

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

Iftach Haitner\*<sup>†</sup>

Yonatan Karidi-Heller \*

May 4, 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 *during* the protocol execution. [Ben-Or and Linial](#) proved that the  $\ell$ -party majority protocol is resilient to  $o(\sqrt{\ell})$  corruptions (ignoring log factors), and conjectured this is a tight upper bound for any  $\ell$ -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, 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 (even single-bit) protocols was left completely open.

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

---

\*School of Computer Science, Tel Aviv University. Emails: [iftachh@cs.tau.ac.il](mailto:iftachh@cs.tau.ac.il), [karidiheller@mail.tau.ac.il](mailto:karidiheller@mail.tau.ac.il). Research supported by ERC starting grant 638121.

<sup>†</sup>Director of the Check Point Institute for Information Security.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Our Results . . . . .	1
1.2	Related Work . . . . .	2
1.2.1	Full-Information Coin Flip . . . . .	2
1.2.2	Computationally Secure Coin Flip . . . . .	2
<b>2</b>	<b>Our Technique</b>	<b>4</b>
2.1	Attacking Single-Turn Coin Flip . . . . .	5
2.2	Attacking Many-Turn Coin Flip . . . . .	7
2.3	Attacking Non-Robust Coin Flip . . . . .	9
<b>3</b>	<b>Preliminaries</b>	<b>10</b>
3.1	Notations . . . . .	10
3.2	Distributions and Random Variables . . . . .	10
3.3	Martingales . . . . .	11
3.4	Full-Information Coin Flip . . . . .	11
3.4.1	Adaptive Adversaries . . . . .	12
3.5	Useful Inequalities . . . . .	13
<b>4</b>	<b>Biasing Robust Coin Flip</b>	<b>13</b>
4.1	Biasing Normal Robust Coin Flip . . . . .	14
4.1.1	Coupling $X_i$ and $Q_i$ . . . . .	20
4.1.2	Bounding $Y$ 's Conditional Variance . . . . .	21
4.1.3	Bounding KL-Divergence between Attacked and Honest Executions . . . . .	25
4.2	Biasing Arbitrary Robust Coin Flip . . . . .	26
<b>5</b>	<b>Biasing Arbitrary Coin Flip</b>	<b>28</b>

# 1 Introduction

In a distributed (also known as, collective) *coin-flipping* protocol, Blum [8], the parties try to output a common (close to) uniform bit, even when some adversarially chosen parties try to bias the output. The adversary is assumed to be Byzantine—once it corrupts a party, it completely controls it and can send arbitrary messages on its behalf. In the *full-information* variant of such protocols, Ben-Or and Linial [7], the parties communicate (solely) over a (single) broadcast channel, and the adversary is assumed to be computationally *unbounded*. For such protocols, two types of (Byzantine, unbounded) 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 [7] proved that the  $\ell$ -party majority protocol is resilient to  $o(\sqrt{\ell})$  corruptions (ignoring poly-logarithmic factors in  $\ell$ ), and conjectured that this is a tight upper bound for any  $\ell$ -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 [19], single-turn (arbitrary message length) *symmetric* protocols Goldwasser, Kalai, and Park [15], and recently for single-turn protocols, Tauman Kalai, Komargodski, and Raz [24]. Yet, the question for many-turn (even single-bit) protocols was left completely open.

## 1.1 Our Results

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

**Theorem 1.1** (Biasing full-information coin-flipping protocols, informal). *For any  $\ell$ -party full-information coin-flipping protocol, there exists  $b \in \{0, 1\}$  and an (unbounded) adversary that by adaptively corrupting  $\sqrt{\ell}$  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  $\ell$ -party majority protocol [7]. 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  $\ell$ -party (single-turn) protocol that is resistant to  $\Theta(\ell)$  corruptions trying to bias its outcome towards one.<sup>1</sup>

---

<sup>1</sup>Consider the  $\ell$ -party  $\ell$ -round protocol in which each party broadcasts a  $(1/\ell, 1 - 1/\ell)$ -biased bit (i.e., equals zero with probability  $1/\ell$ ) and the protocol output is set to the AND of these bits. It is clear that the protocol expected outcome is  $(1 - 1/\ell)^\ell \approx 1/e$  (can be made  $1/2$  by slightly changing the distribution), and that even  $\ell/2$  adaptive corruptions cannot change the protocol outcome to a value larger than  $(1 - 1/\ell)^{\ell/2} \approx \sqrt{1/e}$ .

## 1.2 Related Work

### 1.2.1 Full-Information Coin Flip

We recall the main known results for  $\ell$ -party full-information coin-flipping protocols.

**Adaptive adversaries.** In the following we ignore poly-logarithmic factors in  $\ell$ .

**Upper bounds (protocols).** Ben-Or and Linal [7] proved that the majority protocol is resilient to  $o(\sqrt{\ell})$  corruptions.

**Lower bounds (attacks).** Lichtenstein, Linal, and Saks [19] proved that no single-bit, single-turn protocol is resilient to  $\Omega(\sqrt{n})$  adaptive corruptions (hence, majority is optimal for such protocols). Dodis [12] 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, Kalai, and Park [15] 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 [24] 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 [13] presented an *efficient* and optimal strongly adaptive attack on protocols of certain properties (e.g., public coins). On a related note, Kalai and Komargodski [18] showed that for any  $\ell$ -party  $n$ -round coin-flipping protocol there exists a related  $\ell$ -party  $n$ -round protocol of the same communication pattern, output distribution and security guarantees, but of message length  $\text{polylog}(\ell, n)$ .

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

**Upper bounds (protocols).** Ben-Or and Linal [7] presented a protocol that tolerates  $O(\ell^{0.63})$  corrupted parties (an improvement on the  $\sqrt{\ell}$  corrupted parties it takes to bias the majority protocol). Ajtai and Linal [1] presented a protocol that tolerates  $O(\ell/\log^2 \ell)$  corruptions. Saks [23] presented a protocol that tolerates  $O(\ell/\log \ell)$  corruptions. The protocol of [23] was later improved by Alon and Naor [3] to tolerate a constant fraction of corrupted parties. Shortly afterwards, Boppana and Narayanan [9] presented an optimal protocol resilient to  $(1/2 - \delta)\ell$  corruptions for any  $\delta > 0$ .

**Lower bounds (attacks).** Kahn, Kalai, and Linal [17] proved that no single-bit single-round protocol can tolerate  $\Omega(\ell/\log \ell)$  corruptions. Russell, Saks, and Zuckerman [22] proved that a protocol tolerating  $\Omega(\ell)$  corruptions is either many-bit or has  $\Omega(1/2 - o(1)) \cdot \log^*(\ell)$  rounds.

### 1.2.2 Computationally Secure Coin Flip

There is a rich literature on coin-flipping protocols secure against polynomially bounded adversaries. Several such models were considered, where the most relevant model to our settings are the so-called *fair* coin-flipping protocols: the adversary is polynomially bounded and can corrupt all but one party, the corrupted parties are statically chosen, and the honest parties must output a bit as their outcome (i.e., are not allowed to abort even if malicious party is “detected”).

**Upper bounds.** The following results hold under the proper hardness assumptions (existence of one-way functions or of oblivious transfer protocols). Blum [8] presented a two-party two-round coin-flipping protocol of (maximal) bias  $1/4$ . Awerbuch, Blum, Chor, Goldwasser, and Micali [4] presented an  $\ell$ -party  $n$ -round protocol with bias  $O(\ell/\sqrt{n})$  (the two-party case appears also in Cleve [11]). Moran, Naor, and Segev [21] solved the two-party case (matched the lower bound for such protocols, see below) by giving a two-party  $n$ -round protocol with bias  $\Theta(1/n)$ . Haitner and Tsfadia [16] solved the three-party case (up to poly-logarithmic factor) by presenting a three-party protocol with bias  $O(\text{polylog}(n)/n)$ . Buchbinder, Haitner, Levi, and Tsfadia [10] presented an  $\ell$ -party  $n$ -round with bias  $\tilde{O}(\ell^3 2^\ell / n^{\frac{1}{2} + \frac{1}{2^\ell - 1 - 2}})$ . In particular, their protocol for four parties has bias  $\tilde{O}(1/n^{2/3})$ , and for  $\ell = \log \log n$  their protocol has bias smaller than Awerbuch et al. [4].

For the case where less than  $2/3$  of the parties are corrupt, Beimel, Omri, and Orlov [5] showed an  $\ell$ -party  $n$ -round protocol with bias  $2^{2^k}/n$ , tolerating up to  $t = (\ell + k)/2$  corrupt parties. Alon and Omri [2] showed an  $\ell$ -party  $n$ -round protocol with bias  $\tilde{O}(2^{2^\ell}/n)$ , tolerating up to  $t$  corrupted parties, for constant  $\ell$  and  $t < 3\ell/4$ .

**Lower bounds.** Cleve [11] proved that for any  $n$ -message two-party coin-flipping protocol, there exists an efficient adversary that can bias the output by  $\Omega(1/n)$ . The bound extends to the multi-party case (with no honest majority) via a simple reduction. Beimel, Haitner, Makriyannis, and Omri [6] showed that *any*  $n$ -round  $\ell$ -parties coin-flipping protocol with  $\ell^k > n$ , for some  $k \in \mathbb{N}$ , can be biased by  $1/(\sqrt{n} \cdot (\log n)^k)$ . Ignoring logarithmic factors, this means that if the number of parties is  $n^{\Omega(1)}$ , the majority protocol of Awerbuch et al. [4] is optimal.

## Open Questions

In this work we show that the outcome of *any*  $\ell$ -party full-information coin-flipping protocol can be biased either to  $o(1)$  or to  $1 - o(1)$ , using  $\sqrt{\ell}$  corruptions. However, the above  $o(1)$  stands for  $1/\log \log(\ell)$ , and it remains an intriguing question whether it can be pushed to  $2^{-\text{polylog}(\ell)}$  as can be achieved, for instance, when attacking the  $\ell$ -party majority protocol. Such attacks are known for *uniform* single-bit single-turn protocols (a secondary result of [24]) and for *strongly adaptive* attackers against single-turn protocols [13].

