Electronic Colloquium on Computational Complexity, Report No. 39 (2000)

ECCC

Impossibility of Black-Box Reduction from Non-Adaptively to Adaptively Secure Coin-Flipping

Yevgeniy Dodis
MIT*

April 24, 2000

Abstract

Collective Coin-Flipping is a classical problem where n computationally unbounded processors are trying to generate a random bit in a setting where only a single broadcast channel is available for communication. The protocol is said to be $b(n)$-resilient if any adversary that can corrupt up to $b(n)$ players, still cannot bias the coin to some desired outcome almost certainly. The problem is extensively studied for the case of *non-adaptive* adversaries who have to decide which players to corrupt before the protocol starts. In particular, it is well-known that the optimum resilience threshold is $n/2$ in this case. However, none of these protocols is resilient against an *adaptive* adversary who can corrupt just a *single* player in the course of the execution. In fact, Ben-Or and Linial [BL90] conjectured that the adaptive adversary is much more powerful than the non-adaptive adversary. More specifically, that the optimal resilience threshold for adaptive adversaries is only $O(\sqrt{n})$ (which is achieved by a simple "majority" protocol).

We give strong evidence towards this conjecture by showing that no *black-box* transformation from any statically secure coin-flipping protocol can yield an adaptively secure protocol tolerating $\omega(\sqrt{n})$ players, so it is impossible to beat the simple majority protocol in this way. The result is proven by reducing the question in hand to the analysis of a novel *imperfect random source* of independent interest. This imperfect random source generalizes and unifies two well-known imperfect random sources: the SV-source of Sántha-Vazirani [SV86] and the bit-fixing source of Lichtenstein-Linial-Saks [LLS89]. While from each of these sources it is easy to extract a "somewhat random" bit, we show this this is no longer possible for the generalized source.

*Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139. Email: yevgen@theory.lcs.mit.edu.

ISSN 1433-8092

1 Collective Coin-Flipping

The Setting. *Collective Coin-Flipping* in the full-information model is a classical problem introduced by Ben-Or and Linial [BL90], where n *computationally unbounded* processors are trying to generate a random bit in a setting where only a *single broadcast channel* is available for communication. As usual, we assume that some subset of the parties can be *faulty* or malicious, and we would like our protocol to be *"resilient"* against the faulty players (which we define precisely later). Taking the worst case scenario, we assume that all the faulty parties are coordinated by a central *adversary* $\mathcal{A}$, who can corrupt up to b out of n players. We call such an adversary b-*bounded*. The computation proceeds in rounds, in which each processor broadcasts a message to the other processors. The crucial complication is that the network is assumed to be *asynchronous within a round* and is synchronized only in between the rounds. For example, players cannot flip a coin by broadcasting a random bit and taking their exclusive OR: the last player to talk can completely control the output. Again taking the worst case scenario, we assume that in each round first $\mathcal{A}$ receives all the messages broadcast by the honest players, and only then decides which messages to send on behalf of the bad players. Finally, we assume that $\mathcal{A}$ never violates the protocol in a manner that can be detected (for example, if a faulty processor has to send a random bit, he does so; however, the bit need no be random). The output of the protocol is some pre-agreed deterministic function of the messages exchanged over the broadcast channel.

The Goal. As we said, the objective of collective coin-flipping is for the players to agree on a random bit. Given a bit σ generated by some random experiment, we define its *fairness* $\gamma \leq \frac{1}{2}$ to be the minimum of the probability that $\sigma = 0$ and that $\sigma = 1$, and call such a bit γ-fair. Thus, constant bit is 0-fair while a random bit is $\frac{1}{2}$-fair. When talking about coin-flipping protocols, we usually talk about a family a protocols parametrized by the number of players, n. Having this in mind, a coin-flipping protocol Π is said to be *weakly* $b(n)$-*resilient* if for any $b(n)$-bounded adversary Π produces a coin which is γ_0-fair, where γ_0 is a fixed (possibly very small) constant *independent of* n. Such a coin is called *slightly random*. Π is said to be *strongly* $b(n)$-*resilient* if for any $b(n)$-bounded adversary Π produces a $(\frac{1}{2} - o(1))$-fair coin. Such coin is called *almost random*. Traditionally, the "standard" definition of resilience for coin-flipping is that of weak resilience, so this is the notion that we will use, unless we state otherwise.[1]

