On the Computational Complexity of Coin Flipping

Hemanta K. Maji* Manoj Prabhakaran* Amit Sahai†

June 6, 2011

Abstract

Coin flipping is one of the most fundamental tasks in cryptographic protocol design. Informally, a coin flipping protocol should guarantee both (1) Completeness: an honest execution of the protocol by both parties results in a fair coin toss, and (2) Security: a cheating party cannot increase the probability of its desired outcome by any significant amount. Since its introduction by Blum [B82], coin flipping has occupied a central place in the theory of cryptographic protocols. In this paper, we explore what are the implications of the existence of secure coin flipping protocols for complexity theory. As exposited recently by Impagliazzo [I], surprisingly little is known about this question.

Previous work has shown that if we interpret the Security property of coin flipping protocols very strongly, namely that nothing beyond a negligible bias by cheating parties is allowed, then one-way functions must exist [IL89]. However, for even a slight weakening of this security property (for example that cheating parties cannot bias the outcome by any additive constant $\epsilon > 0$), the only complexity-theoretic implication that was known was that $\mathbf{PSPACE} \nsubseteq \mathbf{BPP}$.

We put forward a new attack to establish our main result, which shows that, informally speaking, the existence of any (weak) coin flipping protocol that prevents a cheating adversary from biasing the output by more than $\frac{1}{4} - \epsilon$ implies that $\mathbf{NP} \nsubseteq \mathbf{BPP}$. Furthermore, for constant-round protocols, we show that the existence of any (weak) coin flipping protocol that allows an honest party to maintain any noticeable chance of prevailing against a cheating party implies the existence of (infinitely often) one-way functions.

*Department of Computer Science, University of Illinois, Urbana-Champaign. {hmaji2,mmp}@uiuc.edu Supported in part by NSF grants CNS 07-16626 and CNS 07-47027.

†Department of Computer Science, UCLA. {sahai}@cs.ucla.edu

1 Introduction

A fundamental problem in cryptography is to design protocols that allow two mutually distrusting parties to agree on a random coin. The problem of coin flipping is certainly intrinsically fascinating, but moreover coin flipping protocols have proven to be extremely useful to the theory and design of secure protocols: For example, they are an essential ingredient in all known secure two-party and multi-party computation protocols (e.g. Goldreich, Micali, and Wigderson [GMW87]). They have also proven to be influential more widely: For example, they provide a primary motivation for the utility of the Common Random String (CRS) model [BFM88], one of the most popular models for cryptographic protocol design.

The problem of coin flipping was introduced in the seminal work of Blum [B82], who described the task by means of the following scenario: Alice and Bob are divorcing, and have agreed to let the ownership of their favorite car be decided by a coin toss: Heads means that Alice gets the car, and Tails means that Bob gets it. Unfortunately, Alice and Bob are not willing to be in the same room, and need to implement this coin toss over the telephone. As such, Alice and Bob want a protocol such that (informally speaking):

1. The transcript of their conversation uniquely determines who gets the car.
2. If both Alice and Bob behave honestly, then Alice and Bob should both get the car with probability $\frac{1}{2}$.
3. If Alice behaves maliciously but Bob behaves honestly, then Alice cannot significantly increase the probability that she gets the car. Similarly, if Bob behaves maliciously but Alice behaves honestly, then Bob cannot significantly increase the probability that he gets the car.

Such a protocol incentivizes honest behavior by both parties, since they know that deviating from the protocol would not allow them to obtain any significant gain. Here, we are making the non-trivial assumption that both parties *want* to get the car – that is, we do not disallow a cheating Alice to increase the probability of Bob getting the car[1]. As such, this notion of coin flipping is often called "weak" coin flipping (explicitly in the quantum cryptography literature [KN04]), in contrast to "strong" coin flipping where neither party should be able to bias the coin significantly in either direction. [2] Of course, a crucial parameter here is how much bias constitutes a "significant gain." Let us define a $(1-\delta)$-secure weak coin flipping protocol to be one where no cheating party can increase the probability of their desired outcome by more than an additive factor of δ.

The Computational Complexity of Coin Flipping. The goal of this paper is to explore the implications of the existence of such (weak[3]) coin flipping protocols for complexity theory. Despite the centrality of randomness and coin flipping to complexity theory and cryptography, surprisingly little is known about this question, as was recently exposited by Impagliazzo[I].

To the best of our knowledge, the only nontrivial results on the subject show, informally, that if one-way functions do not exist, then it must be possible to bias every r-round coin flipping protocol by an additive $\Theta(1/\sqrt{r})$ factor [IL89, CI93, I10]. Informally speaking, this does show that $(1-$ *negligible*$)$-secure weak coin flipping implies the existence of one-way functions. (This result is "tight" for this setting, since if

[1] Note that by symmetry, our requirements imply that if a protocol fails to meet the requirements, it must be the case that either (1) a single party can bias the outcome significantly in both directions, or (2) both parties can bias the outcome significantly in the *same* direction. If neither of these attacks are possible, then there is always a renaming of Alice and Bob that implies that the protocol meets our requirements.

[2] This is also closely related to the notion of "coin flipping with abort", where the requirement, informally speaking, is that unless a cheating party aborts or is "caught cheating" by the honest party, it cannot bias the coin in either direction. Note that such a protocol immediately implies weak coin flipping, since we can define the output of the protocol to be Tails if Alice aborts or is caught cheating, and similarly Heads if Bob aborts or is caught cheating.

[3] Since strong coin flipping and "coin flipping with abort" both imply weak coin flipping, our results of course also apply to the existence of these other types of protocols.

one-way functions exist then weak coin flipping protocols do exist that rule out non-negligible additive bias [B82, GL89].)

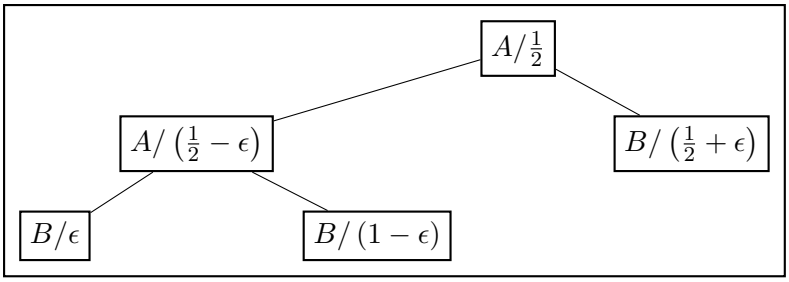
However, what about other natural notions of significant gain? For example, what are the consequences of $(1-\epsilon)$-*secure weak coin flipping protocols* – protocols that do not allow a bias of any additive constant ϵ or more?[4] What about c-secure weak coin flipping protocols where $c \in (0,1)$ is a fixed constant?

For both these questions, the only consequences known are of the flavor that $\mathbf{PSPACE} \not\subseteq \mathbf{BPP}$. Indeed, it is not difficult to see that if $\mathbf{PSPACE} \subseteq \mathbf{BPP}$, then for any coin flipping protocol, either a cheating Alice could force the output 1 with probability 1 or Bob could force the output 0 with probability 1. This attack would proceed by using the power of $\mathbf{PSPACE}$ to perform an iterated min-max (actually max-average) computation, with polynomial look-ahead depth for each round of the protocol. The question before us is whether a similar attack (but with relaxed success goals) could be carried out with much less computational power, for instance with only a constant level of look-ahead, or using a max (instead of max-average) computation – something that intuitively can be carried out with only the power of $\mathbf{NP}$ – even though the overall protocol can have polynomially many rounds?

Our Main Result and Intuition. In this work, we show (as our main result) that the existence of any $\left(\frac{3}{4}+\epsilon\right)$-secure weak coin flipping protocol implies that $\mathbf{NP} \not\subseteq \mathbf{BPP}$. This resolves an open question posed by Impagliazzo [I]: whether $(1-\epsilon)$-secure weak coin flipping protocols are possible if $\mathbf{P} = \mathbf{NP}$ (we show they are not). To prove this result, we introduce a new attack strategy that we call *Hedged Greedy* that is fundamentally different from previous attacks in this setting.

At an intuitive level, previous attacks [IL89, CI93] work by having the attacking party behave honestly until it notices that it has reached a node where its choice will have a significant effect on the expected outcome assuming honest behavior (conditioned on its choices so far) from that point onwards. The attack only deviates from honest behavior at this one point, and the non-triviality of this attack follows from an argument that there must be at least one round in the protocol where the attacker's choice influences the outcome by an additive factor of at least $\Theta(1/\sqrt{r})$ for an r-round protocol. Informally speaking, as pointed out to us by Impagliazzo [I10], this technique fundamentally cannot get a stronger result since a stronger result by this attack would also imply better unconditional attacks on strong coin flipping protocols by fail-stop adversaries, where a bias of $\Theta(1/\sqrt{r})$ is known to be tight (see e.g. [MNS09]).

A conceptually even simpler attack strategy is a "greedy" strategy. To illustrate, let's consider a toy protocol, as described in the protocol tree drawn in Example 1 below.



Example 1: Motivating Hedged-Greedy

In this protocol tree, for instance the annotation "$A/\frac{1}{2}$" on the root node denotes two facts: (1) the *value* (or "color") of this node is $\frac{1}{2}$, meaning that an honest execution of the protocol from this node would lead to a coin with expected value $\frac{1}{2}$ (i.e. a fair coin), and (2) the first message of the protocol is sent by Alice, and the bit that is sent determines which child of this node corresponds to the next step of the protocol. The honest Alice will place appropriate probabilities on its children so as to maintain the value of the coin –

[4] More precisely, for every constant $\epsilon > 0$, for large enough security parameters 1^k provided as common input to both Alice and Bob, the protocol should not allow additive bias of at least ϵ.

in this example at the root node, the honest Alice would proceed left with probability $\frac{1}{2}$, and proceed right otherwise. A leaf node marked "B/ϵ", for instance, just means that honest Bob declares the output to be 1 with probability ϵ, and declares the output to be 0 otherwise (but Bob has full control over the outcome of the protocol if this leaf node is reached).

We would define the "greedy" attack strategy for Alice (when she is attempting to bias the outcome towards 1) to be one where she always proceeds toward the child with higher value. As this example illustrates, however, this attack may obtain only a tiny additive bias: Here, greedy Alice would proceed to the right child at the root of the tree, obtaining an additive bias of only ϵ, which can be arbitrarily small. (In this example, however, greedy Bob would still be quite successful. In Appendix A, we show an extension of this example where greedy strategies for both Alice and Bob perform poorly.)

In this example, the optimal strategy for Alice is in fact to always proceed to the left child; however, in more complex versions of this example (where "dummy" rounds are added in between the actual rounds of the protocol), it may be very difficult for an attacking Alice to realize that she would have near complete control of the outcome on the left branch of the tree. Given only a bounded look-ahead capability to observe nodes in the protocol tree, for instance, Alice would not be able to ascertain whether or not she can control the output of the protocol on the left due to "dummy" rounds.

Instead, the basic intuitive idea behind our attack strategy is to implement a *hedged greedy* strategy: instead of always following the greedy strategy, our attack will "hedge its bet" by also proceeding to the other child with some probability. Intuitively, when the advantage of behaving greedily is clearer, the attack will potentially deviate more from honest behavior. In the example above, at the root node, since the values of the children are so close, the hedged greedy Alice strategy will place almost equal probabilities to both children (behaving very much like the honest Alice would). But at the level below, on the left, where the values of the two children are so different, the hedged greedy Alice strategy would proceed to the node with value $(1-\epsilon)$ with probability very close to 1 (thus deviating very strongly from how honest Alice would behave at this node).

In the example above, the hedged greedy Alice strategy would be able to bias the coin to nearly $\frac{3}{4}$. Through a careful choice of the exact hedging behavior of our strategy, we show that in fact we can guarantee similar performance for *any* protocol – in the sense that either hedged greedy Alice will be able to bias to at least roughly $\frac{3}{4}$ or hedged greedy Bob will be able to bias to at most roughly $\frac{1}{4}$. In fact we prove a more general tradeoff between the relative success of hedged greedy Alice and hedged greedy Bob: for example, if hedged greedy Bob *does not* significantly bias the outcome below $\frac{1}{2}$, then we show that hedged greedy Alice must be able to bias the outcome all the way to roughly 1. (In the example above, however, note that even greedy Bob would be able to guarantee the output 0. As such, the example above only illustrates the idea behind our attack, not the actual analysis of it, which is fairly delicate. We are not aware of any simpler attack and analysis that even guarantees a tiny constant additive bias.) We also show that our analysis of our attack is tight, by showing that in fact *any* attack that bases its decisions on only a bounded look-ahead view of the protocol tree (including the values of the nodes, as illustrated above) cannot obtain better bias.

At a technical level, our proof proceeds in two stages: First, we use the power of $\mathbf{NP}$ to convert an arbitrary coin flipping protocol to a *stateless* coin flipping protocol with nearly identical security guarantees. In a stateless protocol, honest parties only need the transcript of the protocol so far (and fresh randomness) to determine their next move. We then show an unconditional polynomial-time "hedged greedy" attack on any stateless protocol. We believe this modular approach may be of independent interest.

A stronger result for constant-round protocols. For constant round protocols, it is not difficult to see that the "$\mathbf{PSPACE}$" attack described above can be implemented if $\mathbf{NP} \subseteq \mathbf{BPP}$, since a constant round protocol would only involve a constant number of alternations. A natural question is whether any potentially stronger consequence is true.