A second question is when the outcome of a protocol can be biased to *both* directions. While there are (single-turn, single-bit) protocols that are resistant to such double-sided attacks (see Footnote 1), [24] proved that a uniform single-bit single-turn protocols can be always biased to both direction. Yet, the exact class of protocols and attack models for which such double-sided attacks always exist is an interesting open question. In particular, it is plausible that double-sided attacks are always achievable by strongly adaptive adversaries.

On a different thread, Mahloujifar, Diochnos, and Mahmoody [20] drew a very interesting connection between (adaptively) attacking coin-flipping protocols and *poisoning attacks on robust learners*. They showed that in some settings, the process of adaptively perturbing the training data in order to fail the online learner, can be interpreted as an adaptive attack on a related full-information coin flip. They then used ideas from [24] to build, for some distributions, a poisoning attack that by adaptively perturbing  $\sqrt{n}$  points (out of  $n$  samples) fails *any* online learner for the given distribution. An attack on larger set of distributions was given in [13] by facilitating their strongly adaptive attack on one-turn coin-flipping protocols (see Section 1.2). The above attacks are limited to perturbing  $\sqrt{n}$  points (out of  $n$  samples). Assuming the samples are coming from

$\ell \ll n$  sources, can we facilitate our attack to fail the learner by perturbing only  $\sqrt{\ell}$  sources?

## 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  $\Pi$  be an  $\ell$ -party,  $n$ -message full-information coin-flipping protocol. We prove that one can either bias the expected outcome of  $\Pi$  to less than  $\varepsilon := 1/\log\log(\ell)$ , or to larger 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  $\text{Msg} = (\text{Msg}_1, \dots, \text{Msg}_n)$  denote the protocol transcript (i.e., parties’ messages) in a random all-honest execution of  $\Pi$ . For  $\text{msg} \in \text{Supp}(\text{Msg})$ , let  $\Pi(\text{msg})$  denote the final outcome of the execution described by  $\text{msg}$ , and for  $\text{msg}_{\leq i} \in \text{Supp}(\text{Msg}_{\leq i} := (\text{Msg}_1, \dots, \text{Msg}_i))$  let  $\Pi(\text{msg}_{\leq i}) := \mathbb{E}[\Pi(\text{Msg}) \mid \text{Msg}_{\leq i} = \text{msg}_{\leq i}]$  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  $i^{\text{th}}$  message) as the  $i^{\text{th}}$  jump in the protocol execution.

Our attack exploits these gaps in a different way than what previous attacks did. First, it mainly cares about the (conditional) variance of the jumps, rather than their support. 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. Similar to some of the previous works, 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.

We start by describing an attack on *robust* protocols: for some  $b \in \{0, 1\}$ , the protocol has *no*  $1/\sqrt{\ell}$  jumps towards  $b$ .<sup>2</sup> In Section 2.3, we explain how to attack arbitrary (non robust) protocols. For correctness, we focus on robust protocols with respect to  $b = 0$ . That is, we assume that (for simplicity, with probability one)

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

We start by describing an attack on single-turn protocols (i.e.,  $\ell = n$ ), and in Section 2.2 extend it to the more complicated case of many-turn protocols. In the following for  $\text{msg}_{< i} \in \text{Supp}(\text{Msg}_{\leq i})$ ,

---

<sup>2</sup>An almost accurate example for a (bi-directional) robust protocol is the *majority* protocol of  $n \geq \ell$  rounds: 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 apart from the very last jumps, the absolute value of each jump is order of  $1/\sqrt{n} \leq 1/\sqrt{\ell}$ .

let  $\text{jump}(\text{msg}_{\leq i}) := \Pi(\text{msg}_{\leq i}) - \Pi(\text{msg}_{< i})$ , i.e., the difference in expected outcome induced by the  $i^{\text{th}}$  message.

## 2.1 Attacking Single-Turn Coin Flip

We use the following procedure for biasing a single message of the protocol.

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

$$\mathbb{P}[\text{Biased}_t^f(P) = x] := \mathbb{P}[P = x] \cdot (1 + t \cdot f(x))$$

It is easy to verify that  $\text{Biased}_t^f(P)$  is indeed a probability distribution. Note that  $\mathbb{E}_{\text{Biased}_t^f(P)}[f]$  is non negative, and the larger  $t$  is, the larger the expectation is (i.e., the bias is more significant). Given the above definition, our attacker is defined as follows:

**Algorithm 2.2** (Single-turn attacker).

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

1. *Let  $\text{msg}_{< i}$  be the previously sent messages, let  $Q_i := \text{Msg}_i |_{\text{Msg}_{< i} = \text{msg}_{< i}}$ , let  $\text{jump}_i := \text{jump}(\text{msg}_{< i}, \cdot)$  and let  $v_i := \text{Var}[\text{jump}_i(Q_i)]$ .*

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

*Else, corrupt the  $i^{\text{th}}$  party with probability  $1/\varepsilon^3 \cdot 1/\sqrt{\ell}$ . If corrupted, instruct it to send the next message according to  $\text{Biased}_{\sqrt{\ell}}^{\text{jump}_i}(Q_i)$ .*<sup>3</sup>

That is, a message (party) is corrupted with probability that is proportional to the square root of the (conditional) variance it induces on the expected outcome of  $\Pi$ , but at least  $1/\sqrt{\ell}$ . If corrupted, the message is modified so that the change it induces on the expected outcome of  $\Pi$  is biased towards one, where the bias is proportional to the inverse of the square root of the above variance (up to  $\sqrt{\ell}$ ).<sup>4</sup> In the following we argue that the attacker indeed biases the outcome of  $\Pi$  to  $1 - \varepsilon$ , and that the expected number of corruptions is  $O(\sqrt{\ell}/\varepsilon^3)$ . Thus, a Markov bound yields the existence of the required attacker.

Let  $\widehat{\text{Msg}} = (\widehat{\text{Msg}}_1, \dots, \widehat{\text{Msg}}_n)$  be the message distribution induced by the above attack, and consider the Doob *sub*-martingale  $S = (S_0, \dots, S_n)$  with respect to  $\widehat{\text{Msg}}$ , induced by the outcome of  $\Pi$ . That is,  $S_i := \Pi(\widehat{\text{Msg}}_{\leq i})$ . By definition,  $S_0 = \mathbb{E}[\Pi(\text{Msg})] = 1/2$  and  $S_n \in \{0, 1\}$ . For  $i \in [n]$ , let  $Q_i$  be the value of  $Q_i$  in the attack execution, determined by  $\widehat{\text{Msg}}_{i-1}$ , and let  $Y_i = \text{jump}(\widehat{\text{Msg}}_{< i}, q)$  for  $q \sim Q_i$ . Since  $Q_i$  is the  $i^{\text{th}}$  message distribution of the unbiased protocol  $\Pi$  (determined by  $\widehat{\text{Msg}}_{< i}$ ), it holds that  $\mathbb{E}[Y_i | \widehat{\text{Msg}}_{< i}] = 0$ . Since the  $Y_i$ 's are also independent of each other conditioned on  $\widehat{\text{Msg}}$ , the sequence  $Y = (Y_1, \dots, Y_n)$  is a *martingale difference sequence with respect to*  $(\widehat{\text{Msg}}_i, Y_i)$ . The core of our analysis lies in the following lemma.

<sup>3</sup>Since we assume Equation (1), in both cases  $\text{Biased}(Q_i)$  is indeed a valid probability distribution.

<sup>4</sup>Assuming  $\Pi$  is the  $\ell$ -round majority protocol, then (apart from the very last rounds) each  $v_i$  is of (absolute) order  $1/\ell$ . Thus, in expectation, the above attack corrupts  $1/\varepsilon^3 \cdot \sqrt{\ell}$  parties. If corrupted, the party's bit message is set to 1 with probability  $\approx 1/2 \cdot (1 + \sqrt{\ell} \cdot 1/\sqrt{\ell}) = 1$ .

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

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  $i^{\text{th}}$  party is

$$\begin{aligned} 1/\varepsilon^3 \cdot \max\left\{\sqrt{\text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]}, 1/\sqrt{\ell}\right\} &\leq 1/\varepsilon^3 \cdot \max\left\{\sqrt{\ell} \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}], 1/\sqrt{\ell}\right\} \\ &\leq 1/\varepsilon^3(\sqrt{\ell} \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}] + 1/\sqrt{\ell}). \end{aligned} \quad (2)$$

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

$$1/\varepsilon^3 \cdot \mathbb{E}\left[\sum_{i=1}^{\ell} \left(\sqrt{\ell} \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}] + 1/\sqrt{\ell}\right)\right] \leq \sqrt{\ell}(1 + 1/\varepsilon^3).$$

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

$$\mathbb{E}\left[\left(\sum_{i=1}^n Y_i\right)^2\right] = \sum_{i=1}^n \mathbb{E}[Y_i^2] = \mathbb{E}\left[\sum_{i=1}^n \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]\right] \quad (3)$$

Hence by Lemma 2.3 and a Markov bound, we deduce that

$$\mathbb{P}\left[\left|\sum_{i=1}^n Y_i\right| \geq \varepsilon\right] = \mathbb{P}\left[\left(\sum_{i=1}^n Y_i\right)^2 \geq \varepsilon^2\right] \leq \varepsilon \quad (4)$$

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

$$S_i - S_{i-1} \geq Y_i \quad (5)$$

Since, by definition,  $S_0 = 1/2$ , it follows that  $\mathbb{P}[S_n \leq 0] = \mathbb{P}[\sum_{i=1}^n Y_i \leq -1/2] \leq \varepsilon$ , and since  $S_n \in \{0, 1\}$ , we deduce that  $\mathbb{P}[S_n = 1] \geq 1 - \varepsilon$ . 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  $t \geq 0$  and function  $f : \text{Supp}(P) \mapsto [-1/\alpha, \infty)$  such that  $\mathbb{E}_P[f] = 0$ , it holds that

$$\begin{aligned} \mathbb{E}_{\text{Biased}_t^f(P)}[f] &= \sum_{x \in \text{Supp}(P)} f(x) \cdot \mathbb{P}[\text{Biased}_t^f(P) = x] \\ &= \sum_{x \in \text{Supp}(P)} f(x) \cdot \mathbb{P}[P = x] \cdot (1 + tf(P)) \\ &= \mathbb{E}_P[f \cdot (1 + tf)] = \mathbb{E}_P[tf] + \mathbb{E}_P[t \cdot f^2] = 0 + t \cdot \text{Var}_P[f]. \end{aligned} \quad (6)$$