Type of Adversary. So far we have been very vague about the type of adversary that we have. The only thing we specified about it, is that it coordinates the faulty players and can make them deviate in any manner undetected by the honest players. However, we have not talked about how and when the player becomes faulty. Most of the papers in the full-information model assume and *crucially use* the fact that the adversary $\mathcal{A}$ is *static* (or non-adaptive), i.e. it decides on which b parties to corrupt *before the protocol starts*. The honest player do not know which b players were selected by $\mathcal{A}$, but the resulting coin has to be slightly random for any *fixed* set of b players. A somewhat more realistic and much more powerful type of an adversary is an *adaptive* adversary. This adversary can listen to all the communication and corrupt up to b players anywhere *in the course of the execution*. As we will see, this indeed seems to give an adaptive adversary a lot of power over the static adversary.

Coin-Flipping with Static Adversaries. The case of static adversaries has been extensively studied and is understood very well by now. Historically, coin-flipping protocols are divided into one-round/one-bit protocol and general (many-round/many-bit) protocols.

[1] In fact, since our main result is an impossibility result, it will become only stronger if we consider strong fairness.

In the one-round/one-bit protocols each player i is supposed to send a single bit x_i, and the resulting coin is some deterministic function $f(x_1, \ldots, x_n)$. Such protocols deserve such a special attention because of their simplicity (they are given by a boolean function $f : \{0,1\}^n \to \{0,1\}$) and the connection to "influence of variables" on boolean functions [KKL89]. If f defines a $b(n)$-resilient protocol, it itself is called $b(n)$-resilient. Ben-Or and Linial [BL90] defined an "iterated majority of 3" function that is resilient against $\Omega(n^{\log_3 2}) \approx \Omega(n^{0.63})$ players. Ajtai and Linial [AL93] non-constructively showed that there exist $\Omega(n/\log^2 n)$-resilient functions. Unfortunately, there is not much potential in improving this result, since Kahn, Kalai and Linial [KKL89] used a beautiful argument to show that there are no $\omega(n/\log n)$-resilient functions.

In contrast, general (statically secure) coin-flipping protocols can achieve much better resilience. Historically, *all* such protocols first elect a single representative player (called a *leader*), who then flips the final coin by itself. If the probability that the leader is non-faulty is lower bounded by a constant γ_0 (independent of n) for any $b(n)$-bounded adversary, then the fairness of the resulting coin is at least $\gamma_0/2$, yielding a weakly $b(n)$-resilient coin-flipping protocol. The intermediate *leader election* (where the players are trying to select a non-faulty leader[2]) is by itself very important and, as we said, has been typically considered instead of solving a seemingly easier coin-flipping problem.[3]

The first interesting leader election (and thus, coin-flipping) protocol was given by Saks [S89], who designed a very simple "baton passing" algorithm which he showed was $\Omega(n/\log n)$-resilient (Ajtai and Linial [AL93] improved the analysis of Saks to show that baton passing is in fact *strongly* $\Omega(n/\log n)$-resilient). Saks also observed that no leader election and coin-flipping protocol could be $n/2$-resilient (formal proof appears in [BN]). The question of achieving $\Omega(n)$-resilience was affirmatively resolved by Alon and Naor [AN93]. Using an elegant, but non-constructive "random tree" protocol, they showed the existence of an $n/4$-resilient leader election protocol.[4] Adding several "tricks", they moved the resilience threshold to $(\frac{1}{3} - \delta)n$ (for any $\delta > 0$). However, Boppana and Narayanan [BN] showed that these tricks were not necessary and the "random tree" protocol by itself is $(\frac{1}{2} - \delta)n$-resilient. This result showed that the *optimal resilience of static coin-flipping (and leader election) is* $n/2$.