For this setting, informally speaking, we obtain essentially the best possible result: even the existence

of ϵ-secure weak coin flipping protocols implies the existence of (infinitely often) one-way functions. The core difference between the setting of $\mathbf{NP} \subseteq \mathbf{BPP}$ and the non-existence of one-way functions is that in the latter case, one only obtains an inverse sampler and approximator that works with high probability over a fixed distribution of inputs. Our attack works by showing how to combine a constant number of inverters for a constant number of related functions to carry out the desired attack.

Conclusions and Future Directions. This work revisits the fundamental question of the computational complexity implications of the existence of coin flipping protocols, where surprisingly little was known. We provide new results which show that in many natural settings of parameters, weak coin flipping protocols in fact imply $\mathbf{NP} \not\subseteq \mathbf{BPP}$, where previously only $\mathbf{PSPACE} \not\subseteq \mathbf{BPP}$ was known. We do this by introducing new techniques for this setting, including a *hedged greedy* attack strategy and a method for its analysis.

A number of important natural questions remain open: For the parameters that we consider in our main result, can we conclude that one-way functions exist (and not just that $\mathbf{NP} \not\subseteq \mathbf{BPP}$)? Is it possible that an ϵ-secure weak coin flipping protocol (with polynomially many rounds) can exist even if $\mathbf{P} = \mathbf{NP}$?

2 Preliminaries and Conventions

Consider any 2-party protocol π. We view the transcript generation procedure as traversal of a tree, called the *transcript tree* of π. Any transcript prefix v is a node in this tree and the two extensions of the transcript $v0$ and $v1$ are its two children in the tree. The leaves of the tree are labeled with an output 0 or 1. The depth of the tree D is the communication complexity (maximum number of bits exchanged) of the protocol. Each node is annotated as an Alice node or a Bob node, indicating which party must send the next message in the protocol. When a polynomial blow-up in the round complexity of the protocol is not important, we may consider the Alice and Bob nodes as alternating in any path in the tree. (In Section 4, where the number of rounds is important, we remove the restriction that the tree is binary, but will retain the convention that Alice and Bob nodes alternate.)

The protocol is specified by a randomized algorithm f_π which takes as input a transcript prefix v and a private "state" and outputs an updated state and a next bit (or, if v is a complete transcript, produces a deterministic binary output based only on v). A protocol is called *stateless* if the state variable is always empty.

Let χ_v be the probability of the output of the protocol being 1 conditioned on v being a prefix of the final transcript (when both parties honestly follow the protocol). We call this the *color* of the node v. We shall denote the subtree rooted at v by S_v. We will assume that the height D of the protocol π is the security parameter. When we mention that some event occurs with high probability (written as w.h.p.) it implies that the probability of that event is at least $1 - \exp(-\Theta(D + 1/\epsilon))$.

We define a μ-secure protocol for a χ^*-weak coin as follows:

Definition 1 (μ-secure implementation of χ^*-Weak coin)**.** *For $\chi^* \in [0,1]$ and $\mu \in [0,1]$, let $\chi^+ = 1 - \mu(D)(1-\chi^*)$ and $\chi^- = \mu(D)\chi^*$. A protocol π is said to be a μ-secure implementation of χ^* weak coin-flipping, if the outcome is a χ^*-coin if both parties follow the protocol honestly, and either*

1. (Secure when Alice wants 1 and Bob wants 0) *For any efficient (PPT) adversarial Alice strategy, the expected outcome of the protocol when playing against the honest Bob strategy is no higher than χ^+ and for any efficient Bob strategy, the expected outcome when playing against the honest Alice strategy is no lower than χ^-, or*

2. (Secure when Alice wants 0 and Bob wants 1) *For any efficient (PPT) adversarial Alice strategy, the expected outcome of the protocol when playing against the honest Bob strategy is no lower than χ^- and for any efficient Bob strategy, the expected outcome when playing against the honest Alice strategy is no higher than χ^+.*

With increasing μ, the protocol has a better security guarantee: if $\mu = 1$, then neither party can bias the coin away from χ^* towards their desired outcome. Against an adversary with access to a $\mathbf{PSPACE}$ oracle, it is easy to see that any efficient protocol is 0-secure. Our attack in Section 3 renders any protocol about $\frac{1}{2}$-secure; for $\chi^* = \frac{1}{2}$, this means that some party can bias the protocol to about $\frac{1}{4}$ (if its desired outcome is 0) or to about $\frac{3}{4}$ (if its desired outcome is 1). (Our attack in Section 4 on the other hand renders any protocol μ secure with μ close to 0.)

In Section 4 we show that if a constant round weak coin-flipping protocol must be secure, then a standard weaker variant of one-way functions, called infinitely-often one-way functions must exist. This variant appears in earlier work like [OW93], but seems to have been named so in [HI07]. A polynomial time computable function $f : \{0,1\}^* \to \{0,1\}*$ is called an infinitely-often one-way function if for any polynomial p and any PPT adversary A, if for infinitely many values n, $\Pr_{x \leftarrow \{0,1\}^n}[f(A(f(x)) = f(x)] < \frac{1}{p(n)}$ (where the probability is also over the coins of A). Thus, if f is not an infinitely-often one-way function, then there is a PPT adversary A which *for all but finitely many values of* n has a significant probability of inverting f on random inputs from $\{0,1\}^n$.

3 Complexity of Weak Coin-Tossing

In this section we show our main result, that if there is a polynomial time weak coin-tossing protocol, then $\mathbf{NP} \not\subseteq \mathbf{BPP}$. In fact, we show that any weak coin-tossing protocol can be attacked significantly (biasing the outcome by close to 0.75) by polynomial time adversaries with access to an $\mathbf{NP}$ oracle. We arrive at this result in a few steps:

- First, we observe that for any polynomial time protocol π, there exists a *state-less* protocol π' that runs in polynomial time with access to an $\mathbf{NP}$ oracle, such that π' is "as secure as" π when considering adversaries with access to $\mathbf{NP}$ oracles. (Lemma 2.)
- Next we show that, unconditionally, any state-less protocol for weak coin-flipping can be attacked efficiently, using just the protocol itself as a black-box.

Together, these give an attack on any polynomial time protocol π, wherein the attack will use an $\mathbf{NP}$ oracle (which will be used to implement the state-less protocol π' that will be accessed as a black-box by the attack).

The first of these follows rather easily from a result on uniform generation of $\mathbf{NP}$-witnesses given an $\mathbf{NP}$ oracle [JVV86, BGP00]. (See Appendix B.) We remark that while much weaker computational power (namely inverting a one-way function) is enough for carrying out such a reconstruction in the normal course of the protocol, it is much harder to ensure that the reconstruction works with adequate accuracy even when the protocol is under attack and may result in a transcript distribution significantly different from that in the normal execution. However, by [JVV86, BGP00], such reconstruction can be accurately carried out for any transcript history when an $\mathbf{NP}$-oracle is given.

Our main work then is in showing an attack on a state-less protocol. Surprisingly we can do this efficiently using the protocol itself as a black-box, and with no further computational complexity assumption. In Section 3.1 we provide an intuition for the way the attack works, assuming we have certain additional oracles related to the protcol. Then in Section 3.2 we present the actual attack, which involves additional checks to make the simpler attack robust, and then replaces the oracles it required by approximate implementations.

3.1 A Simplified Sketch of the Attack

In this section we describe an attack on any weak coin flipping protocol π, given an oracle that, for any partial transcript v, can return χ_v, the color of v in the protocol π.

Given such an oracle for the colors, we define four attacks, two each for corrupt Alice and corrupt Bob — for each party, one to bias the outcome towards 0 and one to bias it towards 1. In Figure 1, we describe the attack for Alice to bias the outcome towards 1; the other attacks are symmetric.

Intuition of Attack $\mathsf{Adv}_A^{(1)}$

A D round protocol π with for a χ^*-coin is given. We have access to an oracle which provides the exact color χ_v of any node v.

Suppose the protocol is currently at an Alice-node v (i.e., the next message is sent by Alice). Let $v0$ and $v1$ be its two children. For convenience we write χ, χ_0 and χ_1 respectively for χ_v, χ_{v0}, and χ_{v1}. Let $p_0 = \Pr_\pi[v0|v]$ and $p_1 = \Pr_\pi[v1|v]$ so that $p_0 + p_1 = 1$ and $\chi = p_0\chi_0 + p_1\chi_1$. Given χ, χ_0 and χ_1 we can calculate p_0 and p_1.

- Let $t_b = \frac{p_b\chi_b\left(1-\chi_{(1-b)}\right)}{(\chi-\chi_0\chi_1)}$, for $b \in \{0,1\}$. Send 0 as the next message with probability t_0 and 1 as the next message with probability t_1.

Figure 1: *Intuition of Attack $\mathsf{Adv}_A^{(1)}$ for Alice to bias towards outcome 1.*

Note that indeed $t_0 + t_1 = \frac{(p_0\chi_0+p_1\chi_1)-(p_0+p_1)\chi_0\chi_1}{\chi-\chi_0\chi_1} = 1$, since $p_0\chi_0 + p_1\chi_1 = \chi$ and $p_0 + p_1 = 1$.

We shall show that our choice of the probabilities t_0 and t_1 are such that no matter what the protocol is, these attacks break the security of the protocol. More precisely, if we denote the four attacks by $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ (with $\mathsf{Adv}_A^{(0)}$ corresponding to Alice trying to bias towards 0 and so on), we show that in any such protocol either $\mathsf{Adv}_A^{(1)}$ biases the outcome to 1 with probability at least 0.75, or $\mathsf{Adv}_B^{(0)}$ biases the outcome to 0 with probability at least 0.75. Also, either $\mathsf{Adv}_A^{(0)}$ biases the outcome to 0 or $\mathsf{Adv}_B^{(1)}$ biases the outcome to 1 with probability at least 0.75. Then, the protocol π is not a secure weak coin-flipping protocol.

In order to analyze these attacks, we will define the following functions to assign a score for the (failure of) performance in biasing the orginal color χ at a node to a value x when the goal is to bias towards a bit b: $s_b(x,\chi) := \frac{|b-x|}{|b-\chi|}$ (for $\chi \neq b$). That is,

$$s_1(x,\chi) = \frac{1-x}{(1-\chi)} \qquad\qquad s_0(x,\chi) = \frac{x}{\chi}$$

In addition, we define $s_0(0,0) := 0$ and $s_1(1,1) := 0$. Note that the lower the score, the better the performance in biasing towards b.

For any node v in the transcript tree, let $A^{(0)}(v), A^{(1)}(v), B^{(0)}(v), B^{(1)}(v)$ denote the colors induced at the node v by our four attacks. Then, we will show that:

$$A^{(1)}(v), B^{(1)}(v) \in [\chi_v, 1] \text{ and } A^{(0)}(v), B^{(0)}(v) \in [0, \chi_v] \tag{1}$$

$$s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 \tag{2}$$

$$s_0(A^{(0)}(v), \chi_v) + s_1(B^{(1)}(v), \chi_v) \leq 1 \tag{3}$$

Intuitively these two inequalities state that if Alice fails to bias the output to b by a significant amount then Bob can bias the output to $(1-b)$ by a significant amount. More precisely, when v is the root of a protocol that yields a fair coin under honest execution ($\chi_v = \frac{1}{2}$), the first equation above shows that either $s_1(A^{(1)}(v), \chi_v) \geq \frac{1}{2}$ (which implies that $A^{(1)}(v) \geq \frac{3}{4}$) or $s_0(B^{(0)}(v), \chi_v) \geq \frac{1}{2}$ (which implies that $B^{(0)}(v) \leq \frac{1}{4}$). That is, the protocol is not secure against an Alice who prefers 1 and a Bob who prefers 0. Similarly, the second equation shows that protocol is not secure when Alice and Bob prefer 0 and 1 respectively either.

We will prove the result by induction on the height h of S_v the subtree rooted at v. If $h = 1$, it is trivial to see that both the conditions are satisfied. Let the four tuple associated with the performance of our attack

on the S_{vb} be $(A_b^{(0)}, A_b^{(1)}, B_b^{(0)}, B_b^{(1)}) = (A^{(0)}(vb), A^{(1)}(vb), B^{(0)}(vb), B^{(1)}(vb))$. We will only show how the induction works for the first case, i.e. $s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1$. By induction hypothesis, we know that: $B_b^{(0)} \leq \chi_b$ and

$$\frac{1 - A_b^{(1)}}{(1 - \chi_b)} + \frac{B_b^{(0)}}{\chi_b} \leq 1 \implies (1 - A_b^{(1)}) \leq \left(1 - \frac{B_b^{(0)}}{\chi_b}\right)(1 - \chi_b).$$

(This inequality in fact holds for the extreme cases of $\chi_0 = 0$ and $\chi_0 = 1$ as well: when $\chi_0 = 1$, we have $A_1^{(0)} \in [\chi_0, 1] \implies A_1^{(0)} = 1$; when $\chi_0 = 0$, then $B_0^{(0)} = 0$ and our convention for the score will interpret $\frac{B_0^{(0)}}{\chi_0}$ as 0.)