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

$$(p \cdot \text{Biased}_t^f(P) + (1 - p) \cdot P) \equiv \text{Biased}_{p,t}^f(P) \quad (7)$$

Let  $V_i$  be the value of the variable  $v_i$  in the execution of the attack (determined by  $\widehat{\text{Msg}}_{<i}$ ), and let  $V'_i := \max\{v_i, 1/\ell\}$ . For a partial transcript  $\text{msg}_{<i} \in \text{Supp}(\widehat{\text{Msg}}_{<i})$ , applying the above observation with respect to  $P := \text{Msg}_i |_{\text{Msg}_{<i}=\text{msg}_{<i}}$ ,  $p := 1/\varepsilon^3 \cdot \sqrt{V'_i |_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}}}$ ,  $t := 1/\sqrt{V'_i |_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}}}$  and  $f := \text{jump}(\text{msg}_{<i}, \cdot)$ , yields that

$$\begin{aligned} \mathbb{E}[S_i - S_{i-1} | \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}] &= \mathbb{E}[\text{Biased}_{1/\varepsilon^3}^f(\text{Msg}_i |_{\text{Msg}_{<i}=\text{msg}_{<i}})] \\ &= 1/\varepsilon^3 \cdot \text{Var}[\text{jump}(\text{Msg}_{<i}) | \text{Msg}_{<i} = \text{msg}_{<i}] \\ &= 1/\varepsilon^3 \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}]. \end{aligned} \quad (8)$$

It follows that

$$\mathbb{E}[S_n - S_0] = 1/\varepsilon^3 \cdot \sum_{i=1}^n \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}] \quad (9)$$

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

## 2.2 Attacking Many-Turn Coin Flip

The first challenge when moving to many-turn protocols is that we can no longer decide whether to attack a message *independently* of the other messages. There are simply too many messages, and whatever such strategy one takes, it either corrupts too many parties, or biases the protocol outcome by too little. So rather, the strategy we take is to decide whether to corrupt or not, *per party*, and not per message, with the exception of “highly influential” messages. A second 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. This is achieved by partitioning, if needed, the messages of the parties into sequences of “bounded influence”, viewing each such sequence as a separate party, and deciding whether to corrupt it or not independently. In our terminology, we turn  $\Pi$  into an  $\ell$ -normal protocol.

**Normal protocols.** Let  $\text{party}(\text{msg}_{<i})$  be the identity of the party about to send the  $i^{\text{th}}$  message as described by the partial transcript  $\text{msg}_{<i}$ . A party  $P$  has *large jumps* in  $\text{msg} \in \text{Supp}(\text{Msg})$ , if  $\text{party}(\text{msg}_{<i}) = P$  for some  $i \in [n]$  such that  $\text{Var}[\text{jump}(\text{Msg}_{<i}) | \text{Msg}_{<i} = \text{msg}_{<i}] \geq 1/\ell$ . If the above does not happen *and*  $P$  *participates in the execution* (sends a message), we say that  $P$  has *small jumps* (in  $\text{msg}$ ). We assume without loss of generality, in the price of increasing the number of parties and making the identity of the parties sending the messages dynamic (i.e., function of the protocol transcript), that  $\Pi$  is  $\ell$ -normal:

**Definition 2.4** ( $\ell$ -normal protocols, informal). *Protocol  $\Pi$  is  $\ell$ -normal if the following hold:*

1. *A large-jump party (a party that has large jumps) sends only a single message.*

2. For a small-jumps party (a party that has small jumps)  $\mathsf{P}$  it holds that

$$\sum_{i \in [n]: \text{party}(\text{Msg}_{<i}) = \mathsf{P}} \text{Var}[\text{jump}(\text{Msg}_{\leq i}) \mid \text{Msg}_{<i}] \leq 2/\ell.$$

(I.e., the overall sum of conditional variances the party  $\mathsf{P}$  “has” is bounded.)

3. There are at most  $\ell$  (small-jumps) parties  $\mathsf{P}$  with

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

The advantage of assuming  $\Pi$  is  $\ell$ -normal is that for such protocols our attack either corrupts *all* messages sent by a party, or corrupts *none* of them. Our attack can be easily adapted for arbitrary protocols, and then it modifies a (typically small, non continuous) subset of the corrupted party’s messages.

**The attack.** Given the above assumptions, our many-turn attacker is defined as follows:

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

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

1. Let  $\text{msg}_{<i}$  be the previously sent messages, let  $Q_i := \text{Msg}_{\leq i} \mid \text{Msg}_{<i} = \text{msg}_{<i}$ , let  $\text{jump}_i := \text{jump}(\text{msg}_{<i}, \cdot)$ , and let  $v_i := \text{Var}[\text{jump}_i(Q_i)]$ .

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

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

That is, a large-jump party is treated like in the single-turn case. Where a small-jumps party is corrupted with probability proportional to  $1/\sqrt{\ell}$  (again like in the single-turn case). But if corrupt, *all* messages of the small-jumps party are modified.<sup>5</sup> The analysis of the above attack is similar to the single-turn case. Let  $\widehat{\text{Msg}} = (\widehat{\text{Msg}}_1, \dots, \widehat{\text{Msg}}_n)$ ,  $S = (S_0, \dots, S_n)$  and  $Y = (Y_0, \dots, Y_n)$  be as in the single-turn case. Similarly to the single turn case, the core of the proof is in the following lemma.

**Lemma 2.6.**  $\mathbb{E}[\sum_{i=1}^{\ell} \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i}]] = O(\varepsilon^3)$ .

The attacks analysis follows by the above lemma very similarly to the single-turn case. The only difference is that since, in order to turn  $\Pi$  into an  $\ell$ -normal protocol, we might have partitioned each of its parties into several (pseudo) parties, and since we corrupt each small-jumps party with probability proportional to  $1/\sqrt{\ell}$ , it seems that we might have corrupted too many parties. But since, by the  $\ell$ -normality of the protocol, there are at most  $\ell$  parties with sum of conditional variances less than  $1/\ell$ , Lemma 2.6 yields that, in expectation, the above partition only induces  $O(\ell)$  additional (pseudo) parties.

---

<sup>5</sup>Assuming  $\Pi$  is an  $\ell^2$ -round majority protocol in which each party sends  $\ell$  coins, then (apart from the very last rounds) the change induced by any given message is (absolute) order of  $1/\ell$ . Hence, each  $v_i$  is of order  $1/\ell^2$ , and each party will be independently corrupted with probability  $1/\varepsilon^3 \cdot 1/\sqrt{\ell}$  (i.e., first *if* of Step (2) is never triggered). Thus, in expectation, the above attack corrupts  $1/\varepsilon^3 \cdot \sqrt{\ell}$  parties. If corrupt, each of the  $\ell$  bit-messages the party sends is 1 with probability  $\approx 1/2 \cdot (1 + \sqrt{\ell} \cdot 1/\ell) = 1/2 + 1/2\sqrt{\ell}$ .

The challenge in proving Lemma 2.6 is that unlike the single-turn proof, it might be that the following core inequality does not hold:<sup>6</sup>

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

Indeed, let  $V_i$  be the value of the variables  $v_i$  in the execution of the attack described by  $\widehat{\text{Msg}}_{<i}$ . Assume that conditioned on  $\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}$ , it holds that  $V_i < 1/\ell$  and that a small-jumps party  $P$  is about to send the  $i^{\text{th}}$  message. Unlike the single-turn case, the conditional probability that  $P$  is corrupted is no longer guaranteed to be  $1/\varepsilon^3 \cdot 1/\sqrt{\ell}$ : 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  $\mathbb{E}[S_i - S_{i-1} \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}] < 1/\varepsilon^3 \cdot \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i} = \text{msg}_{<i}]$ .

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  $2/\ell$ , a KL-divergence argument yields that on average (in some sense) for a message sent by a small-jumps party it holds that  $\mathbb{E}[S_i - S_{i-1} \mid \widehat{\text{Msg}}_{i-1} = \text{msg}_{<i}] = \Omega(1/\varepsilon^3 \cdot \text{Var}[Y_i \mid \widehat{\text{Msg}}_{i-1} = \text{msg}_{<i}])$ , which suffices for the proof of the lemma to go through.

### 2.3 Attacking Non-Robust Coin Flip

The high level idea of attacking arbitrary (non-robust) protocols that might have  $1/\sqrt{\ell}$  jumps in both direction (e.g.,  $\Pi(\text{Msg}_{<i}) < \Pi(\text{Msg}_{<i}) - 1/\sqrt{\ell}$ ) is rather straightforward. Attempt biasing the protocol toward zero using large negative jumps. If failed, you are essentially in the situation that allows you to apply the above attack for robust protocols. More formally, assume a protocol  $\Pi$  has a (large) negative jump with probability at least  $1/\log \ell$ , and consider the following single-corruption attacker:<sup>7</sup>

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

For  $i = 1$  to  $\ell$ , do the following before the  $i^{\text{th}}$  message is sent:

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

It is clear that the above adversary biases the outcome of  $\Pi$  toward zero by at least  $1/\log(\ell) \cdot \sqrt{\ell}$ . Let  $\Pi_1$  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  $\Pi_1$  has a negative jump with probability larger than  $1/\log n$ , apply the above attack on  $\Pi_1$  to get a protocol  $\Pi_2$ , and so on...

It is clear that the above process halts after at most  $t \leq \log(\ell) \cdot \sqrt{\ell}$  such iterations. Consider the  $t$ -corruption adversary for  $\Pi$  defined by these  $t$  attacks: it runs the  $t$  adversaries in parallel,

<sup>6</sup>Recall that the above equality allowed us to argue that  $\mathbb{E}[\sum_{i=1}^{\ell} \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i}]] \leq \varepsilon^3 \cdot \mathbb{E}[S_n - S_0] \leq \varepsilon^3$ .

