^{16\lambda^2}/\!\sqrt{n}$ (see Step 4b of Algorithm 4.8). If the i^{th} message is altered, it is sampled according to $\text{Biased}_{\sqrt{n}}^{\text{jump}^{\Pi}(\text{msg}_{<i},\,\cdot)}\big(\text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)$. By Lemma 4.7(3), $\widehat{\text{Msg}}_i|_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}}$ is distributed like $\text{Biased}_{p\cdot\sqrt{n}}^{\text{jump}^{\Pi}(\text{msg}_{<i},\,\cdot)}\big(\text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)$. Hence, by Lemma 4.7(2)

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

P **is a large-jump party.** Conditioned on $\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}$, the i^{th} message is altered from its honest (conditional) distribution according Π, with probability $\lambda^2\cdot\sqrt{v}$ where $v := \text{Var}\big[\text{jump}^{\Pi}\big(\text{Msg}_{\leq i}^{\Pi}\big) \mid \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}\big]$. If the i^{th} message is altered, it is sampled according to $\text{Biased}_{1/\sqrt{v}}^{\text{jump}^{\Pi}(\text{msg}_{<i},\,\cdot)}\big(\text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)$. Hence, by Lemma 4.7(3), $\widehat{\text{Msg}}_i|_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}}$ is distributed like $\text{Biased}_{\lambda^2\cdot\sqrt{v}\cdot 1/\sqrt{v}}^{\text{jump}^{\Pi}(\text{msg}_{<i},\,\cdot)}\big(\text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)$. By Lemma 4.7(2), we conclude that

$$D_{\text{KL}}\big(\widehat{\text{Msg}}_i|_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}} \,\|\, \text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big) \leq 2\cdot\lambda^4\cdot \text{Var}\big[\text{jump}^{\Pi}\big(\text{Msg}_{\leq i}^{\Pi}\big) \mid \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}\big].$$

□

Proof of Claim 4.11. Let RobustJumps(msg), a set, denote the *value* of RobustJumps determined by $\widehat{\text{Msg}} = \text{msg}$. Compute,

$$\begin{aligned}
&D_{\text{KL}}\big(\widehat{\text{Msg}}_{\leq \ell} \,\|\, \text{Msg}_{\leq \ell}^{\Pi}\big)\\
&= \sum_{i=1}^{\ell} \mathop{\mathbb{E}}_{\text{msg}_{<i}\leftarrow\widehat{\text{Msg}}_{\leq i}}\Big[D_{\text{KL}}\big(\widehat{\text{Msg}}_i|_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}} \,\|\, \text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)\Big] && (46)\\
&= \mathop{\mathbb{E}}_{\text{msg}\leftarrow\widehat{\text{Msg}}}\Big[\sum_{i\in\text{RobustJumps(msg)}} D_{\text{KL}}\big(\widehat{\text{Msg}}_i|_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}} \,\|\, \text{Msg}_i^{\Pi}\,|_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}\big)\Big] && (47)\\
&\leq \mathop{\mathbb{E}}_{\text{msg}\leftarrow\widehat{\text{Msg}}}\Big[\sum_{i\in\text{RobustJumps(msg)}} 16^3\lambda^4\cdot \text{Var}\big[\text{jump}^{\Pi}\big(\text{Msg}_{\leq i}^{\Pi}\big) \mid \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}\big]\Big] && (48)\\
&= 16^3\lambda^4\cdot\mathbb{E}\Big[\sum_{i\in\text{RobustJumps}} \text{Var}\big[\text{jump}^{\Pi}\big(\text{Msg}_{\leq i}^{\Pi}\big) \mid \text{Msg}_{<i}^{\Pi} = \widehat{\text{Msg}}_{<i}\big]\Big]\\
&= 16^3\lambda^4\cdot\mathbb{E}\Big[\sum_{i\in\text{RobustJumps}} \text{Var}\big[Y_i \mid \widehat{\text{Msg}}_{<i}\big]\Big] && (49)\\
&\leq 16^3\lambda^3. && (50)
\end{aligned}$$

Equality (46) follows by chain rule of KL Divergence, Equality (47) follows since non-RobustJumps are not corrupted (as they belong to NonRobust), Inequality (48) follows by Claim 4.17, Inequality (49) follows by definition of Y_i (see Claim 4.9) and inequality (50) follows by Claim 4.10. □

4.2 Biasing Arbitrary Robust Coin Flip

In this section we use the attack on normal robust protocols proved to exists 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 Lemma 4.5, yields an attack of essentially the same quality on the original (non normal) protocol, thus proving Theorem 4.3.