Suppose v is an Alice node and she outputs b as the next message with probability t_b, where $b \in \{0, 1\}$. Then $A^{(1)}(v) = t_0 A_0^{(1)} + t_1 A_1^{(1)}$ and $B^{(0)}(v) = p_0 B_0^{(0)} + p_1 B_1^{(0)}$.

$$\begin{aligned} s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) &= \frac{1 - A^{(1)}(v)}{(1 - \chi)} + \frac{B^{(0)}(v)}{\chi} = \frac{t_0(1 - A_0^{(1)}) + t_1(1 - A_1^{(1)})}{(1 - \chi)} + \frac{p_0 B_0^{(0)} + p_1 B_1^{(0)}}{\chi} \\ &\leq B_0^{(0)} T_0 + B_1^{(0)} T_1 + \frac{t_0(1 - \chi_0) + t_1(1 - \chi_1)}{(1 - \chi)} \end{aligned}$$

where $T_0 = \left[\frac{p_0}{\chi} - \frac{t_0(1-\chi_0)}{(1-\chi)\chi_0}\right]$ and $T_1 = \left[\frac{p_1}{\chi} - \frac{t_1(1-\chi_1)}{(1-\chi)\chi_1}\right]$. If we show that $T_0 \geq 0$ and $T_1 \geq 0$, then indeed

$$\begin{aligned} s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) &\leq \chi_0 T_0 + \chi_1 T_1 + \frac{t_0(1 - \chi_0) + t_1(1 - \chi_1)}{(1 - \chi)} \\ &= \frac{p_0\chi_0 + p_1\chi_1}{\chi} = 1 \end{aligned}$$

(using the fact that $B_b^{(0)} \leq \chi_b$). Now, substituting $t_0 = \frac{p_0\chi_0(1-\chi_1)}{(\chi - \chi_0\chi_1)}$ and $t_1 = \frac{p_1\chi_1(1-\chi_0)}{(\chi - \chi_0\chi_1)}$, we observe that

$$T_0 = \frac{p_0\left[(1 - \chi)(\chi - \chi_0\chi_1) - \chi(1 - \chi_0)(1 - \chi_1)\right]}{\chi(1 - \chi)(\chi - \chi_0\chi_1)} = \frac{p_0(\chi_1 - \chi)(\chi - \chi_0)}{\chi(1 - \chi)(\chi - \chi_0\chi_1)} \geq 0$$

(using the fact that $\min\{\chi_0, \chi_1\} \leq \chi \leq \max\{\chi_0, \chi_1\}$), and similarly $T_1 \geq 0$.

Now we need to show that $A^{(1)}(v) \in [\chi, 1]$ and $B^{(0)}(v) \in [0, \chi]$. Note that if $\chi_0 \geq \chi_1$ then $t_0 \geq p_0$ and if $\chi_1 \geq \chi_0$, then $t_1 \geq p_1$. Hence we have $A^{(1)}(v) = t_0\chi_0 + t_1\chi_1 \geq p_0\chi_0 + p_1\chi_1 = \chi$. Also, since $B_0^{(0)} \leq \chi_0$ and $B_1^{(0)} \leq \chi_1$, we have $B^{(0)}(v) = p_0 B_0^{(0)} + p_1 B_1^{(0)} \leq p_0\chi_0 + p_1\chi_1 = \chi$. This completes the analysis of this simplified attack, which assumes t_0 and t_1 can be computed correctly.

Our actual attack is significantly complicated than the one explained in this section, by the fact that we do not have oracles to find χ_v exactly (even given an **NP** oracle). In fact, we can only estimate χ_v with a small additive error term. The effect of this error on our attack can be severe when χ is very close to 1 or $(\chi - \chi_0\chi_1)$ is very small. The actual attack takes care of these special cases separately.

3.2 The Actual Attack

We describe our actual attack against a stateless protocol in Appendix C. The attack closely follows the intuition above, but significantly differs in the details and the analysis. The differences arise from the fact that the above attack depended on accurately knowing certain ratios, which simply cannot be estimated sufficiently accurately.

As described in Figure 5, the attack has two additional checks before carrying out an approximate version of the above attack. Firstly, if the color of the current node is very close to 0 or 1, the attack continues by

simply following the protocol honestly (even if it later encounters nodes with different colors). Note that this is done even on reaching a node with the color opposite to what the attack desires. The second check is more subtle, and is designed to handle the technical difficulty in accurately estimating t_0 and t_1 when the denominator is close to 0. In the case of $\mathsf{Adv}_A^{(1)}$ (Alice biasing towards 1), this denominator is $\chi - \chi_0\chi_1$; if we see that χ is close to $\min\{\chi_0, \chi_1\}$, then the current step of the attack is changed to weigh the two children using the contribution (in the honest execution) that they make to the color of the current node, i.e., using probabilities $h_b = \frac{p_b\chi_b}{\chi}$ instead of t_b. If these checks pass, then the original attack (but with ratios calculated according to the estimated values) is carried out.

In Appendix C, first we describe the attack in terms of a couple of oracles; but using the fact that π is a stateless protocol, we show that we can indeed implement statistically close approximations of these oracles, using black-box access to (the next message function of) π. Also, the attack needs estimating various quantities with sufficient accuracy, which also can be carried out with black-box access to π.

In Appendix D, we prove the following theorem.

Theorem 1. *Let π be a D round stateless coin-flipping protocol with expected outcome (under honest execution) $\chi \in (0,1)$. For any function (of the security parameter) $0 < \epsilon < 1$ define $\chi^- = \chi - \frac{\chi}{2}(1-\epsilon)$, and $\chi^+ = \chi + \frac{(1-\chi)}{2}(1-\epsilon)$. Then there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ which use black-box access to π and run in poly$(\frac{1}{\epsilon} + D)$ time, such that*

1. *$A^{(1)} \geq \chi^+$ or $B^{(0)} \leq \chi^-$, (i.e., not secure if Alice wants 1 and Bob wants 0)*
2. *and, $B^{(1)} \geq \chi^+$ or $A^{(0)} \leq \chi^-$ (i.e., not secure if Bob wants 1 and Alice wants 0).*

where $A^{(b)}$ (resp. $B^{(b)}$), for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

As a corollary of Theorem 1, we can conclude that:

Corollary 2. *If a stateless protocol π is a μ-secure polynomial time implementation of χ^*-weak coin for any constant $0 < \chi^* < 1$, then $\mu \leq \frac{1}{2} + negl(D)$, where D is the number of rounds in π and negl is a negligible function.*

Note that an ideal secure protocol for weak coin-flipping would be a 1-secure protocol. Relative to adversaries with access to a $\mathbf{PSPACE}$ oracle, all protocols are 0-secure protocols. Our attack is not as effective as an attack with a $\mathbf{PSPACE}$ oracle, but instead renders any (stateless) protocol at most $\frac{1}{2} + \text{negl}(D)$-secure. Theorem 1 is proven in Appendix D. Here we give a brief summary. At the heart of the analysis is an analogue of the inductively maintained inequalities equation (1)-equation (3). These inequalities had depended on the fact the we could arrange the quantities T_0 and T_1 to be positive. Unfortunately, this is no more the case in the analysis of the actual attack. But by carefully choosing our parameters, we can carry out a case analysis and prove Lemma 3. Note that we described the attack assuming access to oracles Π_H and Π_T in addition to Π, and required estimates of various quantities. To complete the proof we show how to estimate these values required by our attack (Lemma 4) and implement (good approximations of) the oracles Π_T and Π_H (Lemma 5) using black-box access to the protocol π. Finally, in Appendix D.2 we give the choice of parameters to conclude the proof of Theorem 1.

Finally, from Theorem 1 and Lemma 2 we obtain our main result.

Theorem 3. *Let π be a polynomial time (possibly stateful) coin-flipping protocol with expected outcome (under honest execution) $\chi \in (0,1)$. For any function (of the security parameter) $0 < \epsilon < 1$ define $\chi^- = \chi - \frac{\chi}{2}(1-\epsilon)$, and $\chi^+ = \chi + \frac{(1-\chi)}{2}(1-\epsilon)$. Then there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ which use an $\mathbf{NP}$ oracle, but otherwise run in poly$(\frac{1}{\epsilon} + D)$ time, such that at*

1. $A^{(1)} \geq \chi^+$ *or* $B^{(0)} \leq \chi^-$,

2. *and,* $B^{(1)} \geq \chi^+$ *or* $A^{(0)} \leq \chi^-$,

where $A^{(b)}$ *(resp.* $B^{(b)}$*), for* $b \in \{0,1\}$*, is the expectation of the outcome when Alice runs the attack* $\mathsf{Adv}_A^{(b)}$ *against honest Bob (resp. Bob runs* $\mathsf{Adv}_B^{(b)}$ *against honest Alice).*

Proof Sketch. Consider the stateless protocol π' guaranteed by Lemma 2. (This protocol is polynomial time given an $\mathbf{NP}$ oracle.) By Theorem 1, there are adversaries $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_B^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(1)}$, which attack π', and access π' as a black-box. Thus these adversaries can be implemented in polynomial time, given an $\mathbf{NP}$ oracle. By Lemma 2 (or rather, its proof) these adversaries have the same advantage with π as with π', and hence one of the four conditions stated in Theorem 1 hold with respect to π and these adversaries. Note that these are the same conditions described above, arranged differently. □

Finally, note that if $\mathbf{NP} \subseteq \mathbf{BPP}$, then a highly accurate $\mathbf{NP}$ oracle can be implemented in probabilistic polynomial time, and hence the adversaries in the above theorem can be converted to PPT adversaries.

Corollary 4. *If* π *is a* μ*-secure polynomial time implementation of* χ^**-weak coin for any constant* $0 < \chi^* < 1$*, then unless* $\mathbf{NP} \not\subseteq \mathbf{BPP}$*,* $\mu \leq \frac{1}{2} + negl(D)$*, where* D *is the number of rounds in* π *and negl is a negligible function.*

In particular, if there is a weak coin-flip protocol for a coin of bias $\frac{1}{2}$ which is μ-secure with $\mu > \frac{1}{2} + \alpha$ for some non-negligible function α (corresponding to limiting the bias approximately to the range $[\frac{1}{4}, \frac{3}{4}]$), then $\mathbf{NP} \not\subseteq \mathbf{BPP}$.

4 Constant Round Weak Coin-Flipping

In this section we show a much stronger intractability implication of a weak coin-flipping protocol with a very weak unbiasability guarantee, if the protocol has only constantly many rounds. Note that we do allow the communication complexity of the protocol to be polynomial. We show the following result:

Theorem 5. *If infinitely-often one-way functions do not exist then for any constant round coin-tossing protocol* π*, there exist attacks* $\mathsf{Adv}_A^{(0)}$*,* $\mathsf{Adv}_A^{(1)}$*,* $\mathsf{Adv}_B^{(0)}$ *and* $\mathsf{Adv}_B^{(1)}$*, such that for any* $\epsilon = 1/poly(k)$ *(*$0 < \epsilon < 1$*), the attacks run in polynomial time in* k*, and for sufficiently large* k*:*

1. $\min\left\{1 - A^{(1)}, B^{(0)}\right\} \leq \epsilon$*, and*

2. $\min\left\{A^{(0)}, 1 - B^{(1)}\right\} \leq \epsilon$,

where $A^{(b)}$ *(resp.* $B^{(b)}$*) for* $b \in \{0,1\}$*, is the expectation of the outcome when Alice runs the attack* $\mathsf{Adv}_A^{(b)}$ *against honest Bob (resp. Bob runs* $\mathsf{Adv}_B^{(b)}$ *against honest Alice).*

In other words, if infinitely-often one-way functions do not exist then with probability close to 1 either Alice can bias the outcome to b or Bob can bias it to $(1 - b)$ starting from any transcript prefix v.

The attack has the following intuitive form: use the fact that any polynomial time function can be inverted to implement next message function oracle for the protocol. At any point in the protocol, use this to sample a polynomial-sized sub-tree of the protocol (with the density of children sampled at each node increasing with depth), and run the $\mathbf{PSPACE}$ attack on this sampled tree to decide on the next move in the attack. While conceptually simple, this idea runs into two complications.

- At each round, the response from the honest party may not fall within the sub-tree that was sampled for the attack at that round; and as such the original attack computed may have no further relevance, and no use in deciding the response in subsequent rounds. Further, the $\mathbf{PSPACE}$ attack involves evaluating a max-average tree, and by sampling it is quite possible to miss the maximum.

- During the attack the distributions on the nodes at each level of the tree can deviate significantly from that under the honest execution, and the next message function oracles need to work well on these distributions. These distributions depend on the behavior of the attack in the previous rounds, which however carries out recursive look-aheads (in implementing the **PSPACE** attack), and these look-aheads in turn involve accessing the next message function oracles. A simple attempt at implementing the next message function oracles can lead to circularity.

The second issue can be taken care of by carefully defining a family of next message function oracles, which not only depend on what depth in the protocol it is sampling a message for, but also on which iteration in the **PSPACE** attack it appears in.

The first issue is addressed by the following intuition: even though it is possible for (say) Alice to miss the maximum (i.e., the child where her advantage is maximum) by a large margin when sampling, this means that a random choice should cause Alice to perform badly; hence this node confers advantage to Bob who is trying to bias the coin in the opposite direction. The actual calculations include more details, and are given in Appendix E.

Implementing Inverters. Our attack is based on realizing an efficient algorithm I, called *inverter*, which can efficiently perform the following task: Given a partial transcript v, it outputs a k-bit message m such that $\Pr(m|v)$ is identical to the probability of π generating a transcript vm conditioned on the fact that v is generated as a partial transcript.

Alternately, if we are able to sample uniformly at random from the set of randomness R_v of pairs (r_A, r_B) such that Alice and Bob with local randomness r_A and r_B generate the partial transcript v when running the protocol π, then we can implement I. We shall reverse sample from the set R_v and run the protocol for one more round and obtain a transcript prefix vm.

We will show the following result:

Lemma 1. *If infinitely-often one-way functions do not exist, then, for sufficiently large k, there exists a class $\mathcal{I} = \{\tilde{I}_{i,j} | i \in [D] \text{ and } j \in [i]\}$ of efficient inverters, such that if Alice uses $\tilde{I}_{i,j}$ to invert v at height j when she is attacking the $(D-i+1)$-th round then the behavior of $\mathsf{Adv}_A^{(1)}$ is at most $1/poly(k)$ different from the case when she uses the actual inverter I.*

Let $f(x) = y$ be a polynomial time function and $\mathcal{D}$ be the distribution of $f(x)$ when x is uniformly sampled. If one-way functions do not exist, then there exists an efficient algorithm A such that $A(y)$ is $1/k^c$ close to the uniform distribution when y is sampled according to the distribution $\mathcal{D}$. Note that the guarantee is only for the distribution $\mathcal{D}$ and not for any arbitrary distribution.