<sup>7</sup>Interestingly, assuming the next-message function of  $\Pi$  is efficient, e.g.,  $\Pi$  is public-coin, the following attacker, when used, is the only reason for the inefficiency of our attack.

and corrupt a message whenever one of the  $t$  adversaries decides to do so. Assuming the expected outcome of  $\Pi_t$  is at most  $\varepsilon$ , we are done. Otherwise, for some  $i \in [t]$  the protocol  $\Pi_i$  has the following property:

$$\mathbb{P}[\exists j \in [n]: \Pi_i(\widetilde{\text{Msg}}_{\leq j}) < \Pi_i(\widetilde{\text{Msg}}_{< j}) - 1/\sqrt{\ell}] \leq 1/\log(\ell) \quad (10)$$

letting  $\widetilde{\text{Msg}}$  be the messages of a random execution of  $\Pi_i$ .

If the above happens, then we apply the attack on robust protocols (Algorithm 2.5) on  $\Pi_i$ , instructing the adversary not to attack a message of large negative jump. With a careful analysis (actually, we need to slightly refine the attack for that), one can show that the above attack on  $\Pi_i$  encounters a large negative jump with probability only  $O(\varepsilon)$ . Hence, it successfully biases the expected output of  $\Pi_i$  to  $1 - O(\varepsilon)$  (since with overwhelming probability the attack carries as if there are no large negative jumps). Combining the attack that turns  $\Pi$  into  $\Pi_i$  with the attack on  $\Pi_i$ , yields the required attacker.

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

#### 3.2 Distributions and Random Variables

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

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

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

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

where  $0 \cdot \log \frac{0}{0} = 0$ , and if there exists  $x \in \mathcal{X}$  such that  $P(x) > 0 = Q(x)$  then  $D_{\text{KL}}(P \| Q) := \infty$ . 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  $D_{\text{KL}}(\lambda \cdot P + (1 - \lambda) \cdot Q \| Q) \leq \lambda \cdot D_{\text{KL}}(P \| Q)$ .*

The following fact (see Fedotov et al. [14]) relates small KL-divergence to small statistical distance:

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

### 3.3 Martingales

Martingales play an important role in our analysis.

**Definition 3.4** (Martingales). *A sequence of random variables  $M = (M_1, \dots, M_n)$  is a martingale with respect to a sequence of random variables  $X_1, \dots, X_n$ , if  $\mathbb{E}[M_{k+1} | X_{\leq k}] = M_k$  and  $M_k$  is determined by  $X_{\leq k}$  for every  $k \in [n]$ . The sequence  $M$  is a martingale, if it is a martingale with respect to itself. The increments (also known as, differences) sequence of  $M$  are the random variables  $\{M_{k+1} - M_k\}_{k=1}^{n-1}$ .*

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

**Definition 3.5** (Doob martingales). *The Doob martingale of the random variables  $X = (X_1, \dots, X_n)$  induced by the function  $f: \text{Supp}(X) \mapsto \mathbb{R}$ , is the sequence  $M_1, \dots, M_n$  defined by  $M_k := \mathbb{E}[f(X_1, \dots, X_n) | X_{\leq k}]$ .*

The proof of the following known fact is immediate.

**Fact 3.6** (Martingale increments are orthogonal). *Let  $X_1, \dots, X_n$  be a sequence of random variables. If there exist random variables  $Z_1, \dots, Z_n$  such that  $\mathbb{E}[X_k | Z_{<k}] = 0$  and  $X_k$  is determined by  $Z_{\leq k}$  (i.e.,  $\sum X_i$  is a martingale with respect to  $Z_k$ ), then  $\text{Var}[\sum_{i=1}^n X_i] = \sum_{i=1}^n \text{Var}[X_i]$ .*

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

**Definition 3.7** (Sub-martingales). *A sequence of random variables  $S = (S_1, \dots, S_n)$  is a sub-martingale with respect to a sequence of random variables  $X_1, \dots, X_n$ , if  $\mathbb{E}[S_{k+1} | X_{\leq k}] \geq S_k$  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  $S_1, \dots, S_n$  be a non-negative sub-martingale, then for any  $c > 0$  it holds that  $\mathbb{P}[\sup_k S_k \geq c] \leq \mathbb{E}[S_n]/c$ .*

### 3.4 Full-Information Coin Flip

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

**Definition 3.9** (Full-information coin-flipping protocols). *A protocol  $\Pi$  is a full-information coin-flipping protocol if it is stateless,<sup>8</sup> 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. [24]). 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 the natural formulation of this model) to the static adversary cases, in which we know that  $\Theta(\ell/\log \ell)$  corruptions ( $\ell$  being the number of parties) are required.*

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

**Notation 3.11.** *Let  $\Pi$  be an  $n$ -round  $\ell$ -party full-information coin-flipping protocol.*

- *Let  $\text{Msg}^\Pi = (\text{Msg}_1^\Pi, \dots, \text{Msg}_n^\Pi)$  denote a random transcript (i.e., parties' messages) of  $\Pi$ .*
- *For partial transcript  $\text{msg}_{\leq i} \in \text{Supp}(\text{Msg}_{\leq i}^\Pi)$ , let  $\Pi(\text{msg}_{\leq i}) := \mathbb{E}[\Pi(\text{Msg}^\Pi) \mid \text{Msg}_{\leq i}^\Pi = \text{msg}_{\leq i}]$ .*

*(I.e., the expected outcome of  $\Pi$  given  $\text{msg}_{\leq i}$ )*

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

- *For  $\text{msg}_{< i} \in \text{Supp}(\text{Msg}_{< i}^\Pi)$ , let  $\text{party}(\text{msg}_{< i}) \in [\ell]$  be the identity of the party to send the  $i^{\text{th}}$  message, as determined by  $\text{msg}_{< i}$ .*

*For party  $P \in [\ell]$  and transcript  $\text{msg} \in \text{Supp}(\text{Msg}^\Pi)$ , let  $\text{Idx}_P(\text{msg}) := \{i \in [n] : \text{party}(\text{msg}_{< i}) = P\}$ .*

- *For  $\text{msg}_{\leq i} \in \text{Supp}(\text{Msg}_{\leq i}^\Pi)$ , let  $\text{jump}^\Pi(\text{msg}_{\leq i}) := \Pi(\text{msg}_{\leq i}) - \Pi(\text{msg}_{< i})$ .  
(i.e.,  $\text{jump}^\Pi(\text{msg}_{\leq i})$  is the increment in expectation caused by the  $i^{\text{th}}$  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.

---

<sup>8</sup>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.

## The attacked protocol.

**Definition 3.13** (The attacked protocol). *Given a full-information coin-flipping protocol  $\Pi$  and a deterministic (adaptive) adversary  $A$  for attacking it, let  $\Pi_A$  be the full-information coin-flipping protocol induced by this attack: the parties act according to  $\Pi$  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  $\Pi_A$  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  $\Pi$  there exist deterministic adversaries  $A^+$  and  $A^-$  such that  $\mathbb{E}[\Pi_{A^+}] \geq \mathbb{E}[\Pi_A]$  and  $\mathbb{E}[\Pi_{A^-}] \leq \mathbb{E}[\Pi_A]$ .*

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

## Composition.

**Definition 3.15** (Composing adaptive adversaries). *Let  $\Pi$  be a coin-flipping protocol, let  $A$  be a deterministic  $k_A$ -adaptive adversary for  $\Pi$ , and let  $B$  be a  $k_B$ -adaptive adversary for  $\Pi_A$ . The  $(k_A + k_B)$ -adaptive adversary  $B \circ A$  on  $\Pi$  is defined as follows:*

**Algorithm 3.16** (Adversary  $B \circ A$  on  $\Pi$ ).

**For**  $i := 1$  **to**  $\text{NumMsgs}(\Pi)$ :

*If*  $P \in \{A, B\}$  *would like to modify the*  $i^{\text{th}}$  *message, alter it according to*  $P$ .

**Proposition 3.17.** *Let  $\Pi$ ,  $A$  and  $B$  be as in Definition 3.15, then  $\mathbb{E}[\Pi_{B \circ A}] = \mathbb{E}[(\Pi_A)_B]$ .*

*Proof.* Since  $A$  is deterministic,  $B$  never modifies a message in  $\Pi_A$  that is to be modified by  $A$ . Indeed, since  $B$  is a valid attacker it never sends a message out of the support of  $\Pi_A$ , and a message to be corrupted by  $A$  is fixed. It follows that  $(\Pi_A)_B$  and  $\Pi_{B \circ A}$  induce the same distribution on the protocol tree of  $\Pi$ , 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 for 0, see below, fix  $\ell \in \mathbb{N}$  (the number of parties of the robust protocol), and make use of the following constants.

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

The main result of this section is stated below.