We start with defining the normal form variant of a coin-flipping protocol. Let Π be an t-party, ℓ-round, full-information coin-flipping protocol. Its ($r := 2\ell \cdot t + 1$)-party, ℓ-round, n-normal variant $\widetilde{\Pi}$ is defined as follows:

Protocol 4.18 (n-normal protocol $\widetilde{\Pi}$)**.**

- *For each party* P *of the protocol* Π*, the protocol* $\widetilde{\Pi}$ *has* 2ℓ *parties* $\mathsf{P}_1^{\mathsf{small}}, \ldots, \mathsf{P}_\ell^{\mathsf{small}}$ *and* $\mathsf{P}_1^{\mathsf{large}}, \ldots, \mathsf{P}_\ell^{\mathsf{large}}$*. In addition,* $\widetilde{\Pi}$ *has an special party named* NonRobust*.*
- *For each party* P *of* Π*, start three counters* $L_{\mathsf{P}} = S_{\mathsf{P}} = 1$*, and* $A_{\mathsf{P}} = 0$*.*
- *In rounds* $i = 1$ *to* ℓ*, the protocol is defined as follows.*
  1. *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.*
  2. *Let* Q_i *be the distribution* $\mathrm{Msg}_i^{\Pi}|_{\mathrm{Msg}_{<i}^{\Pi}=\mathrm{msg}_{<i}}$ *and let* $v_i := \mathrm{Var}\big[\mathrm{jump}^{\Pi}(\mathrm{msg}_{<i}, Q_i)\big]$*.*
  3. *Set* P' *(the "active" party) as follows:*
     (a) **If** $\mathrm{Supp}\big(\mathrm{jump}^{\Pi}\big(\mathrm{msg}_{<i,} Q_i\big)\big) \cap (-\infty, {}^1/\lambda\cdot\sqrt{n}] \neq \emptyset$*, set* P' *to* NonRobust*.*
     (b) **Else, If** $v_i \geq {}^1/\lambda n$*, set* P' *to* $\mathsf{P}_{L_{\mathsf{P}}}^{\mathsf{large}}$*, and update* $L_{\mathsf{P}} = L_{\mathsf{P}} + 1$*.*
     (c) **Else, If** $v_i < {}^1/\lambda n$*:*
         Set P' *to* $\mathsf{P}_{S_{\mathsf{P}}}^{\mathsf{small}}$ *and update* $A_{\mathsf{P}} = A_{\mathsf{P}} + v_i$
         If $A_{\mathsf{P}} > {}^1/\lambda n$*:*
         – *Set* $S_{\mathsf{P}} = S_{\mathsf{P}} + 1$*.*
         – *Set* $A_{\mathsf{P}} = 0$*.*
  4. P' *sends the* i^{th} *message, as* P *would in* Π *given the partial transcript* $\mathrm{msg}_{<i}$*.*

Claim 4.19. *Assume* Π *is a* t*-party full-information coin-flipping protocol, then* $\widetilde{\Pi}$ *is a an* n*-normal full-information coin-flipping protocol.*

Proof. We handle each of the conditions independently,

Single non-robust party: Step (3a) properly handles jumps that should belong to NonRobust.

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

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

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

□

Proving Theorem 4.3. Given the above tool and Lemma 4.5, 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 4.18. By Lemma 4.5 and Claim 5.4, there exits a $O(\sqrt{n} \cdot \log n)$-adaptive adversary $\widetilde{\mathsf{A}}$ for $\widetilde{\Pi}$ such that $\mathbb{E}\big[\widetilde{\Pi}_{\widetilde{\mathsf{A}}}\big] \geq 1 - O(\varepsilon)$. 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 4.18. It is clear that $\mathbb{E}\big[\Pi_{\mathsf{A}}\big] = \mathbb{E}\big[\widetilde{\Pi}_{\widetilde{\mathsf{A}}}\big] \geq 1 - O(\varepsilon)$. In addition, since by construction the parties in $\widetilde{\Pi}$ are refinements of the parties in Π, corrupting k parties in $\widetilde{\Pi}$ is translated to corrupting at most k parties of Π. We conclude that A is the desired $O(\sqrt{n} \cdot \log n)$-adaptive □

5 Biasing Arbitrary Coin Flip

In this section we use the attack on robust protocol described in Section 4 to prove our main result: an adaptive attack on any full-information coin-flipping protocols.