So, we can not use this result to directly create an inverter. The main observation is that the set of nodes $Q_{i,j}$ at height j required to invert when Alice is attacking the i-th round could be performed simultaneously. In other words, they only depend on the nodes inverted while attacking $i' > i$ rounds or at higher levels $j' > j$. So, we define the the execution of $\mathsf{Adv}_A^{(1)}$ just before it inverts $Q_{i,j}$ as the function f. Now, the inverter $I_{i,j} = A$ could be used to invert all partial transcripts in $Q_{i,j}$.

It is worth mentioning that the time complexity of $I_{i,j}$ is only guaranteed to be polynomial in the time complexity of all the inverters $\{I_{i',j'} | i' > i \text{ or } j' > j\}$. So, the time complexity of the inverter $I_{1,1}$ turns out to be $k^{\Theta(1)^D}$, which is polynomial if and only if D is constant. Therefore, this approach works only when D is a constant. For details see Appendix E.

Acknowledgements. We are grateful to Russell Impagliazzo for proposing this intriguing area of investigation, and a number of useful conversations. We also thank Boaz Barak and Yael Kalai for collaboration at an early stage of this research.

References

[B82] Manuel Blum. Coin flipping by phone. In *Proc.* 24*th IEEE Computer Conference (CompCon)*, pages 133–137, 1982. See also *SIGACT News*, Vol. 15, No. 1, 1983.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.

[BGP00] M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of NP-witnesses using an NP-oracle. *Information and Computation*, 163, 2000.

[CI93] Richard Cleve and Russell Impagliazzo. Martingales, collective coin flipping and discrete control processes, 1993.

[G04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proc.* 21*st STOC*, pages 25–32. ACM, 1989.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play ANY mental game. In ACM, editor, *Proc.* 19*th STOC*, pages 218–229. ACM, 1987. See [G04, Chap. 7] for more details.

[HI07] Edward A. Hirsch and Dmitry Itsykson. An infinitely-often one-way function based on an average-case assumption. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(117), 2007.

[I] Russell Impagliazzo. 5 worlds of problems. Talk at the workshop Complexity and Cryptography: Status of Impagliazzo's Worlds, Princeton, NJ. Video available from http://intractability.princeton.edu/videos/stream/videoplay.html? videofile=cs/IW2009-500kb/Russel%20Impagliazzo.mp4, year = 2009.

[I10] Russell Impagliazzo. Personal communication, 2010.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *Proc.* 30*th FOCS*, pages 230–235. IEEE, 1989.

[JVV86] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

[KN04] Iordanis Kerenidis and Ashwin Nayak. Weak coin flipping with small bias. *Inf. Process. Lett.*, 89(3):131–135, 2004.

[MNS09] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. Technical Report TR-93-073, International Computer Science Institute, Berkeley, CA, November 1993. Preliminary version in Proc. 2nd Israeli Symp. on Theory of Computing and Systems, 1993, pp. 3–17.

A Examples

Greedy does not work well. Greedy strategy is one of the basic strategies to bias the outcome towards b. At a partial transcript v, output a bit d such that the color χ_{vd} is closer to b than $\chi_{v(1-d)}$. But this strategy is not good and we explicitly construct a protocol tree where we can make the bias obtained by the greedy algorithm arbitrarily small. Recall that if A/χ_v is written at a node, it means that Alice is supposed to send the next message after the partial transcript v is generated and the subtree S_v computes a χ_v-coin when both parties are honest. For simplicity, when we explain the transcript tree construction, we do not insist that Alice and Bob nodes alternate. If an Alice node v follows an Alice node v', then we can assume that there is a dummy Bob node v'' such that whatever bit is sent at v'' it does not not effect the outcome. We can assume that both children of v'' are identical to v'.

Consider the following recursive graph construction (Figure 2):

1. For the base case of $k = 0$, define G_0 as the tree where Alice announces the outcome of the $\frac{1}{2}$-coin.
2. For any other $k > 0$, we recursively define G_k using G_{k-1}. The root node v is an Alice node that implements a $\frac{1}{2}$-coin. Its two children $v0$ and $v1$ are Bob nodes which implement $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$-coins respectively. Nodes $v00$ is an Alice node implementing $\frac{1}{2} - 2\epsilon$-coin and $v11$ is a Bob node implementing $\frac{1}{2} + 2\epsilon$ coin. The S_{v01} and S_{v10} are the tree G_{k-1}.

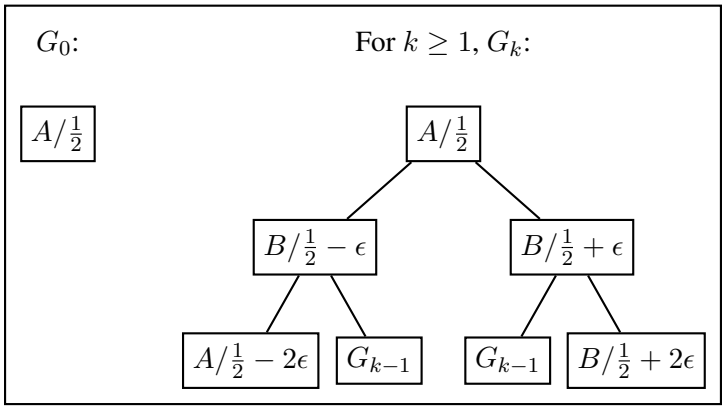


Figure 2: Greedy strategy is not good.

Let $g_b^A(k)$ (resp. $g_b^B(k)$) be the expectation of the outcome when Alice (resp. Bob) is trying to bias the outcome to b in G_k.

$$g_1^A(k) = \frac{1}{2}g_1^A(k-1) + \left(\frac{1}{4} + \epsilon\right) = \frac{1}{2} + 2\epsilon + \frac{\Theta(1)}{2^k}$$
$$g_0^B(k) = \frac{1}{2}g_0^B(k-1) + \left(\frac{1}{4} - \epsilon\right) = \frac{1}{2} - 2\epsilon + \frac{\Theta(1)}{2^k}$$

By symmetry, $1 - g_0^A(k) = g_1^A(k)$ and $1 - g_1^B(k) = g_0^B(k)$. We see that we can drive the bias obtained by the greedy algorithm to negligibly close to $\frac{1}{2}$.

Let $h_b^A(k)$ (resp. $h_b^B(k)$) be the expectation of the outcome when Alice (resp. Bob) is trying to bias the

outcome to b in G_k. We can write the following recurrence:

$$\begin{aligned} h_1^A(k) &= \frac{1}{2}h_1^A(k-1) + \left(\frac{3}{8} + \frac{2\epsilon^2(1-\epsilon)}{(1+4\epsilon^2)}\right)\left(\mathsf{Adv}_A^{(1)} \text{ on } G_k\right) \\ &= \frac{3}{4} + \frac{4\epsilon^2(1-\epsilon)}{(1+4\epsilon^2)} + \frac{\Theta(1)}{2^k} \\ h_0^B(k) &= \frac{1}{2}h_0^B(k-1) + \left(\frac{1}{8} - 2\epsilon^2\right)\left(\mathsf{Adv}_B^{(0)} \text{ on } G_k\right) \\ &= \frac{1}{4} - 4\epsilon^2 + \frac{\Theta(1)}{2^k} \end{aligned}$$

This shows that our attack achieves nearly close to $3/4$ and $1/4$ bias.

Need to attack at more than constant rounds. Consider the recursive graph construction as shown in Figure 3. Fix a particular k and consider the tree G_k such that the probability that the honest protocol reaches any $A/\frac{1}{2}$ leaf is ϵ and the probability of reaching any $B/\frac{1}{2}$ leaf is $\frac{1}{k+1} - \frac{k\epsilon}{k+1}$. If Bob is honest then Alice can not bias the outcome by more than $k\epsilon$, which can be made arbitrarily small. If Bob is not honest and he attacks at only c rounds, then the maximum bias he could generate is $\frac{c}{k+1} - \frac{kc\epsilon}{k+1}$. So, to go beyond $1/\text{poly}(k)$ bias, Bob needs to attack at more than constant number of rounds.

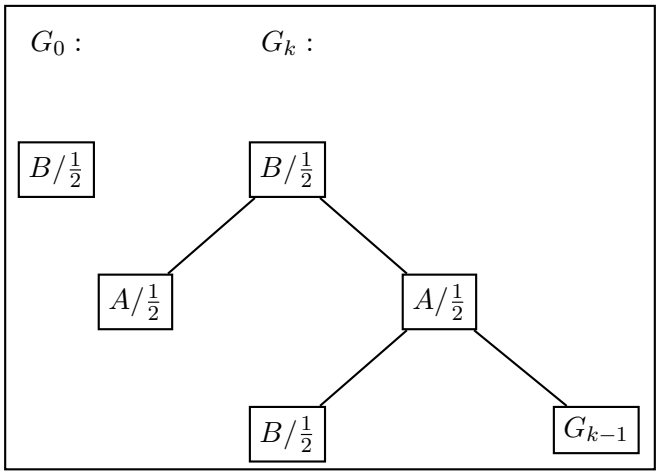


Figure 3: Attacking constant number of grounds does not help.

Beyond $1/4$ bias. Consider the class of adversarial strategies which uses only χ_v, χ_{v0}, χ_{v1} and $|v|$ to determine the distribution according to which the next bit is sampled. We will show that any adversary in this class can not bias by more than $1/4$. Consider the performance of such adversaries on the graph in Figure 4. Suppose the probability of reaching any child vb from a node v in the honest protocol is $\frac{1}{2}$. If Bob is not honest, then with probability $\frac{1}{2}$ it can decide the outcome of a $\frac{1}{2}$-coin. So, the maximum bias it can obtain is $1/4$. Now, if Alice is not honest then she can reach a leaf $A/\frac{1}{2}$ coin with probability $\frac{1}{2}$ independent of her strategy. So, she can obtain a maximum bias of $1/4$.

If we expand the class of adversaries to include any adversary with constant look ahead in the protocol tree, then we can generalize the graph in Figure 4 so that they can obtain at most $1/4$ bias. The class of adversaries we have considered currently can be interpreted as 1-look ahead adversaries. Suppose, we introduce c redundant levels between any two levels of the graph in Figure 4. Then any adversary with $(c+1)$ look ahead in the protocol tree can not obtain more than $1/4$ bias. Note that by adding dummy nodes, even if the adversary's strategy looks ahead a bounded depth or tries to take into account whose turn is next, this cannot help it achieve a better bias.

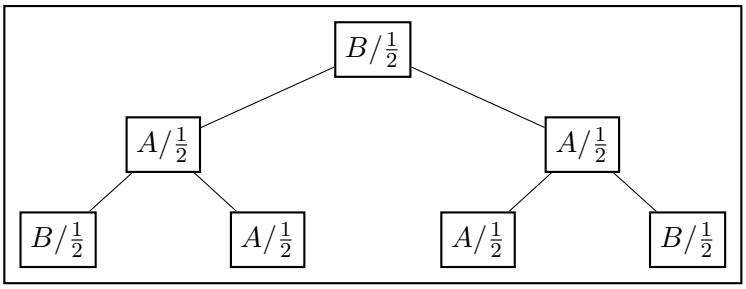


Figure 4: Certain class of local algorithms can not go beyond $1/4$ bias.

B Private State is Not Useful

Access to an $\mathbf{NP}$-oracle can be used to reconstruct a correctly distributed random tape for a party in a protocol, using just the public history of the protocol. We remark that while much weaker computational power (namely inverting a one-way function) is enough for carrying out such a reconstruction in the normal course of the protocol, it is much harder to ensure that the reconstruction works with adequate accuracy even when the protocol is under attack and may result in a transcript distribution significantly different from that in the normal execution. However, by a result on uniform generation of $\mathbf{NP}$-witnesses given an $\mathbf{NP}$ oracle [JVV86, BGP00], such reconstruction can be accurately carried out for any transcript history when an $\mathbf{NP}$-oracle is given. More formally, we need the following result.

Lemma 2. *For any polynomial time protocol π for χ^* weak coin-flipping that is μ-secure against polynomial time adversaries with access to $\mathbf{NP}$ oracles, there is a* state-less *protocol π' that runs in (expected) polynomial time with access to an $\mathbf{NP}$ oracle, and is also a χ^* weak coin-flipping that is μ-secure against polynomial time adversaries with access to $\mathbf{NP}$ oracles.*

Proof Sketch. We shall in fact show a much more general result here: π can be any arbitrary polynomial time protocol (not necessarily for coin-flipping) and we can consider its behavior against any arbitrary class of adversaries. We shall describe a protocol π' which is executed with the help of an $\mathbf{NP}$ oracle, such that any adversary's view when interacting with honest players running π and those running π' are identical (or statistically close, if using a strict polynomial time implementation of π').

To define π' we shall use the result of Bellare et al. [BGP00] that for any $\mathbf{NP}$ relation $R(\cdot;\cdot)$, there is an expected polynomial time algorithm S_R^{SAT} (with access to the oracle for SAT) such that given x, it samples an element uniformly from the set $R^{-1}(x) := \{w | R(x;w) = 1\}$, provided that this set is non-empty.