**Definition 4.2** (Robust coin-flipping protocols). *An  $n$ -round full-information coin-flipping protocol  $\Pi$  is  $\ell$ -robust, if  $\mathbb{P}[\exists i \in [n]: \text{Supp}(\text{jump}^\Pi(\text{Msg}_{\leq i}^\Pi) \mid \text{Msg}_{< i}^\Pi) \cap (-\infty, -1/\lambda \cdot \sqrt{\ell}) \neq \emptyset] \leq \delta$ .*

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

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 a party does not influence the protocol “by much” over multiple rounds, though it might have large influencing over a single round. In Section 4.2, we use this attack for proving Theorem 4.3, by transforming the given protocol into a normal protocol, and show that the guaranteed attack on the latter protocol yields an attack of essentially the same quality on the original protocol.<sup>9</sup>

#### 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 loss in parameters, as a normal coin-flipping protocol.

**Definition 4.4** (Normal coin-flipping protocols). *Let  $\Pi$  be a  $t$ -party,  $n$ -round full-information coin-flipping protocol and let  $\ell \in \mathbb{N}$ . We say  $\Pi$  is  $\ell$ -normal, if the following hold:*

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

**Large-jump party sends a single message:** *a party  $P \in [t] \setminus \{\text{NonRobust}\}$  has large jumps in  $\text{msg}$ , if  $\exists i \in \mathcal{Idx}_P(\text{msg})$  s.t.  $\text{Var}[\text{jump}^\Pi(\text{Msg}_{\leq i}^\Pi) \mid \text{Msg}_{< i}^\Pi = \text{msg}_{< i}] \geq 1/\lambda\ell$ . Then for every large-jump party  $P$  in  $\text{msg} \in \text{Supp}(\text{Msg}^\Pi)$  it holds that  $|\mathcal{Idx}_P(\text{msg})| = 1$ .*

**Small-jumps party has bounded overall variance:** *a party  $P \in [t] \setminus \{\text{NonRobust}\}$  has small jumps in  $\text{msg}$ , if it participates (sends a message) but has no large jumps in  $\text{msg}$ .*

*Then for every small-jumps party  $P$  in  $\text{msg} \in \text{Supp}(\text{Msg}^\Pi)$  it holds that*

$$\sum_{i \in \mathcal{Idx}_P(\text{msg})} \text{Var}[\text{jump}^\Pi(\text{Msg}_{\leq i}^\Pi) \mid \text{Msg}_{< i}^\Pi = \text{msg}_{< i}] \leq 2 \cdot 1/\lambda\ell.$$

**At most  $\ell$  unfulfilled parties:** *a small-jumps party is unfulfilled in  $\text{msg}$  if  $\sum_{i \in \mathcal{Idx}_P(\text{msg})} \text{Var}[\text{jump}^\Pi(\text{Msg}_{\leq i}^\Pi) \mid \text{Msg}_{< i}^\Pi = \text{msg}_{< i}] < 1/\lambda\ell$ .*

*Then in every  $\text{msg} \in \text{Supp}(\text{Msg}^\Pi)$  there are at most  $\ell$  unfulfilled parties.*

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

<sup>9</sup>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.



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  $\Pi$  be a  $\ell$ -normal,  $n$ -round full-information coin-flipping protocol. If  $\mathbb{E}[\Pi] \geq \varepsilon$  and  $\mathbb{P}[\mathcal{I}dx_{\text{NonRobust}}(\text{Msg}^\Pi) \neq \emptyset] \leq \delta$ , then there exists an  $O(\sqrt{\ell} \cdot \log \ell)$ -adaptive adversary  $\mathbf{A}$  such that  $\mathbb{E}[\Pi_{\mathbf{A}}] \geq 1 - O(\varepsilon)$ .*

That is, if  $\Pi$  is  $\ell$ -normal and  $\ell$ -robust, then it can be biased by an  $O(\sqrt{\ell} \cdot \log \ell)$ -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  $f: \text{Supp}(X) \mapsto [-1/\alpha, \infty)$  such that  $\mathbb{E}[f(X)] = 0$ . We define the distribution  $\text{Biased}_\alpha^f(X)$  as follows:  $\mathbb{P}[\text{Biased}_\alpha^f(X) = x] = \mathbb{P}[X = x] \cdot (1 + \alpha f(x))$ .*

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  $f: \text{Supp}(X) \mapsto [-1/2\alpha, \infty)$  such that  $\mathbb{E}[f(X)] = 0$ . The following hold:*

1.  $\mathbb{E}[f(\text{Biased}_\alpha^f(X))] = \alpha \cdot \text{Var}[f(X)]$ .
2.  $D_{\text{KL}}(\text{Biased}_\alpha^f(X) \parallel X) \leq 2\alpha^2 \cdot \text{Var}[f(X)]$ .
3. For any  $0 \leq p \leq 1$  it holds that  $(p \cdot \text{Biased}_\alpha^f(X) + (1 - p) \cdot X) \equiv \text{Biased}_{p\alpha}^f(X)$ .
4. There exists a pair of random variables  $(A, B)$ , i.e., a coupling, such that  $A \equiv X$ ,  $B \equiv \text{Biased}_\alpha^f(X)$ , and  $f(B) \geq f(A)$ .

*Proof.*

**Item 1:**

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

**Item 2:**

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

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

**Item 3:**

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

**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

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

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  $\text{Biased}_\alpha^f(X)$ , respectively.  $\square$

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

**Algorithm 4.8** (Adversary A).

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

1. *Let  $\mathsf{P}$  be the party about to send the  $i^{\text{th}}$  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  $\text{Msg}_i^\Pi \mid_{\text{Msg}_{<i} = \text{msg}_{<i}}$ , let  $\text{jump}_i := \text{jump}^\Pi(\text{msg}_{<i}, \cdot)$  and let  $v_i := \text{Var}[\text{jump}_i(Q_i)]$ .*
3. **If** *this is the first message sent by  $\mathsf{P}$ , corrupt  $\mathsf{P}$  according to the following method:*
  - (a) **If**  $\mathsf{P}$  *is a large-jump party, i.e.,  $v_i \geq 1/\lambda\ell$ , corrupt it with probability  $\lambda^2 \cdot \sqrt{v_i}$ .*<sup>10</sup>

<sup>10</sup> $\lambda^2 \cdot \sqrt{v_i}$  is indeed a number in the range  $[0, 1]$ :  $\mathsf{P} \neq \text{NonRobust}$  implies that  $v_i \leq 1/\lambda \cdot \sqrt{\ell}$ .

(b) **Else** ( $P$  is a small-jumps party), corrupt  $P$  with probability  $\lambda^2/\sqrt{\ell}$ .

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

(b) **Else**:

- i. **If**  $\mathbb{P}[P \text{ is corrupted by } A \mid \text{Msg}_{<i}^A = \text{msg}_{<i}] \leq 16\lambda^2/\sqrt{\ell}$ , for  $\text{Msg}_{<i}^A$  which is the distribution of messages sent in a random execution of  $\Pi$  in the presence of  $A$  for the first  $i-1$  rounds,<sup>11</sup> instruct  $P$  to broadcast its next message according to  $\text{Biased}_{\sqrt{\ell}}^{\text{jump}_i}(Q_i)$ .
- 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 stop modifying the messages 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.<sup>12</sup>

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

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  $Y = (Y_1, \dots, Y_n)$  jointly distributed with  $\widehat{\text{Msg}}$ , such that the following holds for every  $i \in [n]$ ,  $\text{msg}_{<i} \in \text{Supp}(\widehat{\text{Msg}}_{<i})$  and  $c \in \{0, 1\}$ :*

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

That is,  $Y_i$  is distributed like the (conditional) change in outcome induced by the  $i^{\text{th}}$  step if it were carried out honestly, and is never larger than the (conditional) change induced by the  $i^{\text{th}}$  step of the attacked execution. It is easy to verify that  $\mathbb{E}[Y_k \mid \widehat{\text{Msg}}_{<k}, Y_{<k}] = 0$ , i.e.,  $\sum_{i=1}^k Y_i$  is a

<sup>11</sup>Since we are defining the strategy of  $A$  in the  $i^{\text{th}}$  round using its strategy in the first  $i-1$  rounds, this self-reference is well defined.

<sup>12</sup>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).

martingale with respect to  $(\widehat{\text{Msg}}_k, Y_k)$ . 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  $\widehat{\text{Msg}}$ , defined by

$$\text{RobustJumps} := [n] \setminus \text{Idx}_{\text{NonRobust}}(\widehat{\text{Msg}}) \quad (11)$$

Note that  $\text{RobustJumps}$  is a random set (determined by  $\widehat{\text{Msg}}$ ). 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).  $\mathbb{E}[\sum_{i \in \text{RobustJumps}} \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]] < 2/\lambda$ .

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,  $\text{NonRobust}$  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  $D_{\text{KL}}(\widehat{\text{Msg}} \parallel \text{Msg}) \leq 16^3 \lambda^3$ .*

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

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

The first equality holds by Fact 3.6 (since  $\mathbb{E}[Y_k | \widehat{\text{Msg}}_{<k}, Y_{<k}] = 0$  and  $\mathbb{1}_{i \in \text{RobustJumps}}$  is determined by  $\widehat{\text{Msg}}_{<k}$ ). The inequality holds by Claim 4.10. Since  $\mathbb{E}[\sum_{i \in \text{RobustJumps}} Y_i] = 0$ . Thus, by Markov’s inequality  $\mathbb{P}\left[\left|\sum_{i \in \text{RobustJumps}} Y_i\right|^2 \geq \varepsilon^2/4\right] < \varepsilon/4$ , which implies that

$$\mathbb{P}\left[\sum_{i \in \text{RobustJumps}} Y_i \leq -\varepsilon/2\right] < \varepsilon/4 \quad (13)$$

Next we show that with overwhelming probability  $\text{RobustJumps} = [n]$ , namely  $\text{NonRobust}$  does not participate in the execution. Let  $\text{Bad}^{\Pi}$  be the event  $\{\text{Idx}_{\text{NonRobust}}(\text{Msg}) \neq \emptyset\}$ . By assumption,  $\mathbb{P}[\text{Bad}^{\Pi}] \leq \delta$ . Let  $\text{Bad}$  be the event  $\{\text{Idx}_{\text{NonRobust}}(\widehat{\text{Msg}}) \neq \emptyset\}$ , and let  $p := \mathbb{P}[\text{Bad}]$ . Note that

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

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

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

The penultimate inequality follows by Fact 3.19. Assume by the way of contradiction that  $p \geq \varepsilon/4$ . Thus (for large enough  $\ell$ ),

$$\sqrt{\log \log \ell} > 16^3 \lambda^3 \geq D_{\text{KL}}(\mathbb{1}_{\text{Bad}} \parallel \mathbb{1}_{\text{Bad}^\Pi}) \geq 1/4 \cdot 1/^{50\sqrt{\log \log \ell}} \cdot \log(\log^2 \ell / 4 \cdot 1/^{50\sqrt{\log \log \ell}}) \geq \sqrt{\log \log \ell},$$

yielding the contraction  $\sqrt{\log \log \ell} > \sqrt{\log \log \ell}$  (for large enough  $\ell$ ). We conclude that  $p \leq \varepsilon/4$ . Combining the above observations, we conclude that

$$\begin{aligned} \mathbb{E}[\Pi_A] &= \mathbb{P}[S_n = 1] = \mathbb{P}[S_n > 0] = \mathbb{P}[S_0 + \sum_{i=1}^n X_i > 0] = \mathbb{P}[\sum_{i=1}^n X_i > -\varepsilon \geq -S_0] \\ &\geq \mathbb{P}\left[\sum_{i \in \text{RobustJumps}} Y_i > -\varepsilon\right] - \mathbb{P}[\text{NonRobust}(\widehat{\text{Msg}}_{\leq n}) \neq \emptyset] \geq (1 - \varepsilon/4) - \varepsilon/4 \geq 1 - \varepsilon/2. \end{aligned} \quad (16)$$

**Number of corruptions.** So it is left to prove that  $A$  does not make too many corruptions. We do that by calculating the *expected* number of corruptions, and use a Markov bound. We introduce several additional notations. Let  $\text{SmallParties}$  and  $\text{LargeParties}$  be the (random) set of small-jumps and large-jump parties with respect to  $\widehat{\text{Msg}}$  (that participate in the execution), respectively. Let  $\text{SmallJumps} := \{k \in [n]: \text{party}(\widehat{\text{Msg}}_{<k}) \in \text{SmallParties}\}$  be the set of small jumps, and let  $\text{LargeJumps} := \{k \in [n]: \text{party}(\widehat{\text{Msg}}_{<k}) \in \text{LargeParties}\}$  be the set of large jumps. Note that the above random sets are determined by  $\widehat{\text{Msg}}$ .