We fix $\ell \in \mathbb{N}$ (the number of parties of the robust protocol), and make use of the following constants.

Notation 5.1. *Let* $\varepsilon := 1/\sqrt[50]{\log\log n}$, $\lambda := 1/100\varepsilon^5$ *and* $\delta := 1/\log^2 n$.

This main result of our paper is given below.

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

Our proof make use of the following deterministic one-shot (modifies at most a single message) adversary that take advantage of large negative jumps for biasing the protocol's output towards 0.

Algorithm 5.3 (One-shot adversary B for protocol Γ)**.**

```
For $i = 1$ to $\mathrm{NumMsgs}(\Gamma)$:
    1. Let $\mathrm{msg}_{<i}$ be the messages sent in the previous rounds, and let $\mathsf{P}$ be the the party about to send the $i^{\mathrm{th}}$ message.
    2. If no message was corrupted before, and $\mathcal{M}_i := \mathrm{Supp}\big(\mathrm{jump}^{\Gamma}\big(\mathrm{Msg}_i^{\Gamma}\big) \mid \mathrm{Msg}_{<i}^{\Gamma} = \mathrm{msg}_{<i}\big) \cap (-\infty, -1/\lambda\cdot\sqrt{n}] \neq \emptyset$:
        Instruct $\mathsf{P}$ to broadcast $m$ as it next message, for some $m \in \mathcal{M}_i$.
```

The proof of the following fact is immediate.

Claim 5.4. *For any n-party full-information coin-flipping protocol Γ, it holds that:*

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

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

Proof of Theorem 5.2. For $t := {}^{\sqrt{n}}\!/\!{}_{\lambda\delta} = O\left(\sqrt{n}\cdot\log^3 n\right)$, consider the protocols $\Pi^0,\ldots,\Pi^t$ recursively defined by $\Pi^0 := \Pi$ and $\Pi^{i+1} := {\Pi^i}_{\mathsf{B}}$, where B is according to Algorithm 5.3. If $\mathbb{E}\left[\Pi^t\right] < \varepsilon$, then by Proposition 3.17 there exists a t-adaptive adversary that biases Π's output to less than ε (the composition of all intermediate adversaries), and we are done. Else, by Claim 5.4 there exists $i \in [t]$ such that for $\tau := \Pi^i$ it holds that

$$\mathbb{P}\left[\exists i\colon \operatorname{Supp}\left(\mathrm{jump}^{\tau}\left(\mathrm{Msg}^{\tau}_{\leq i}\right) \mid \mathrm{Msg}^{\tau}_{<i}\right) \cap \left(-\infty, {}^{1}\!/\!{}_{\lambda\cdot\sqrt{n}}\right] \neq \emptyset\right] \leq \delta.$$

Hence by Theorem 4.3, there exists an $O(\sqrt{n}\cdot\log n)$-adaptive adversary A (which we assume without loss of generality to be deterministic, see Proposition 3.14) such that

$$\mathbb{E}\left[\tau_{\mathsf{A}}\right] \geq 1 - O(\varepsilon).$$

Denote by C the (deterministic) i-adaptive adversary according to Definition 3.15 (the composition of all intermediate adversaries) such that $\Pi_{\mathsf{C}} \equiv \Pi^i$. Let $\mathsf{A}\circ\mathsf{C}$ be the $O\left(\sqrt{n}\cdot\log^3 n\right)$-adaptive adversary according to Definition 3.15, by Proposition 3.17 it holds that $\mathbb{E}\left[\Pi_{\mathsf{A}\circ\mathsf{C}}\right] = \mathbb{E}\left[\tau_{\mathsf{A}}\right] \geq 1 - O(\varepsilon)$, concluding the proof. $\square$

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. Impossibility of black-box reduction from non-adaptively to adaptively secure coin-flipping. Technical Report TR00-39, Electronic Colloquium on Computational Complexity, 2000.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] I. Haitner and Y. Karidi-Heller. A tight lower bound on strongly adaptively secure full-information coin flip and applications to data poisining. Work in progress, 2020.

[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.