Define $R_A(v; r_A) = 1$ iff v is a prefix of the transcript generated when Alice executes the protocol π with a random tape r_A and receives responses from Bob consistent with v. Note that this is indeed an $\mathbf{NP}$ relation[5] since Alice's program is polynomial time. Hence, by [BGP00], $S_{R_A}^{\mathsf{SAT}}(v)$ outputs a random tape r_A uniformly from $R_A^{-1}(v)$. Similarly define R_B and S_{R_B}.

Define π' to be the stateless protocol in which Alice and Bob behave as follows: on being given a transcript prefix v, Alice uses $S_{R_A}^{\mathsf{SAT}}(v)$ to sample a random tape r_A, internally simulates the stateful protocol π with this random tape, and responses from Bob as given in v, and outputs the next bit after v in the transcript so generated. Bob behaves symmetrically, using S_{R_B} instead.

Then, the protocol tree defined for π' is identical to that of π, and for any adversary, the view on attacking π' is the same as that of attacking π.

□

[5] More formally we can include the security parameter on both arguments to R to ensure that it is polynomially balanced.

C Attack on Stateless Protocols

In this section we describe our actual attack against a stateless protocol, which follows the intuition of the attack in Section 3.1, but does not use exact color oracles. Instead, the attack uses only the protocol itself as a black-box.

Let π be a protocol for coin with bias $\chi^* \in (0,1)$ (henceforth, called a χ^*-coin). For convenience first we shall describe the attack using a few oracles related to the π:

1. Π: Given a partial transcript v as input, it outputs the next message (bit) as specified by the protocol π. (Recall that the protocol is stateless.)

2. Π_H: Given a partial transcript v as input, it samples a transcript τ as generated by the protocol, conditioned on v being a prefix of τ and the outcome of the protocol at τ being 1 (heads). Then it outputs the bit b such that vb is a prefix of τ.

3. Π_T: Given a partial transcript v as input, it samples a transcript τ as generated by the protocol, conditioned on v being a prefix of τ and the outcome of the protocol at τ being 0 (tails). Then it outputs the bit b such that vb is a prefix of τ.

Attack $\mathsf{Adv}_A^{(1)}$

A D round protocol π with bias χ^* (along with corresponding oracles Π and Π_H) is given. The attack is parametrized by a function of the security parameter $0 < \epsilon < 1$. Let $\delta = \min\left\{\frac{(1-\chi^*)\epsilon}{4}, \frac{\chi^*\epsilon}{4}\right\}$ and $\lambda = \min\left\{\frac{\delta^3}{3^6 D}, \frac{\epsilon^3\delta^3}{(72)^3 D^3}, \frac{1}{2^9}\right\} = \frac{\epsilon^3\delta^3}{(72)^3 D^3}$.

Alice performs the following attack at all nodes v where she is supposed to send the next bit. First, compute the following estimates, as described in Lemma 4. Let $\tilde{\chi}$ be an estimate of $\chi := \chi_v$, so that $|\tilde{\chi} - \chi| \leq \lambda$ w.h.p.. Similarly, let $\tilde{\chi}_0$ and $\tilde{\chi}_1$ be estimates of $\chi_0 := \chi_{v0}$ and $\chi_1 := \chi_{v1}$ respectively. Then proceed as follows:

1. If $\tilde{\chi} \geq 1 - (\delta + \lambda)$ or $\tilde{\chi} \leq (\delta + \lambda)$: Henceforth, follow the protocol honestly by making calls to Π. This case takes care of nodes in the transcript tree such that χ is too close to 0 or 1.

2. Else, if $\tilde{\chi} - \min\{\tilde{\chi}_0\, \tilde{\chi}_1\} < \lambda^{1/3} + 2\lambda$, then output $d = \Pi_H(v)$ as the next message. This case takes care of nodes in the transcript tree such that $\chi - \chi_0\chi_1$ is too small.

3. Else (here, $\chi \in [\delta, 1-\delta]$ and $\chi - \min\{\chi_0, \chi_1\} \geq \lambda^{1/3}$), we perform a variant of our original attack. Let $c' \in \{0,1\}$ be such that $\min\{\tilde{\chi}_0, \tilde{\chi}_1\} = \tilde{\chi}_{c'}$ (i.e., the child with lower estimated probability of heads). Let p_0 and p_1 be the probabilities assigned by π to the two possible next messages at v, so that $p_0 + p_1 = 1$ and $\chi = p_0\chi_0 + p_1\chi_1$. Let $\tilde{h}_{c'}$ be an estimation of $h_{c'} = \frac{p_{c'}\chi_{c'}}{\chi}$ such that $\left|\tilde{h}_{c'} - h_{c'}\right| \leq 3\lambda^{1/3}$ (see Lemma 4). Evaluate $\tilde{t}_{c'}$ which is an approximation of

$$t_{c'} = \frac{p_{c'}\chi_{c'}\left(1 - \chi_{(1-c')}\right)}{(\chi - \chi_0\chi_1)},$$

such that $\left|\tilde{t}_{c'} - t_c\right| \leq 9\lambda^{1/3}$ (Lemma 4). Set $\tilde{r}_{c'} = \min\{\tilde{t}_{c'}, \max\{0, \tilde{h}_{c'} - 3\lambda^{1/3}\}\}$.

Send the bit c' with probability $\tilde{r}_{c'}$ and send $1 - c'$ with probability $1 - \tilde{r}_{c'}$.

Figure 5: *Attack $\mathsf{Adv}_A^{(1)}$ for Alice to bias towards outcome 1.*

We will define the four attacks $\mathsf{Adv}_A^{(b)}$ and $\mathsf{Adv}_B^{(b)}$, where $b \in \{0,1\}$ as before: $\mathsf{Adv}_A^{(b)}$ is an algorithm which will provide Alice with a strategy to bias the output towards b. Similarly, $\mathsf{Adv}_B^{(b)}$ will provide a strategy for Bob to bias the output towards b. In Figure 5, we explicitly define $\mathsf{Adv}_A^{(1)}$ and all other attacks can be symmetrically defined: algorithm $\mathsf{Adv}_A^{(0)}$ is obtained from $\mathsf{Adv}_A^{(1)}$ by interchanging the interpretations of 1 (Heads) and 0 (Tails). And $\mathsf{Adv}_B^{(b)}$ is defined similarly where that attack is carried out at Bob-nodes in the protocol (i.e., where Bob sends the next bit of the transcript).

The attack refers to oracles Π_H and Π_T, and also estimates of various quantities. But (as we shall see) since π is a stateless protocol, we can indeed implement statistically close approximations of these oracles, and also obtain good estimates of the quantities used in the attack, simply using black-box access to the protocol π.

D Proof of Theorem 1

In this section we analyze the protocol in Appendix C and prove Theorem 1.

First, similar to the analysis of our simpler attack, we shall prove a lower-bound on the sum of the scores of a pair of attacks.

Lemma 3. *For a stateless weak coin-flipping protocol π, if v is a node in the protocol tree at height h, we have $A^{(0)}(v), B^{(0)}(v) \in [0, \chi_v]$ and $A^{(1)}(v), B^{(1)}(v) \in [\chi_v, 1]$; and*

$$s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$$
$$s_0(A^{(0)}(v), \chi_v) + s_1(B^{(1)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$$

where $A^{(b)}(v)$ (resp. $B^{(b)}$), for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice), and $\nu_0 = 0$ and $\nu_{h+1} = \frac{9\lambda^{1/3}}{\delta} + \nu_h \left(1 + \frac{9\lambda^{1/3}}{\delta}\right)$.

We will just prove the first part of the result, i.e. $s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$. We will proceed by induction on the height of v (i.e., height of S_v, the sub-tree rooted at v). It is easy to see that for the base case of $h = 1$, the result is true since $\chi_v \in \{0,1\}$ and then one of the two terms is 1 and the other is 0 (corresponding to the fact that one of Alice and Bob has zero advantage (and hence score 1) in biasing to their desired value, while for the other party, the outcome is completely biased to their desired value).

Suppose v has height $(h+1)$ and we will use the notation $\chi = \chi_v$, $\chi_0 = \chi_{v0}$ and $\chi_1 = \chi_{v1}$. Let $\chi_{low} = \min\{\chi_0, \chi_1\} = \chi_c$ and $\chi_{high} = \max\{\chi_0, \chi_1\} = \chi_{(1-c)}$, where $c \in \{0,1\}$. If p_0 and p_1 are the probabilities that the next bit after v is 0 and 1, respectively, then we can express $\chi = p_c \chi_{low} + p_{(1-c)} \chi_{high}$. The four tuple summarizing the performance of our attack on the protocol on S_{vb} be $(A_b^{(0)}, A_b^{(1)}, B_b^{(0)}, B_b^{(1)}) = (A^{(0)}(vb), A^{(1)}(vb), B^{(0)}(vb), B^{(1)}(vb))$. By induction hypothesis, we have the following constraint:

$$s_1(A_b^{(1)}, \chi_b) + s_b(B_b^{(0)}, \chi_b) = \frac{1 - A_b^{(1)}}{(1-\chi_b)} + \frac{B_b^{(0)}}{\chi_b} \leq 1 + \frac{\delta}{(1-\chi_b)} + \frac{\delta}{\chi_b} + \nu_h$$
$$\implies (1 - A_b^{(1)}) \leq \left[1 + \frac{\delta}{(1-\chi_b)} + \frac{\delta}{\chi_b} + \nu_h - \frac{B_b^{(0)}}{\chi_b}\right](1-\chi_b)$$

Suppose, Alice is expected to send the bit after v is generated in the protocol. She decides to send 0 with probability $\tilde{r}_0$ and 1 with probability $\tilde{r}_1 = 1 - \tilde{r}_0$. Now, $A^{(1)}(v) = \tilde{r}_0 A_0^{(1)} + \tilde{r}_1 A_1^{(1)}$ and $B^{(0)}(v) =$

$p_0 B_0^{(0)} + p_1 B_1^{(1)}$. We define the following quantity E:

$$\begin{aligned}
E &= s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) = \frac{1 - A^{(1)}(v)}{(1-\chi)} + \frac{B^{(0)}(v)}{\chi} \\
&= \frac{\tilde{r}_0(1 - A_0^{(1)}) + \tilde{r}_1(1 - A_1^{(1)})}{(1-\chi)} + \frac{p_0 B_0^{(0)} + p_1 B_1^{(0)}}{\chi} \\
&\leq B_0^{(0)} \left[\frac{p_0}{\chi} - \frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)\chi_0}\right] + B_1^{(0)} \left[\frac{p_1}{\chi} - \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)\chi_1}\right] + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} \\
&\quad + \frac{(\tilde{r}_0 + \tilde{r}_1)\delta}{(1-\chi)} + \frac{\delta}{(1-\chi)} \left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right) + \nu_h \left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right)
\end{aligned}$$

Let $\tilde{T}_0 = \left[\frac{p_0}{\chi} - \frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)\chi_0}\right]$ and $\tilde{T}_1 = \left[\frac{p_1}{\chi} - \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)\chi_1}\right]$. Let $\frac{\delta}{\chi'} = \frac{\delta}{(1-\chi)} \left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right)$ and $\nu_h' = \nu_h \left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right)$. We define r_b^* as the value of $\tilde{r}_b$ such that $\tilde{T}_b = 0$. It is impossible to have $\tilde{T}_0, \tilde{T}_1 < 0$ because $r_0^* + r_1^* > 1$ (Lemma 6). So, there are only three cases to consider:

1. If $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 \geq 0$, then

$$\begin{aligned}
E &\leq \chi_0 \tilde{T}_0 + \chi_1 \tilde{T}_1 + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' \\
&= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' = E^{(+,+)}
\end{aligned}$$

2. If $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 < 0$, then:

$$\begin{aligned}
E &\leq \chi_0 \tilde{T}_0 + 0 \cdot \tilde{T}_1 + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' \\
&= \frac{p_0 \chi_0}{\chi} + \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' \\
&= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \left(\frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)} - \frac{p_1 \chi_1}{\chi}\right) + \nu_h' \\
&= E^{(+,-)}
\end{aligned}$$

3. Similarly, if $\tilde{T}_0 < 0$ and $\tilde{T}_1 \geq 0$, then:

$$\begin{aligned}
E &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \left(\frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)} - \frac{p_0 \chi_0}{\chi}\right) + \nu_h' \\
&= E^{(-,+)}
\end{aligned}$$

In our attacks, we intend to use $\tilde{r}_c \leq h_c = \frac{p_c \chi_c}{\chi}$ because:

$$\begin{aligned}
\frac{1}{\chi'} = \frac{1}{(1-\chi)} \left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right) &\leq \frac{1}{\chi} \\
\iff \quad \frac{\tilde{r}_0}{\chi_0} + \frac{\tilde{r}_1}{\chi_1} &\leq \frac{1}{\chi} \\
\iff \quad \tilde{r}_c \left(\frac{1}{\chi_c} - \frac{1}{\chi_{1-c}}\right) &\leq \left(\frac{1}{\chi} - \frac{1}{\chi_{1-c}}\right) \\
\iff \quad \tilde{r}_c &\leq \frac{p_c \chi_c}{\chi} = h_c
\end{aligned}$$

The three cases of our attack are analyzed below:

Case 0: Suppose $\tilde{\chi} \geq 1-(\delta+\lambda)$ or $\tilde{\chi} \leq (\delta+\lambda)$, then we know that w.h.p. $\chi \geq 1-(\delta+2\lambda)$ or $\chi \leq (\delta+2\lambda)$. In this case Lemma 8 shows that the induction goes through.