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

$$\mathbb{E}[|\text{SmallParties} \cap \text{CorruptedParties}|] = \lambda^2/\sqrt{\ell} \cdot \mathbb{E}[|\text{SmallParties}|] \quad (17)$$

In addition, the definition of an  $\ell$ -normal protocols stipulates that for any transcript of  $\Pi$  there are at most  $\ell$  unfulfilled parties. Since all non-unfulfilled parties contribute at least  $1/\lambda\ell$  to the sum of variances, which is small by Claim 4.10, we deduce that

$$\mathbb{E}[|\text{SmallParties}|] \leq 3\ell \quad (18)$$

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

$$\mathbb{E}[|\text{SmallParties} \cap \text{CorruptedParties}|] \leq \lambda^2/\sqrt{\ell} \cdot 3\ell = 3\lambda^2\sqrt{\ell} \quad (19)$$

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

$$\begin{aligned} & \mathbb{E}[|\text{LargeParties} \cap \text{CorruptedParties}|] \\ &= \mathbb{E}\left[\sum_{i \in \text{LargeJumps}} \lambda^2 \cdot \sqrt{\text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]}\right] \leq \mathbb{E}\left[\sum_{i \in \text{LargeJumps}} \lambda^2 \cdot \sqrt{\lambda \ell} \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]\right] \\ &\leq \lambda^3 \cdot \sqrt{\ell} \cdot \mathbb{E}\left[\sum_{i \in \text{RobustJumps}} \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]\right] \leq \lambda^3 \cdot \sqrt{\ell} \cdot 2/\lambda = 2\lambda^2 \sqrt{\ell}. \end{aligned} \tag{20}$$

The last inequality follows by Claim 4.10. Thus in total, the expected amount of corruptions is at most  $5\lambda^2 \sqrt{\ell}$ . Hence by Markov inequality, with probability at least  $1 - \varepsilon/2$  the amount of corruptions made by  $A$  is at most  $10\lambda^3 \sqrt{\ell}/\varepsilon < 10\lambda^4 \sqrt{\ell}$ .

**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  $10\lambda^4 \sqrt{\ell} = O(\sqrt{\ell} \cdot \log \ell)$ . It holds that

$$\begin{aligned} \mathbb{E}[\Pi_{A'}] &= \mathbb{P}[\Pi_{A'} = 1] \geq \mathbb{P}[\Pi_A = 1 \wedge |\text{CorruptedParties}| < 10\lambda^4 \sqrt{\ell}] \\ &\geq \mathbb{P}[\Pi_A = 1] - \mathbb{P}[|\text{CorruptedParties}| \geq 10\lambda^4 \sqrt{\ell}] \geq 1 - \varepsilon/2 - \varepsilon/2 = 1 - \varepsilon, \end{aligned}$$

which concludes the proof of the lemma.  $\square$

#### 4.1.1 Coupling $X_i$ and $Q_i$ , Proving Claim 4.9

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

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

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

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  $X_i \geq Y_i$ , that the conditional

distributions are as required, and that  $Y_i$  is independent of  $Y_{<i}$  and  $C$  (conditioned on  $\widehat{\text{Msg}}_{<i}$ ).  $\square$

#### 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  $\text{RobustJumps} = [n] \setminus \text{Idx}_{\text{NonRobust}}(\widehat{\text{Msg}}) = \text{SmallJumps} \cup \text{LargeJumps}$ .

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,  $\alpha : \text{Supp}(Z) \mapsto \mathbb{R}_+$  and  $f : \text{Supp}(X) \mapsto \mathbb{R}$  such that  $\forall z \in \text{Supp}(Z): f(X|_{Z=z}) \geq -1/\alpha(z)$ , and  $\mathbb{E}[f(X)|Z] = 0$ . We define a random variable  $\text{Biased}_\alpha^f(X|Z)$  jointly distributed with  $Z$ , by  $\text{Biased}_\alpha^f(X|Z)|_{Z=z} \equiv \text{Biased}_{\alpha(z)}^f(X|_{Z=z})$ , for any  $z \in \text{Supp}(Z)$ .*

#### Large jumps.

**Claim 4.13.**  $\mathbb{E}[\sum_{i \in \text{LargeJumps}} \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]] < 1/\lambda$ .

*Proof.* We assume towards a contradiction that  $\mathbb{E}[\sum_{i \in \text{LargeJumps}} \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}]] \geq 1/\lambda$ , prove that this implies that  $\mathbb{E}[S_n] > 1$ , and derive a contradiction to the fact that  $S_n \in \{0, 1\}$ . For  $k \in [n]$ , let  $E_k := \{k \in \text{LargeJumps}\}$ , and note that  $\mathbb{1}_{E_k}$  is determined by  $\widehat{\text{Msg}}_{<k}$ . Compute,

$$\mathbb{1}_{E_k} \cdot \mathbb{E}[X_k | \widehat{\text{Msg}}_{<k}] \tag{21}$$

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

$$\begin{aligned} &+ \mathbb{1}_{E_k} \cdot \left(1 - \lambda^2 \sqrt{\text{Var}[Y_k | \widehat{\text{Msg}}_{<k}]}\right) \cdot \mathbb{E}[Y_k | \widehat{\text{Msg}}_{<k}] \\ &= \mathbb{1}_{E_k} \cdot \lambda^2 \sqrt{\text{Var}[Y_k | \widehat{\text{Msg}}_{<k}]} \cdot 1/\sqrt{\text{Var}[Y_k | \widehat{\text{Msg}}_{<k}]} \cdot \text{Var}[Y_k | \widehat{\text{Msg}}_{<k}] + 0 \\ &= \mathbb{1}_{E_k} \cdot \lambda^2 \cdot \text{Var}[Y_k | \widehat{\text{Msg}}_{<k}]. \end{aligned} \tag{23}$$

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  $\ell$ ),

$$\mathbb{E}[S_n] = \mathbb{E}[S_0 + \sum_{i=1}^n X_i] = \mathbb{E}[S_0] + \mathbb{E}[\sum_{i=1}^n \mathbb{E}[X_i | \widehat{\text{Msg}}_{<i}]] \tag{24}$$

$$\geq \varepsilon + \mathbb{E}[\sum_{i=1}^n \mathbb{1}_{\{i \in \text{LargeJumps}\}} \cdot \mathbb{E}[X_i | \widehat{\text{Msg}}_{<i}]] \tag{25}$$

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

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

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

**Small jumps.**

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

In the following for a party  $P$ , let  $I_P$  be the random variable  $\mathcal{I}dx_P(\widehat{\text{Msg}})$ .

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

$$\mathbb{E}[|\text{SmallParties}|] \leq \gamma\lambda\ell + \ell < 2\gamma\lambda\ell \quad (27)$$

We say a party  $P$  is *contributonal* if

$$\mathbb{P} \left[ \sum_{i \in I_P} \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}] > \frac{1}{8} \cdot \frac{1}{\lambda\ell} \mid P \in \text{SmallParties} \right] \geq 1/8 \quad (28)$$

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

$$\mathbb{E}[|\text{SmallContribParties}|] \geq \gamma\lambda\ell/8 \quad (29)$$

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

**Claim 4.15.** *For any contributonal party  $P$  it holds that*

$$\mathbb{E} \left[ \sum_{i \in I_P} X_i \mid P \in \text{CorruptedParties} \wedge P \in \text{SmallContribParties} \right] \geq 1/256\lambda\sqrt{\ell}.$$

Given the above claim, compute

$$\mathbb{E} \left[ \sum_{i=1}^n X_i \right] = \mathbb{E} \left[ \sum_{i=1}^n \mathbb{E}[X_i | \widehat{\text{Msg}}_{<i}] \right] \geq \mathbb{E} \left[ \sum_{P \in \text{SmallContribParties}} \sum_{i \in I_P} \mathbb{E}[X_i | \widehat{\text{Msg}}_{<i}] \right] \quad (30)$$



$$\begin{aligned}
&= \mathbb{E}\left[\sum_{\mathbf{P} \in \text{SmallContribParties}} \sum_{i \in I_{\mathbf{P}}} X_i\right] \\
&= \sum_{\mathbf{P} \in \text{ContribParties}} \mathbb{E}\left[\sum_{i \in I_{\mathbf{P}}} X_i \mid \mathbf{P} \in \text{CorruptedParties} \wedge \mathbf{P} \in \text{SmallContribParties}\right] \\
&\quad \cdot \mathbb{P}[\mathbf{P} \in \text{CorruptedParties} \wedge \mathbf{P} \in \text{SmallContribParties}]
\end{aligned} \tag{31}$$

$$\geq \sum_{\mathbf{P} \in \text{ContribParties}} 1/256\lambda\sqrt{\ell} \cdot \mathbb{P}[\mathbf{P} \in \text{CorruptedParties} \wedge \mathbf{P} \in \text{SmallContribParties}] \tag{32}$$

$$= \sum_{\mathbf{P} \in \text{ContribParties}} 1/256\lambda\sqrt{\ell} \cdot (\mathbb{P}[\mathbf{P} \in \text{SmallContribParties}] \cdot \lambda^2/\sqrt{\ell}) \tag{33}$$

$$= \lambda/256\ell \cdot \mathbb{E}[\text{SmallContribParties}] \geq \gamma\lambda^2/2048. \tag{34}$$