Case 1: In this case $\chi \in [\delta, 1-\delta]$ and $\chi - \chi_c \leq \lambda^{1/3} + 4\lambda$. We use $\tilde{r}_0 = \frac{p_0\chi_0}{\chi}$ and $\tilde{r}_1 = \frac{p_1\chi_1}{\chi}$ and Lemma 9 shows that the induction also works in this case.

Case 2: In this case $\chi \in [\delta, 1-\delta]$ and $\tilde{\chi} - \min\{\tilde{\chi}_0, \tilde{\chi}_1\} \geq \lambda^{1/3} + 2\lambda$. Observe that $|\chi_0 - \chi_1| \geq \lambda^{1/3} \geq 2\lambda$. So, $\tilde{\chi}_0 < \tilde{\chi}_1$ if and only if $\chi_0 \leq \chi_1$. Therefore, $c = c'$.

Since, $(\tilde{\chi} - \min\{\tilde{\chi}_0, \tilde{\chi}_1\}) \geq \lambda^{1/3} + 2\lambda$, we have $(\chi - \chi_0\chi_1) \geq \lambda^{1/3}$. So, we can use Lemma 4 to estimate t_c such that $\left|\tilde{t}_c - t_c\right| \leq 9\lambda^{1/3}$. Recall, $\tilde{r}_c = \min\left\{\tilde{t}_c, \max\left\{0, \tilde{h}_c - 3\lambda^{1/3}\right\}\right\} \leq h_c$ (Lemma 11). Now, Lemma 12 implies that $|\tilde{r}_c - t_c| \leq 9\lambda^{1/3}$ and Lemma 10 shows that the induction works for this case.

Observe that in all cases of our attack, we used $\tilde{r}_c \leq h_c = \frac{p_c\chi_c}{\chi} \leq p_c$ and hence $A^{(1)}(v) = \tilde{r}_c A^{(1)}_c + \tilde{r}_{(1-c)} A^{(1)}_{(1-c)} \geq \tilde{r}_c\chi_c + \tilde{r}_{(1-c)}\chi_{(1-c)} \geq p_c\chi_c + p_{(1-c)}\chi_{(1-c)} = \chi$. And $B^{(0)}(v) = p_0 B^{(0)}_0 + p_1 B^{(0)}_1 \leq p_0\chi_0 + p_1\chi_1 = \chi$. This completes the proof of this lemma.

D.1 Estimating Quantities and Implementing Oracles Π_H and Π_T

Estimation of quantities. In our attack, we used estimations of χ, χ_b, p_b, $h_b = \frac{p_b\chi_b}{\chi}$ and $t_b = \frac{p_b\chi_b(1-\chi_{(1-b)})}{(\chi-\chi_0\chi_1)}$. The color of any node can be estimated by sampling N of transcripts which have v as their prefix, according to the honest distribution, and computing the average of all the outcomes. A random transcript can be generated by repeated invocation of the oracle Π. By simple Chernoff Bound, if $N = (D + 1/\epsilon)/\lambda^2$, then the difference between the estimated and actual colors is at most λ with probability $1 - \exp(-\Theta(D + 1/\epsilon))$. Similarly, to estimate p_b, invoke Π at v for N times and estimate p_b as the fraction of instances where b is obtained the next bit.

Estimation of h_b and t_b is performed by estimating the individual quantities in the expression and then using them for the calculation. For example, to estimate h_b we first compute $\tilde{p}_b$ (estimation of p_b), $\tilde{\chi}_b$ (estimation of χ_b) and $\tilde{\chi}$ (estimation of χ). Then we define $\tilde{h}_b = \frac{\tilde{p}_b\tilde{\chi}_b}{\tilde{\chi}}$. But the error in estimation could increase significantly if χ is very small. So, this method to estimate h_b should only be used when χ is larger than a particular threshold. Lemma 4 provides the exact details and bounds on these values:

Lemma 4 (Estimation)**.** *In the oracle world we can efficiently find $\tilde{\chi}$, $\tilde{\chi}_0$, $\tilde{\chi}_1$, $\tilde{p}_b$, $\tilde{h}_b$ and $\tilde{t}_b$ such that, w.h.p.:*

1. $|\tilde{\chi} - \chi|, |\tilde{\chi}_0 - \chi_0|, |\tilde{\chi}_1 - \chi_1|, |\tilde{p}_c - p_c| \leq \lambda$*, and*
2. $\left|\tilde{h}_b - h_b\right| \leq 3\lambda^{1/3}$*, if $\chi \geq \delta \geq \lambda^{1/3}$ and $\lambda \leq 1/3$.*
3. $\left|\tilde{t}_b - t_b\right| \leq 9\lambda^{1/3}$*, if $\chi - \chi_0\chi_1 \geq \lambda^{1/3}$ and $\lambda \leq \frac{1}{2^9}$.*

Proof. We provide the explicit mechanisms to evaluate these quantities.

1. Estimation of p_b: Call Π N times at node v. Let N_b be the number of times the output of the oracle is b. Define $\tilde{p}_b = N_b/N$. If $N = D/\lambda^2$ then w.h.p. $|\tilde{p}_b - p_b| \leq \lambda$.
2. Estimation of χ, χ_0 and χ_1: Using access to Π, sample N transcripts with prefix v. Let N_1 be the total number of transcripts where the outcome of the coin is 1. Define $\tilde{\chi} = N_1/N$. If $N = D/\lambda^2$ then w.h.p. $|\tilde{\chi} - \chi| \leq \lambda$. Similarly, we can also estimate $\tilde{\chi}_0$ and $\tilde{\chi}_1$.

3. Estimation of h_b: Compute $\tilde{p}_b$, $\tilde{\chi}_b$ and $\tilde{\chi}$ as approximations of p_b, χ_b and χ, such that $|\tilde{p}_b - p_b| \leq \lambda$, $|\tilde{\chi}_b - \chi_b| \leq \lambda$ and $|\tilde{\chi} - \chi| \leq \lambda$. Let $a_1 = p_b\chi_b$. Define the estimation of a_1 as $\tilde{a}_1 = \tilde{p}_b\tilde{\chi}_b$. We know that $|\tilde{a}_1 - a_1| \leq 3\lambda$. Define the estimation of $a_2 = \frac{1}{\chi}$ as $\tilde{a}_2 = \frac{1}{\tilde{\chi}}$. Then, we know that $|\tilde{a}_2 - a_2| \leq \frac{\lambda}{\delta(\delta-\lambda)} \leq 2\lambda/\delta^2 \leq 2\lambda^{1/3}$. Define $\tilde{h}_b = \tilde{a}_1\tilde{a}_2$. Now, $\left|\tilde{h}_b - h_b\right| \leq 2\lambda + 2\lambda^{1/3} \leq 3\lambda^{1/3}$.

4. Estimation of t_b: Suppose we compute $\tilde{\chi}$, $\tilde{\chi}_0$, $\tilde{\chi}_1$ and $\tilde{p}_b$ such that $|\tilde{\chi} - \chi| \leq \lambda$, $|\tilde{\chi}_0 - \chi_0| \leq \lambda$, $|\tilde{\chi}_1 - \chi_1| \leq \lambda$ and $|\tilde{p}_b - p_b| \leq \lambda$. First we will estimate $a_1 = p_b\chi_b\left(1 - \chi_{(1-b)}\right)$. Define $\tilde{a}_1 = \tilde{p}_b\tilde{\chi}_b\left(1 - \tilde{\chi}_{(1-b)}\right)$, then $|\tilde{a}_1 - a_1| \leq 7\lambda$. Next, we will estimate $a_2 = (\chi - \chi_0\chi_1)$. Define $\tilde{a}_2 = (\tilde{\chi} - \tilde{\chi}_0\tilde{\chi}_1)$, then $|\tilde{a}_2 - a_2| \leq 4\lambda$. Let $a_3 = 1/a_2$. If we define $\tilde{a}_3 = 1/\tilde{a}_2$, then:

$$\begin{aligned} |\tilde{a}_3 - a_3| &\leq \frac{4\lambda}{\lambda^{1/3}\left(\lambda^{1/3} - 4\lambda\right)} && \left(\because a_2 \geq \lambda^{1/3}\right) \\ &\leq 8\lambda^{1/3} && \left(\because \frac{1}{\left(\lambda^{1/3} - 4\lambda\right)} \leq \frac{2}{\lambda^{1/3}}\right) \end{aligned}$$

Let $\tilde{t}_b = \tilde{a}_1\tilde{a}_3$, then $\left|\tilde{t}_b - t_b\right| \leq 9\lambda^{1/3}$, because $\lambda \leq \frac{1}{2^9}$. □

Implementing the Oracles Π_H and Π_T. We show how to implement these oracles using only black-box access tothe protocol.

Lemma 5. *Given black-box access to the next message function of a stateless protocol π, we can efficiently implement the oracle Π and provide statistically close approximations of Π_H and Π_T on queries v such that $\chi_v \in [\delta, 1-\delta]$.*

Proof. Implementing Π simply involves picking a random string r uniformly at random and returning the bit $f_\pi(v; r)$, where f_π is the next message function of the protocol π.

Suppose we want to implement Π_H for input v, such that $\chi_v \geq \delta$. We generate $(D + 1/\epsilon)/\delta$ transcripts which are extensions of v. (This is performed by repeatedly calling Π on u_i starting with $u_{|v|} = v$ and $u_{i+1} := \Pi(u_i)$, till getting a complete transcript $\tau = u_D$.)

If there are no transcripts with outcome 1 (Heads) then we return 0 as the bit after v. Otherwise, return the bit after v in the first transcript which has outcome 1 (Heads). Conditioned on there being such a transcript, the bit produced is correctly distributed. On the other hand, the probability that none of the transcripts has outcome 1 is at most $(1-\delta)^{(D+1/\epsilon)/\delta} \leq \exp(-D - 1/\epsilon)$. So the probability of generating the output b is exponentially close to h_b. The oracle Π_T is also implemented similarly.

So, by making at most $D(D + 1/\epsilon)/\delta$ calls to $f(\cdot;\cdot)$ we can implement $\tilde{\Pi}_H$ and $\tilde{\Pi}_T$ that are statistically close to Π_H and Π_T respectively, for all v such that $\chi_v \in [1-\delta, \delta]$. □

D.2 Putting everything together

Now we can combine Lemma 3, Lemma 4 and Lemma 5 to obtain Theorem 1. When we run our attack in the oracle world, we have $\nu_D = \left(1 + \frac{9\lambda^{1/3}}{\delta}\right)^D - 1 \leq 18D\lambda^{1/3}/\delta \leq \epsilon/4$, since $9\lambda^{1/3}/\delta \leq 1$ and $\lambda \leq \left(\frac{\epsilon\delta}{72D}\right)^3$. So, in the oracle world, for the root node of the protocol we have:

$$\begin{aligned} s_1(A^{(1)}(v), \chi^*) + s_0(B^{(0)}(v), \chi^*) &\leq 1 + \frac{\delta}{(1-\chi^*)} + \frac{\delta}{\chi^*} + \frac{\epsilon}{4} \leq 1 + \frac{3\epsilon}{4} \\ s_0(A^{(0)}(v), \chi^*) + s_1(B^{(1)}(v), \chi^*) &\leq 1 + \frac{\delta}{(1-\chi^*)} + \frac{\delta}{\chi^*} + \frac{\epsilon}{4} \leq 1 + \frac{3\epsilon}{4} \end{aligned}$$

In the oracle world, the oracles are accessed at most poly$(D + \epsilon^{-1})$ times. Since the approximate oracles $\tilde{\Pi}_H$ and $\tilde{\Pi}_T$ are statistically close to the respective original oracles, the attack behavior in the real world and the oracle world differ by at most $\epsilon/4$. So, in the real world, for the root node of the protocol π we have:

$$s_1(A^{(1)}(v), \chi^*) + s_0(B^{(0)}(v), \chi^*) \leq 1 + \epsilon$$
$$s_0(A^{(0)}(v), \chi^*) + s_1(B^{(1)}(v), \chi^*) \leq 1 + \epsilon$$

Let χ^+ and χ^- be such that $s_1(\chi^+, \chi^*) = s_0(\chi^-, \chi^*) = \frac{1}{2} + \frac{\epsilon}{2}$. Now, Theorem 1 immediately follows.

D.3 Deferred Calculations

Lemma 6. *If $\chi_0, \chi_1, p_0 \in (0, 1)$ then $1 - r^*_{(1-c)} < r^*_c < p_c$.*

Proof. Recall that $r^*_{(1-c)} = p_{(1-c)}\left[\frac{\chi_{high} - \chi\chi_{high}}{\chi - \chi\chi_{high}}\right]$ and $r^*_c = p_c\left[\frac{\chi_{low} - \chi\chi_{low}}{\chi - \chi\chi_{low}}\right] = p_c\frac{\left(\frac{1}{\chi} - 1\right)}{\left(\frac{1}{\chi_{low}} - 1\right)} < p_c$, where $\chi = p_c\chi_{low} + p_{(1-c)}\chi_{high}$ and $p_c + p_{(1-c)} = 1$.

$$\begin{aligned}
1 - r^*_{(1-c)} &= 1 - \frac{p_{(1-c)}(1-\chi)\chi_{high}}{(1-\chi_{high})\chi} \\
&= \frac{(\chi - p_{(1-c)}\chi_{high}) - (\chi\chi_{high} - p_{(1-c)}\chi\chi_{high})}{(1-\chi_{high})\chi} \\
&= p_c\left[\frac{\chi_{low} - \chi\chi_{high}}{\chi - \chi\chi_{high}}\right]
\end{aligned}$$

Consider the function $f(x) = p_c\left[1 - \frac{\chi - \chi_{low}}{\chi - \chi^x}\right]$. It is easy to see that it is a monotonically decreasing function. Observe that $f(\chi_{low}) = r^*_c$ and $f(\chi_{high}) = 1 - r^*_{(1-c)}$. So, $1 - r^*_{(1-c)} < r^*_c < p_c$. □

Lemma 7. *Let $\chi = p_0\chi_0 + p_1\chi_1 \leq 1 - \delta$ and $p_0, p_1 \in [0, 1]$ such that $p_0 + p_1 = 1$. Suppose $\tilde{r}_0 = r_0 + e$ and $\tilde{r}_1 = 1 - \tilde{r}_0$, where $r_c \leq p_c$ and $e \in [-B, B]$, then:*

$$\left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right) \leq 1 + \frac{B}{\delta}$$

Proof. Consider the following manipulation:

$$\begin{aligned}
\left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right) &= \frac{1 - (r_c\chi_c + (1-r_c)\chi_{(1-c)}) + e(\chi_{(1-c)} - \chi_c)}{(1-\chi)} \\
&\leq \frac{1 - \chi + e}{(1-\chi)} \leq 1 + \frac{B}{\delta}
\end{aligned}$$

□

Lemma 8 (Case 0)**.** *If $\chi \geq 1 - (\delta + 2\lambda)$ or $\chi \leq (\delta + 2\lambda)$ and we honestly follow the protocol then $E \leq 2 \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. The result is immediate from the observation that $E \leq 2$ and from the following inequality:

$$1 \leq \frac{\delta}{1 - (1 - (\delta + 2\lambda))} + \nu_1$$

□

Lemma 9 (Case 1)**.** *If $\chi - \chi_c \leq \lambda^{1/3} + 4\lambda$, $\chi \leq 1 - \delta$, and we substitute $\tilde{r}_0 = \frac{p_0\chi_0}{\chi}$ and $\tilde{r}_1 = \frac{p_1\chi_1}{\chi}$, then $E \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. Since $\tilde{r}_c = \frac{p_c\chi_c}{\chi} > r_c^*$ and $\tilde{r}_{(1-c)} = \frac{p_{(1-c)}\chi_{(1-c)}}{\chi} < r_{(1-c)}^*$, we have $\tilde{T}_c < 0$ and $\tilde{T}_{(1-c)} > 0$. We know that $\frac{1}{\chi'} \leq \frac{1}{\chi}$ if and only if $\tilde{r}_c \leq h_c$. Because $e = 0$, we can use the bound in Lemma 7 in our lower bound to obtain:

$$\begin{aligned} E &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \frac{p_c\chi_c}{\chi}\left(\frac{\chi - \chi_c}{(1-\chi)}\right) + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{\lambda^{1/3} + 4\lambda}{\delta} + \nu_h \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

□

Lemma 10 (Case 2)**.** *If $\chi \leq 1-\delta$, and we substitute $\tilde{r}_0 = t_0 + e$ and $\tilde{r}_1 = t_1 - e$, where $e \in [-9\lambda^{1/3}, 9\lambda^{1/3}]$ and $\tilde{r}_c \leq h_c$, then $E \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. Recall that $\frac{1}{\chi'} \leq \frac{1}{\chi}$ if and only if $\tilde{r}_c \leq h_c$. Since $t_c < p_c$, we can use the bound in Lemma 7. When we substitute $\tilde{r}_0$ and $\tilde{r}_1$ and we get $\tilde{T}_{low} \geq 0$ and $\tilde{T}_{high} \geq 0$ then:

$$\begin{aligned} E^{(+,+)} &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

If we substitute $\tilde{r}_0$ and $\tilde{r}_1$ and we get $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 < 0$ then we know that $\tilde{r}_1 = r_1^* + e'$ such that $e' \in [0, 9\lambda^{1/3}]$ (Lemma 11). In this case:

$$\begin{aligned} E^{(+,-)} &\leq 1 + \frac{\delta}{(1-\delta)} + \frac{\delta}{\chi'} + \frac{e'(1-\chi_1)}{(1-\chi)} + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{e'(1-\chi_1)}{(1-\chi)} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{9\lambda^{1/3}}{\delta} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

Similarly, if $\tilde{T}_0 < 0$ and $\tilde{T}_1 \geq 0$ then:

$$\begin{aligned} E^{(-,+)} &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \frac{e'(1-\chi_0)}{(1-\chi)} + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{e'(1-\chi_0)}{(1-\chi)} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{9\lambda^{1/3}}{\delta} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

□

Lemma 11. *$t_c \le r_c^* \le h_c$ and $t_{(1-c)} \le r_{(1-c)}^*$ and $t_c + t_{(1-c)} = 1$.*

Proof. It is trivial to see that $r_c^* = \frac{p_c\chi_c(1-\chi)}{\chi(1-\chi_c)} \le h_c = \frac{p_c\chi_c}{\chi}$. The remainder of the proof is immediate from simple manipulation of terms:

$$\begin{aligned} & t_c = \frac{p_c\chi_c(1-\chi_{(1-c)})}{(\chi-\chi_0\chi_1)} \le \frac{p_c\chi_c(1-\chi)}{\chi(1-\chi_c)} = r_c^* \\ \iff \quad & \chi - (\chi_0+\chi_1)\chi + \chi\chi_0\chi_1 \le \chi - \chi_0\chi_1 - \chi^2 + \chi\chi_0\chi_1 \\ \iff \quad & 0 \le (\chi_0-\chi)(\chi-\chi_1) \\ \iff \quad & t_{(1-c)} = \frac{p_{(1-c)}\chi_{(1-c)}(1-\chi_c)}{(\chi-\chi_0\chi_1)} \le \frac{p_{(1-c)}\chi_{(1-c)}(1-\chi)}{\chi(1-\chi_{(1-c)})} = r_{(1-c)}^* \end{aligned}$$

And for the second part,

$$t_c + t_{(1-c)} = \frac{p_0\chi_0 - p_0\chi_0\chi_1 + p_1\chi_1 - p_1\chi_0\chi_1}{(\chi-\chi_0\chi_1)} = 1 \qquad \square$$

Lemma 12. *If $\left|\tilde{t}_c - t_c\right| \le 9\lambda^{1/3}$, $\left|\tilde{h}_c - h_c\right| \le 3\lambda^{1/3}$ and $\tilde{r}_c = \min\{\tilde{t}_c, \max\{0, \tilde{h}_c - 3\lambda^{1/3}\}\}$, then $|\tilde{r}_c - t_c| \le 9\lambda^{1/3}$ and $\tilde{r}_c \le h_c$.*

Proof. Let $\tilde{a} = \max\{0, \tilde{h}_c - 3\lambda^{1/3}\}$. It is trivial to observe that $\tilde{a} \le h_c$. We will show that $|\tilde{a} - h_c| \le 6\lambda^{1/3}$, i.e. $\tilde{a}$ is a good approximation of h_c. If $\tilde{h}_c \ge 3\lambda^{1/3}$ then the result is trivial. Otherwise, $h_c \le 6\lambda^{1/3}$ and, hence, $|\tilde{a} - p_c| \le 6\lambda^{1/3}$.

If $\tilde{t}_c \le \tilde{a}$ then $|\tilde{r}_c - t_c| \le 9\lambda^{1/3}$. Otherwise, i.e. $\tilde{t}_c > \tilde{a}$, we need to consider two cases. If $t_c \le \tilde{a}$ then $|\tilde{a} - t_c| = \tilde{a} - t_c \le \tilde{t}_c - t_c = \left|\tilde{t}_c - t_c\right| \le 9\lambda^{1/3}$. If $h_c \ge t_c \ge \tilde{a}$ (Lemma 11) then $|\tilde{a} - t_c| \le |\tilde{a} - h_c| \le 6\lambda^{1/3}$. Hence, $|\tilde{r}_c - t_c| \le \max\{6\lambda^{1/3}, 9\lambda^{1/3}\} = 9\lambda^{1/3}$.

Moreover, $\tilde{r}_c \le \tilde{a} \le h_c$. $\square$

E Constant Round Weak Coin-Flipping

In this section we will consider protocols whose transcripts are polynomially long but there are only constant number of rounds (i.e., alternations between Alice and Bob while generating the transcript). In general, the transcript tree can be thought of as a depth D (constant) tree with 2^k fan-out at each node, where k is the security parameter.

Recall that $A^{(b)}(v)$ (resp. $B^{(b)}(v)$) represents the expectation of the outcome when Alice (resp. Bob) wants to bias the outcome towards b in the subtree S_v. We will show the following result:

Theorem 5. *If infinitely-often one-way functions do not exist then for any constant round coin-tossing protocol π, there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$, such that for any $\epsilon = 1/poly(k)$ ($0 < \epsilon < 1$), the attacks run in polynomial time in k, and for sufficiently large k:*

1. $\min\left\{1 - A^{(1)}, B^{(0)}\right\} \le \epsilon$, *and*
2. $\min\left\{A^{(0)}, 1 - B^{(1)}\right\} \le \epsilon$,

where $A^{(b)}$ (resp. $B^{(b)}$) for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

In other words, if infinitely-often one-way functions do not exist then with probability close to 1 either Alice can bias the outcome to b or Bob can bias it to $(1-b)$ starting from any transcript prefix v.

E.1 Oracle World

For simplicity, we will prove Theorem 5 in an oracle world where we have access to *inverters*. An inverter I, when presented with a partial transcript v, honestly extends v by one round. The challenge is to implement these inverters so that they work well *on the distributions effected by the attack* (which in turn depends on inverters). In Section E.2, we will show how to efficiently approximate these inverters, if infinitely-often one-way functions do not exist, such that the actual algorithm's execution differ from the execution of the attack in the oracle world by at most $1/\text{poly}(k)$.

Hypothetical Attack. First we present the ideal attack we want to perform when we are provided access to inverters. We will describe the attack for Alice to bias towards 1 and other attacks can be analogously defined. When Alice wants to bias the outcome towards 1, she attacks at all nodes in the transcript tree which are Alice nodes. Suppose v is a partial transcript generated during the execution of the protocol. Our attack is recursively defined. Let h be the height of the node v in the transcript tree and $A^{(1)}(v)$ be the expected outcome when Alice is trying to bias towards 1 by performing her attack v onwards. For a leaf, $A^{(1)}(v)$ is defined to be the color of v and for a Bob node v, $A^{(1)}(v)$ is the expectation of $A^{(1)}(u)$, where u is a honest extension of the partial transcript v. When v is an Alice node, we shall use the following strategy for Alice: Alice will sample N_h extensions of the partial transcript v, i.e. $\{u_1, \ldots, u_{N_h}\}$, and finds $i \in N_h$ such that $A^{(1)}(u_i) = \max_{i\in[N_h]} u_i$. She sends the next message so that the computation moves to the node u_i in the transcript tree. Thus, $A^{(1)}(v)$ is defined as the expectation of $\max_{i\in[N_h]} u_i$ where each u_i is an honest extension of v. The quantity N_h will be defined suitably later in this section. We remark that, for every node v in the transcript tree, $A^{(1)}(v)$ or $B^{(0)}(v)$ are close to 1 or 0 respectively. This statement will be formalized as a lemma later in this section and we will also see how we can choose our parameters so that Alice or Bob can force 1 or 0 with near certainty.

Actual Attack. Despite having access to inverters, it is extremely hard to exactly compute the expected $A^{(1)}(u)$ when u is an honest extension of v. The problem is considerably harder when we try to compute the expectation $\max_{i\in[N_h]} u_i$. Instead, we will try to estimate the performance of the hypothetical attack using repetitive sampling. Note that we might incur an additive error in our estimation and, with extremely low probability, our estimation could be completely wrong. Thus, we will try to compute $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ which are good estimations of $A^{(1)}(v)$ and $B^{(0)}(v)$ respectively with high probability; and we will recursively use them in our attack instead of the exact $A^{(1)}(v)$ and $B^{(0)}(v)$ values.

Formally, the functions $\tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)$ are such that, with probability $(1-\epsilon_h)$, the following conditions are satisfied:

1. $\left|A^{(1)}(v) - \tilde{A}^{(1)}(v)\right| \leq \epsilon_h$,
2. $\left|B^{(0)}(v) - \tilde{B}^{(0)}(v)\right| \leq \epsilon_h$, and
3. $\min\left\{1 - \tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)\right\} \leq \epsilon_h$,

where h is the height of v in the transcript tree, $\epsilon_{h+1} = \Gamma^{1/2}\epsilon_h^{1/6}$ and $\epsilon_1 = \Gamma^{3/5}\epsilon^{6^D}$ and Γ is a parameter which will be defined later. The parameter Γ will be a large number which will be of the form $(1/\epsilon)^{\Theta(1)}$. Use ϵ_0 instead of ϵ_1 in the recursion.

Given, a particular $\epsilon = 1/\text{poly}(k)$, we will show how to perform our attack. We will prove the following result:

Lemma 13. *In the oracle world where we have access to the ideal inverters, we can efficiently implement $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ for all partial transcripts v; and $\min\{1 - A^{(1)}(v), B^{(0)}(v)\} \leq \epsilon_h$, where h is the height of the node v in the transcript tree.*

We emphasize that the condition $\min\{1-A^{(1)}(v), B^{(0)}(v)\} \leq \epsilon_h$ is not probabilistic, unlike the properties of the quantities $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$. We also do not try to obtain tighter bounds because the qualitative result does not change; although as intermediate steps, we will prove and use tighter bounds on these quantities. To prove this lemma, we will proceed by induction on the height h of the node v. Recall that leaves, which correspond to complete transcripts, have height $h = 0$ and the root r of the transcript tree has height $h = D$. For the base case, consider any node v at height 0, i.e. either Alice or Bob announces that the outcome is 0 or 1. In this case, it is trivial to implement $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$.