Inequality (30) holds since  $\mathbb{E}[X_i \mid \widehat{\text{Msg}}_{<i}] \geq 0$ . Equality (31) holds since  $\mathbb{E}[\sum_{i \in I_{\mathbf{P}}} X_i \mid \mathbf{P} \notin \text{CorruptedParties} \wedge \mathbf{P} \in \text{SmallContribParties}] = 0$ . 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,  $\mathbb{E}[S_n] = \mathbb{E}[S_0 + \sum_{i=1}^n X_i] \geq \varepsilon + \gamma\lambda^2/2048$ . Thus, since  $\lambda$  is super constant in  $\ell$ , for large enough  $\ell$  it holds that  $\mathbb{E}[\sum_{i=1}^n X_i] > 1$ . This stands in contradiction to the fact that  $S_n \in \{0, 1\}$ .  $\square$

### Proving Claim 4.15.

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

$$H := \left\{\forall k \in I_{\mathbf{P}}: \mathbb{E}[\mathbb{1}_{\{\mathbf{P} \in \text{CorruptedParties}\}} \mid \widehat{\text{Msg}}_{<k}] < 16 \cdot \lambda^2/\sqrt{\ell}\right\},$$

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

To prove that  $\mathbb{P}[H \wedge L \mid S]$  is large, we momentarily move to the conditional probability space where  $S$  occurs (i.e.,  $\mathbf{P}$  participates in the protocol as a small-jumps party). Consider the martingale  $C_0, \dots, C_n$  defined by  $C_k := \mathbb{E}[\mathbb{1}_C \mid \widehat{\text{Msg}}_{\leq k}]$  (that is,  $C_k$  is the projection of the event  $C$  on the information held by  $\widehat{\text{Msg}}_{\leq k}$ ). Since, under the conditioning,  $\mathbf{P}$  is a small-jumps party, it holds that the adversary corrupts  $\mathbf{P}$  with probability  $\lambda^2/\sqrt{\ell}$ , i.e.,  $\mathbb{E}[\mathbb{1}_C] = \mathbb{P}[C] = \lambda^2/\sqrt{\ell}$ . Thus by Doob's maximal inequality (see Lemma 3.8), it holds that

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

Back to the regular probability space, we deduce that

$$\mathbb{P}[L \wedge H \mid S] \geq \mathbb{P}[L \mid S] - \mathbb{P}[\neg H \mid S] \geq 1/8 - 1/16 = 1/16 \tag{36}$$

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

$$D_{\text{KL}}(\widehat{\text{Msg}}|_{S \wedge C} \| \widehat{\text{Msg}}|_S) \quad (37)$$

$$\begin{aligned} &= \sum_{i=1}^{\ell} \mathbb{E}_{\text{msg}_{<i} \leftarrow \widehat{\text{Msg}}|_{S \wedge C}} [D_{\text{KL}}(\widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge C} \| \widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}})] \\ &= \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}|_{S \wedge C}} \left[ \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} D_{\text{KL}}(\widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge C} \| \widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}}) \right] \end{aligned} \quad (38)$$

$$\leq \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}|_{S \wedge C}} \left[ \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} D_{\text{KL}}(\widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge C} \| \widehat{\text{Msg}}_i |_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge \neg C}) \right] \quad (39)$$

$$\leq \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}|_{S \wedge C}} \left[ \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} D_{\text{KL}}(\text{Biased}_{\sqrt{\ell}}^{\text{jump}^{\Pi}(\text{msg}_{<i}, \cdot)}(\text{Msg}_i^{\Pi} |_{\text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}}) \| \text{Msg}_i^{\Pi} |_{\text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}})] \quad (40)$$

$$\leq \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}|_{S \wedge C}} \left[ \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} 2\ell \cdot \text{Var}[\text{jump}^{\Pi}(\text{Msg}_{<i}^{\Pi}) | \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}] \right] \quad (41)$$

$$\leq 2\ell \cdot \mathbb{E}[2/\lambda\ell] \leq 4/\lambda.$$

The first equality follows by chain-rule of KL Divergence. Equality (38) follows by the fact that the distribution of messages not sent by  $\mathbf{P}$  is not affected by conditioning on  $C$ . Inequality (39) follows by Fact 3.2 ( $\widehat{\text{Msg}}_i$  is a convex combination of  $\widehat{\text{Msg}}_i|_C$  and  $\widehat{\text{Msg}}_i|_{\neg C}$ ). 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  $\Pi$  is  $\ell$ -normal.

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

$$\mathbb{P}[H \wedge L | S \wedge C] \geq 1/32 \quad (42)$$

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

$$\begin{aligned} &\mathbb{E}\left[\sum_{i \in I_{\mathbf{P}}} X_i | S \wedge C\right] \\ &= \mathbb{E}\left[\sum_{i \in I_{\mathbf{P}}} X_i | S \wedge C\right] = \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}} \left[ \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} \mathbb{E}[X_i | \widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge C] \right] \\ &\geq \mathbb{E}_{\text{msg} \leftarrow \widehat{\text{Msg}}} \left[ \mathbb{1}_H(\text{msg}) \cdot \sum_{i \in \mathcal{I}dx_{\mathbf{P}}(\text{msg})} \mathbb{E}[\text{Biased}_{\sqrt{\ell}}(Y_i | \widehat{\text{Msg}}_{<i}) | \widehat{\text{Msg}}_{<i} = \text{msg}_{<i} \wedge C] \right] \end{aligned} \quad (43)$$

$$= \mathbb{E}\left[\mathbb{1}_H \cdot \sum_{i \in I_{\mathbf{P}}} \sqrt{\ell} \cdot \text{Var}[Y_i | \widehat{\text{Msg}}_{<i}] | S \wedge C\right] \geq \sqrt{\ell} \cdot \mathbb{E}[\mathbb{1}_H \cdot 1/8\lambda\ell \cdot \mathbb{1}_L | S \wedge C] \quad (44)$$

$$= 1/8\lambda\sqrt{\ell} \cdot \mathbb{P}[H \wedge L \mid S \wedge C] \geq 1/8\lambda\sqrt{\ell} \cdot 1/32 = 1/256\lambda\sqrt{\ell}. \quad (45)$$

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

### 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  $\widehat{\text{Msg}}_{\leq n}$  and  $\text{Msg}_{\leq n}^{\Pi}$  is small. In particular we prove the following claim,

**Claim 4.16** (Restatement of Claim 4.11).  $D_{\text{KL}}(\widehat{\text{Msg}}_{\leq n} \parallel \text{Msg}_{\leq n}^{\Pi}) \leq 16^3 \lambda^3$ .

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  $\text{Biased}_{\alpha}^f$ , is equivalent to biasing this message according to  $\text{Biased}_{p\alpha}^f$ . This fact yields the following observation:

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

$$D_{\text{KL}}(\widehat{\text{Msg}}_i \mid_{\widehat{\text{Msg}}_{<i}=\text{msg}_{<i}} \parallel \text{Msg}_i^{\Pi} \mid_{\text{Msg}_{<i}^{\Pi}=\text{msg}_{<i}}) \leq 16^3 \lambda^4 \cdot \text{Var}[\text{jump}^{\Pi}(\text{Msg}_{\leq i}^{\Pi}) \mid \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}].$$

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

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

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

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

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

□

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

$$\begin{aligned} & D_{\text{KL}}(\widehat{\text{Msg}}_{\leq n} \parallel \text{Msg}_{\leq n}^{\Pi}) \\ &= \sum_{i=1}^{\ell} \mathbb{E}_{\widehat{\text{msg}}_{<i} \leftarrow \widehat{\text{Msg}}_{\leq i}} \left[ D_{\text{KL}}(\widehat{\text{Msg}}_i \mid_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}} \parallel \text{Msg}_i^{\Pi} \mid_{\text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}}) \right] \end{aligned} \quad (46)$$

$$= \mathbb{E}_{\widehat{\text{msg}} \leftarrow \widehat{\text{Msg}}} \left[ \sum_{i \in \text{RobustJumps}(\text{msg})} D_{\text{KL}}(\widehat{\text{Msg}}_i \mid_{\widehat{\text{Msg}}_{<i} = \text{msg}_{<i}} \parallel \text{Msg}_i^{\Pi} \mid_{\text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}}) \right] \quad (47)$$

$$\leq \mathbb{E}_{\widehat{\text{msg}} \leftarrow \widehat{\text{Msg}}} \left[ \sum_{i \in \text{RobustJumps}(\text{msg})} 16^3 \lambda^4 \cdot \text{Var}[\text{jump}^{\Pi}(\text{Msg}_{\leq i}^{\Pi}) \mid \text{Msg}_{<i}^{\Pi} = \text{msg}_{<i}] \right] \quad (48)$$

$$\begin{aligned} &= 16^3 \lambda^4 \cdot \mathbb{E} \left[ \sum_{i \in \text{RobustJumps}} \text{Var}[\text{jump}^{\Pi}(\text{Msg}_{\leq i}^{\Pi}) \mid \text{Msg}_{<i}^{\Pi} = \widehat{\text{Msg}}_{<i}] \right] \\ &= 16^3 \lambda^4 \cdot \mathbb{E} \left[ \sum_{i \in \text{RobustJumps}} \text{Var}[Y_i \mid \widehat{\text{Msg}}_{<i}] \right] \end{aligned} \quad (49)$$

$$\leq 16^3 \lambda^3. \quad (50)$$

Equality (46) follows by chain rule of KL Divergence, Equality (47) follows since non- $\text{RobustJumps}$  are not corrupted (as they belong to  $\text{NonRobust}$ ), 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 protocol proved to exist in Section 4.1, for attacking arbitrary robust protocols. We do that by transforming an arbitrary robust protocol into a related normal coin-flipping protocol, and proving that the attack on the latter normal protocol stated in 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  $\Pi$  be an  $n$ -round,  $\ell$ -party full-information coin-flipping protocol. Its  $n$ -round,  $(t := 2n \cdot \ell + 1)$ -party,  $\ell$ -normal variant  $\widetilde{\Pi}$  is defined as follows:

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

- For each party  $\mathsf{P}$  of the protocol  $\Pi$ , the protocol  $\widetilde{\Pi}$  has  $2n$  parties  $\mathsf{P}_1^{\text{small}}, \dots, \mathsf{P}_n^{\text{small}}$  and  $\mathsf{P}_1^{\text{large}}, \dots, \mathsf{P}_n^{\text{large}}$ . In addition,  $\widetilde{\Pi}$  has an special party named  $\text{NonRobust}$ .
- For each party  $\mathsf{P}$  of  $\Pi$ , start three counters  $L_{\mathsf{P}} = S_{\mathsf{P}} = 1$ , and  $A_{\mathsf{P}} = 0$ .
- In rounds  $i = 1$  to  $n$ , the protocol is defined as follows.
  1. Let  $\text{msg}_{<i}$  denote the messages sent in the previous rounds, and let  $\mathsf{P}$  be the party that would have sent the  $i^{\text{th}}$  message in  $\Pi$  given this transcript.

2. Let  $Q_i$  be the distribution  $\text{Msg}_i^\Pi \mid_{\text{Msg}_{<i}^\Pi = \text{msg}_{<i}}$  and let  $v_i := \text{Var}[\text{jump}^\Pi(\text{msg}_{<i}, Q_i)]$ .
3. Set  $P'$  (the “active” party) as follows:
  - (a) **If**  $\text{Supp}(\text{jump}^\Pi(\text{msg}_{<i}, Q_i)) \cap (-\infty, 1/\lambda \cdot \sqrt{\ell}] \neq \emptyset$ , set  $P'$  to NonRobust.
  - (b) **Else, If**  $v_i \geq 1/\lambda\ell$ , set  $P'$  to  $P_{L_P}^{\text{large}}$ , and update  $L_P = L_P + 1$ .
  - (c) **Else, If**  $v_i < 1/\lambda\ell$ :
    - Set  $P'$  to  $P_{S_P}^{\text{small}}$  and update  $A_P = A_P + v_i$
    - If**  $A_P > 1/\lambda\ell$ :
      - Set  $S_P = S_P + 1$ .
      - Set  $A_P = 0$ .
4.  $P'$  sends the  $i^{\text{th}}$  message, as  $P$  would in  $\Pi$  given the partial transcript  $\text{msg}_{<i}$ .

**Claim 4.19.** Assume  $\Pi$  is an  $\ell$ -party full-information coin-flipping protocol, then  $\tilde{\Pi}$  is a an  $\ell$ -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  $P_k^{\text{large}}$  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  $A_P > 1/\lambda\ell$ , namely the active party has sum of conditional variances larger than  $1/\lambda\ell$ , it is never associated with another jump further along the execution. Thus, since  $A_P$  increases by at most  $1/\lambda\ell$  at a time, it never surpasses  $2 \cdot 1/\lambda\ell$ .

**At most  $\ell$  unfulfilled parties:** Note that parties which have sum of conditional variances at most  $1/\lambda\ell$  must be parties of the form  $P_k^{\text{small}}$ , and it holds that the only parties of this form that participate in the protocol (namely, unfulfilled parties) are  $P_{S_P^f}^{\text{small}}$  where  $P$  is some party (at most  $\ell$ ) and  $S_P^f$  is the final value of  $S_P$ . Thus, in total indeed at most  $\ell$  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  $\tilde{\Pi}$  be the  $\ell$ -normal variant of  $\Pi$  defined by Protocol 4.18. By Lemma 4.5 and Claim 5.4, there exists a  $O(\sqrt{\ell} \cdot \log \ell)$ -adaptive adversary  $\tilde{A}$  for  $\tilde{\Pi}$  such that  $\mathbb{E}[\tilde{\Pi}_{\tilde{A}}] \geq 1 - O(\varepsilon)$ .

Consider the adversary  $A$  on  $\Pi$  that emulates  $\tilde{A}$  while transforming corruptions of the parties of  $\tilde{\Pi}$  to parties of  $\Pi$  according to the mapping implicitly defined in Protocol 4.18. It is clear that  $\mathbb{E}[\Pi_A] = \mathbb{E}[\tilde{\Pi}_{\tilde{A}}] \geq 1 - O(\varepsilon)$ . In addition, since by construction the parties in  $\tilde{\Pi}$  are refinements of the parties in  $\Pi$ , corrupting  $k$  parties in  $\tilde{\Pi}$  is translated to corrupting at most  $k$  parties of  $\Pi$ . We conclude that  $A$  is the desired  $O(\sqrt{\ell} \cdot \log \ell)$ -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  $\varepsilon := 1/\sqrt[5]{\log \log \ell}$ ,  $\lambda := 1/100\varepsilon^5$  and  $\delta := 1/\log^2 \ell$ .

This main result of our paper is given below.

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

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 output towards 0.

**Algorithm 5.3** (One-shot adversary  $\mathbf{B}$  for protocol  $\Gamma$ ).

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

1. Let  $\text{msg}_{<i}$  be the messages sent in the previous rounds, and let  $\mathbf{P}$  be the the party about to send the  $i^{\text{th}}$  message.
2. **If** no message was corrupted before, and  $\mathcal{M}_i := \text{Supp}(\text{jump}^\Gamma(\text{Msg}_i^\Gamma) \mid \text{Msg}_{<i}^\Gamma = \text{msg}_{<i}) \cap (-\infty, -1/\lambda \cdot \sqrt{\ell}] \neq \emptyset$ :  
Instruct  $\mathbf{P}$  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  $\ell$ -party full-information coin-flipping protocol  $\Gamma$ , it holds that:*

$$\mathbb{E}[\Gamma_{\mathbf{B}}] \geq \mathbb{E}[\Gamma] + 1/\lambda \cdot \sqrt{\ell} \cdot \mathbb{P}[\exists i: \text{Supp}(\text{jump}^\Gamma(\text{Msg}_{\leq i}^\Gamma) \mid \text{Msg}_{<i}^\Gamma) \cap (-\infty, 1/\lambda \cdot \sqrt{\ell}] \neq \emptyset].$$

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  $t := \sqrt{\ell}/\lambda\delta = O(\sqrt{\ell} \cdot \log^3 \ell)$ , consider the protocols  $\Pi^0, \dots, \Pi^t$  recursively defined by  $\Pi^0 := \Pi$  and  $\Pi^{i+1} := \Pi_{\mathbf{B}}^i$ , where  $\mathbf{B}$  is according to Algorithm 5.3. If  $\mathbb{E}[\Pi^t] < \varepsilon$ , then by Proposition 3.17 there exists a  $t$ -adaptive adversary that biases  $\Pi$ 's output to less than  $\varepsilon$  (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

$$\mathbb{P}[\exists i: \text{Supp}(\text{jump}^\tau(\text{Msg}_{\leq i}^\tau) \mid \text{Msg}_{<i}^\tau) \cap (-\infty, 1/\lambda \cdot \sqrt{\ell}] \neq \emptyset] \leq \delta.$$

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

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

Denote by  $C$  the (deterministic)  $i$ -adaptive adversary according to Definition 3.15 (the composition of all intermediate adversaries) such that  $\Pi_C \equiv \Pi^i$ . Let  $A \circ C$  be the  $O(\sqrt{\ell} \cdot \log^3 \ell)$  attacker according to Definition 3.15, by Proposition 3.17 it holds that  $\mathbb{E}[\Pi_{A \circ C}] = \mathbb{E}[\tau_A] \geq 1 - O(\varepsilon)$ , 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] B. Alon and E. Omri. Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In *Theory of Cryptography Conference (TCC), 2016-A*, pages 307–335, 2016.
- [3] N. Alon and M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM Journal on Computing*, 22(2):403–417, 1993.
- [4] B. Awerbuch, M. Blum, B. Chor, S. Goldwasser, and S. Micali. How to implement Bracha’s  $O(\log n)$  byzantine agreement algorithm. Unpublished manuscript, 1985.
- [5] A. Beimel, E. Omri, and I. Orlov. Protocols for multiparty coin toss with a dishonest majority. *Journal of Cryptology*, 28(3):551–600, 2015.
- [6] 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 *Annual Symposium on Foundations of Computer Science (FOCS)*, 2018.
- [7] 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.
- [8] M. Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1983.
- [9] R. B. Boppana and B. O. Narayanan. Perfect-information leader election with optimal resilience. *SIAM Journal on Computing*, 29(4):1304–1320, 2000.
- [10] N. Buchbinder, I. Haitner, N. Levi, and E. Tsfadia. Fair coin flipping: Tighter analysis and the many-party case. In *Symposium on Discrete Algorithms (SODA)*, pages 2580–2600, 2017.
- [11] R. Cleve. Limits on the security of coin flips when half the processors are faulty. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 364–369, 1986.
- [12] 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.

- [13] 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.
- [14] A. A. Fedotov, P. Harremoës, and F. Topsøe. Refinements of pinsker’s inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.
- [15] S. Goldwasser, Y. T. Kalai, and S. Park. Adaptively secure coin-flipping, revisited. In *Automata, Languages and Programming, 24th International Colloquium (ICALP)*, pages 663–674, 2015.
- [16] I. Haitner and E. Tsfadia. An almost-optimally fair three-party coin-flipping protocol. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 817–836, 2014.
- [17] 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.
- [18] Y. T. Kalai and I. Komargodski. Compressing communication in distributed protocols. In *International Symposium on Distributed Computing (DISC)*, pages 467–479, 2015.
- [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, D. I. Diochnos, and M. Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4536–4543, 2019.
- [21] T. Moran, M. Naor, and G. Segev. An optimally fair coin toss. In *Theory of Cryptography (TCC)*, pages 1–18, 2009.
- [22] 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.
- [23] M. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM Journal on Discrete Mathematics*, 2(2):240–244, 1989.
- [24] 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.