For the inductive step, we will show that given an implementation of these functions for nodes with height h we can implement these functions for any node at height $(h+1)$. W.l.o.g., let v be an Alice node at height $(h+1)$.

Computation of $\tilde{A}^{(1)}(v)$. Prepare a set $\{u_1, \ldots, u_{N_{h+1}}\}$, where $N_{h+1} = \epsilon_h^{-1/2}$, of honest extensions of the partial transcript v. Find the maximum $\tilde{A}^{(1)}(u_i)$, for $i \in [N_{h+1}]$. Recall, that $A^{(1)}(v)$ is the expected outcome of this experiment. Perform this task $M_{h+1} = \Gamma^{1/3}\epsilon_h^{-1/3}$ times and define $\tilde{A}^{(1)}(v)$ as the average of the respective maximums. Intuitively, we are repeating the experiment M_{h+1} times to obtain a good estimation of $A^{(1)}(v)$. We will prove that this definition of $\tilde{A}^{(1)}(v)$ suffices by considering several intermediate hybrid worlds.

1. As an intermediate world, suppose we have access to the ideal values of $A^{(1)}(u)$, where u is a honest extension of the transcript v. $A^{(1)}(v)$ is defined as the expected outcome when we honestly sample N_{h+1} children of v and compute their average. Repeating this experiment M_{h+1} times helps us estimate $A^{(1)}(v)$ with ζ such that, with probability $1 - \exp(-\Theta(\Gamma))$, the quantity $\left|\zeta - A^{(1)}(v)\right|$ is at most $\sqrt{\frac{\Gamma}{M_{h+1}}} = k^{1/3}\epsilon_h^{1/6}$. We will choose $\Gamma = \Theta\left(\left(\frac{1}{\epsilon^{6^D}}\right)^{5/8}\right)$ such that,

$$\begin{aligned}\exp(-\Theta(\Gamma)) &\leq \Theta\left(\frac{1}{\Gamma}\right)\\ &\leq \Gamma^{3/5}\epsilon^{6^D} = \epsilon_1\\ &\leq \epsilon_h\end{aligned}$$

   This implies that the expression $1 - \exp(-\Theta(\Gamma))$ in the Chernoff bound is at least $1 - \epsilon_h$, i.e. with probability $(1 - \epsilon_h)$ our estimation is within $k^{1/3}\epsilon_h^{1/6}$ additive error of the actual value of $A^{(1)}(v)$.

2. Now, we replace every ideal value of $A^{(1)}(u)$ with $(A^{(1)}(u))'$ such that $(A^{(1)}(u))' = A^{(1)}(u)$ with probability $1-\epsilon_h$. In this hybrid, we get a new estimate ζ' such that, with probability $1-M_{h+1}N_{h+1}\epsilon_h-\epsilon_h = 1-k^{1/3}\epsilon_h^{1/6}-\epsilon_h$, the error in our estimation $\left|\zeta' - A^{(1)}(v)\right|$ is at most $k^{1/3}\epsilon_h^{1/6}$. This step follows from union bound.

3. Finally, replacing $(A^{(1)}(u))'$ values with the $\tilde{A}^{(1)}(u)$ values, the new estimate $\tilde{A}^{(1)}(v)$ can deviate at most ϵ_h away from the estimated ζ'. This step follows from the fact that $\tilde{A}^{(1)}(v)$ is a convex linear combination of $\tilde{A}^{(1)}(u)$ values. Thus, we can conclude that $\left|A^{(1)}(v) - \tilde{A}^{(1)}(v)\right| \leq k^{1/3}\epsilon_h^{1/6} + \epsilon_h \leq \epsilon_{h+1}$, with probability at least $1 - k^{1/3}\epsilon_h^{1/6} - \epsilon_h \geq 1 - \epsilon_{h+1}$.

Computation of $\tilde{B}^{(0)}(v)$. Compute $\{u_1, \ldots, u_{M_{h+1}}\}$ honest extensions of the partial transcript v. Define $\tilde{B}^{(0)}(v)$ as the average of $\tilde{B}^{(0)}(u_i)$, where $i \in [M_{h+1}]$. Similar to the argument presented above, with probability at least $(1 - M_{h+1}\epsilon_h - \epsilon_h) \geq (1 - \epsilon_{h+1})$, the quantity $\left|B^{(0)}(v) - \tilde{B}^{(0)}(v)\right|$ is at most $M_{h+1}\epsilon_h + \epsilon_h \leq \Gamma^{1/3}\epsilon_h^{1/6} + \epsilon_h \leq \epsilon_{h+1}$.

Attacks are good. For the final step in our inductive proof, we need to show that the quantities $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ satisfy the third property of our theorem, i.e. at least one them is a very good attack; and we also need to show that $A^{(1)}(v)$ or $B^{(0)}(v)$ is (respectively) close to 1 or 0 as well. Let $(1-p)$ be the probability of $A^{(1)}(u) \geq 1-\epsilon_h$, where u is some honest extension of v by one round. Let $q \geq p$ (since, inductively, $A^{(1)}(u)$ or $B^{(0)}(v)$ is within ϵ_h of 1 or 0 respectively) be the probability of $B^{(0)}(u) \leq \epsilon_h$, where u is some honest extension of v by one round. Observe that the maximum of N_{h+1} samples of $A^{(1)}(u)$ is at least $1-\epsilon_h$, unless each one of the sampled $A^{(1)}(u)$s were less than $1-\epsilon_h$. Thus, the expected outcome $A^{(1)}(v)$ is at least $\left(1-p^{N_{h+1}}\right)(1-\epsilon_h)$. Similarly, one can argue that $B^{(0)}(v) \leq (1-p)+p\epsilon_h \leq (1-p)+\epsilon_h$. There are two cases to consider:

1. If $p \geq 1-\epsilon_h^{1/4}$, then $B^{(0)}(v) \leq \epsilon_h^{1/4}+\epsilon_h$.
2. If $p \leq 1-\epsilon_h^{1/4}$, then

$$\begin{aligned}
A^{(1)}(v) &\geq \left(1-p^{N_{h+1}}\right)(1-\epsilon_h) \\
&\geq \left(1-\left(1-\epsilon_h^{1/4}\right)^{N_{h+1}}\right)(1-\epsilon_h) && \text{, since } p \leq 1-\epsilon_h^{1/4} \\
&\geq \left(1-\exp\left(-N_{h+1}\epsilon_h^{1/4}\right)\right)(1-\epsilon_h) && \text{, since } (1-x) \leq \exp(-x) \\
&= \left(1-\exp\left(-\epsilon_h^{-1/4}\right)\right)(1-\epsilon_h) && \text{, since } N_{h+1} = \epsilon_h^{-1/2} \\
&\geq (1-\epsilon_h^{1/4})(1-\epsilon_h) && \text{, since } x \leq 1 \implies \exp(-1/x) \leq x \\
&\geq 1-\epsilon^{1/4}-\epsilon_h && \text{, by expansion}
\end{aligned}$$

So, we obtain that $\min\{1-A^{(1)}(v), B^{(0)}(v)\} \leq \epsilon^{1/4}+\epsilon_h \leq \epsilon_{h+1}$. Finally, with probability $(1-\epsilon_{h+1})$, we have $\min\{1-\tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)\} \leq (\Gamma^{1/3}\epsilon_h^{1/6}+\epsilon_h)+(\epsilon_h^{1/4}+\epsilon_h) \leq \epsilon_{h+1}$. This completes the induction step and the proof of the lemma.

Figure 6 describes the algorithm to implement $\tilde{A}^{(1)}(v)$. One important observation is that, since the number of rounds D is constant, we have $\epsilon_i \geq \epsilon_1 = \Gamma^{3/5}\epsilon^{\Theta(1)}$. Since, at each round the time complexity of our attack is poly$(1/\epsilon_i)$, our attack runs in polynomial time. Moreover, it is also easy to see that $1-A^{(1)}(r)$ or $B^{(0)}(r)$ is at most ϵ, where r is the root of the transcript tree.

E.2 Implementing Efficient Inverters

Need to rewrite this section. Given access to efficient inverters, Section E.1 shows how Alice and Bob can efficiently bias the outcome ϵ-close to b or $(1-b)$, respectively. In this section, we will show that if infinitely-often one-way functions do not exist then for sufficiently large Γ, we can efficiently implement close approximations of these inverters.

A closer look at our attack reveals that the inverters are used in the following manner. Suppose Alice wants to bias the outcome to 1. Let the current partial transcript be v (suppose $(D-i)$ message exchanges have already taken place) and Alice is supposed to send the next message of the transcript. To generate her message, she needs to implement $\tilde{A}^{(1)}(\cdot)$ functions for some nodes in S_v. In fact, she samples a subtree of S_v such that, for some node u in it:

1. If u is an Alice node, then its degree is N_jM_j, and
2. If u is a Bob node, then its degree is M_j,

where j is the height of u in the transcript tree. So, in other words, to generate the $(D-i+1)$-th message, Alice accesses polynomially many inverters on nodes u which are at height i, $(i-1)$, ..., 1 in that order. Let

Subroutine to compute $\tilde{A}^{(1)}(v)$

1. If v is a leaf, return $\tilde{A}^{(1)}(v) = \chi_v$.
2. If v is an Alice node at height j: Sample $\{u_1, \ldots, u_{N_j M_j}\}$ honest extensions of v by calling $I(v)$. Return
$$\tilde{A}^{(1)}(v) = \frac{\sum_{k=1}^{M_j} \left[\max_{k'=1}^{N_j} \tilde{A}^{(1)}\left(u_{(k-1)N_j+k'}\right)\right]}{M_j}$$
3. If v is a Bob node at height j: Sample $\{u_1, \ldots, u_{M_j}\}$ honest extensions of v by calling $I(v)$. Return
$$\tilde{A}^{(1)}(v) = \frac{\sum_{k=1}^{M_j} \tilde{A}^{(1)}(u_k)}{M_j}$$

Algorithm $\mathsf{Adv}_A^{(1)}$

Let v be an Alice node at height j.

1. Sample $\{u_1, \ldots, v_{N_j}\}$ honest extensions of v using $I(\cdot)$.
2. Output the message m such that $vm = \text{argmax}_{k \in [N_j]} \tilde{A}^{(1)}(u_k)$.

Figure 6: *Computation of $\tilde{A}^{(1)}(v)$ to help Alice bias towards outcome 1.*

$\mathcal{Q}_{i,j}$ be the set of nodes at height j queried by Alice. Observe that all queries in $\mathcal{Q}_{i,j}$ are independent of each other; and without loss of generality, we can assume that all queries in $\mathcal{Q}_{i,j}$ are performed simultaneously. The number of queries performed is upper bounded in the following manner: $|\mathcal{Q}_{i,j}| \leq \prod_{k=j}^{i} M_k N_k$.

Let $\tilde{I}_{i,j}$ be the inverter used by Alice to query the nodes at height j when she is generating the $(D-i+1)$-th message. We will provide an inductive construction of $\tilde{I}_{i,j}$, where $i \in [D]$ and $j \in [i]$. Define $\mathcal{I}_{i,j}$ as the collection of all inverters of the form $\left\{\tilde{I}_{i',j'} | (D \geq i' > i \text{ or } D \geq j' > j) \text{ and } j' \in [i'] \text{ and } i' \geq i\right\}$. Let $\mathcal{A}(r_A)$ be the attack algorithm for Alice when she tries to bias the outcome to 1, where r_A is its random tape and she uses the oracles provided in $\mathcal{I}_{0,0}$. We can assume, without loss of generality, that the random tape used by $\mathcal{A}(r_A)$ to run an instance of the inverter $\tilde{I}_{i,j}$ is independently chosen. Let $\mathcal{A}_{i,j}^{(\text{pre})}(r_A)$ be the execution of $\mathcal{A}(r_A)$ till it makes the $\mathcal{Q}_{i,j}$ queries and outputs the set $\mathcal{Q}_{i,j}$, using the oracles provided in $\mathcal{I}_{i,j}$. Consider the following machine $\mathcal{C}(k, \mathcal{A}_{i,j}^{(\text{pre})})$: It samples r_A and r_B uniformly at random and simulates a run of $\mathcal{A}_{i,j}^{(\text{pre})}(r_A)$ against a honest Bob with local randomness r_B. Let $\mathcal{C}^*(k, \mathcal{A}_{i,j}^{(\text{pre})})$ be the machine which runs $\mathcal{C}(k, \mathcal{A}_{i,j}^{(\text{pre})})$ and concatenates $r_A \circ r_B$ at the end.

If infinitely-often one-way functions do not exist, then for any constant c and sufficiently large k, there exists an efficient machine $\tilde{I}_{i,j}$ such that [OW93]:

$$\left\| \mathcal{C}^*(k, \mathcal{A}_{i,j}^{(\text{pre})}) - \mathcal{C}(k, \mathcal{A}_{i,j}^{(\text{pre})}) \circ \tilde{I}_{i,j}(\mathcal{C}, \mathcal{A}_{i,j}^{(\text{pre})}, 0^{k^c D^2 |Q_{i,j}|}) \right\|_1 \leq \frac{1}{k^c D^2 |Q_{i,j}|}$$

Note that the time complexity of $\tilde{I}_{i,j}$ is at most $k^{\Theta(1)^{D^2}}$, which is a polynomial because D is a constant. Since there are finitely many inverters $\tilde{I}_{i,j}$ and four different attacks, for sufficiently large k all inverters used in each of our attacks perform well. By union bound, the behavior of our attacks when provided with $\mathcal{I}_{0,0}$ instead of the actual inverters $I(\cdot)$ can differ by at most $1/k^c$. This completes the proof of Theorem 5.