From this point on, the research in statically secure coin-flipping and leader election was focusing on making *constructive* and/or more *efficient* leader election and coin-flipping protocols [ORV94, RZ98, F99]. This culminated in a recent paper of Feige [F99] who gave constructive, extremely simple and efficient $(\frac{1}{2} - \delta)n$-resilient coin-flipping and leader election protocol taking $\log^* n + O(1/\delta)$ rounds with each player sending $O(\log n)$ bits per round (improving and simplifying previous protocols of [RZ98] with similar parameters).

A lot is also known on the optimal dependence $\gamma(b)$ of the fairness of the coin and the number b of faulty players. Namely, $\gamma(b) = \frac{1}{2} - \Theta(\frac{b}{n})$. The upper bound $\gamma(b) \leq \frac{1}{2} - \Omega(\frac{b}{n})$ was elegantly shown by Ben-Or and Linial [BL90]. The lower bound $\gamma(b) \geq \frac{1}{2} - O(\frac{b}{n})$ was proved in a series of papers for larger and larger values of b: by Ben-Or and Linial [BL90] for $b = O(n^{0.63})$, by Ajtai and Linial [AL93] for $b = O(n/\log n)$ and, finally, by Alon and Naor [AN93] for all b. Notice that the upper bound implies that there are no strongly $\Omega(n)$-resilient coin-flipping protocols, while the lower bound implies that there is "no limit" for strongly $o(n)$-resilient protocols. (This is one of

[2] We notice right away that, unlike coin-flipping, leader election makes no sense against adaptive adversaries: the adversary can always corrupt the leader at the end of the protocol.

[3] Feige [F99] recently showed the "converse", i.e. that any $b(n)$-resilient coin-flipping protocol can be efficiently transformed into a $b(n)$-resilient leader election protocol. Thus, in the static setting leader election and coin-flipping are "equivalent".

[4] Alon and Naor [AN93] and later Cooper and Linial [CL95] also gave very complicated but constructive $O(n)$-resilient protocols with truly tiny constants in front of n.

the reasons why weak resilience is typically considered.)

To summarize, statically secure coin-flipping is very well understood by now, the optimal resilience threshold is $n/2$, and all the best protocols (which are quite simple and efficient) elect a single leader who flips the final coin.

Coin-Flipping with Adaptive Adversaries. First we remark that *all the best statically secure coin-flipping protocols are not even* 1*-resilient against adaptive adversaries.* Indeed, all of them first elect the leader, so corrupting the leader allows the adversary to completely fix the coin. More generally, the whole philosophy of most statically secure protocols is not applicable here, as these protocols try to aggressively eliminate players (without significantly changing the fraction of faulty players).

Adaptive adversaries were already considered in the original paper of Ben-Or and Linial [BL90]. In particular, they observed that the following simple "majority" protocol achieves $\Theta(\sqrt{n})$-resilience. Each player sends a random bit, and the final coin is the majority bit. Here any $c\sqrt{n}$ players (for small enough c) do not affect the protocol, since with probability $1 - o(1)$ the majority will be determined anyway. Adaptivity does not help here since in order to bias the coin to 1 (similarly for 0) it does not really matter whom and when to corrupt. Any set B of b players will do: the optimal adversarial strategy for these players is to declare that their random bits are all 1. Surprisingly enough, this simple protocol is the *best known* adaptively secure coin-flipping protocol! In fact, Ben-Or and Linial [BL90] conjectured that this protocol is indeed optimal.

Conjecture 1 ([BL90]) *Majority is the optimal coin-flipping protocol against adaptive adversaries. In particular, the maximum threshold that can be tolerated is* $O(\sqrt{n})$.

This conjecture, if true, would imply that adaptive adversaries are much more powerful than static adversaries for the problem of collective coin-flipping. The only result addressing this conjecture is a very nice paper by Lichtenstein, Linial and Saks [LLS89]. By looking at another question that we will discuss later (for a different reason), they derived along the way the following result, that *seems* to strongly support the conjecture above.

Theorem 1 ([LLS89]) *If each player is allowed to broadcast at most* 1 *bit (possibly, taking* n *rounds overall), the most resilient adaptively-secure coin-flipping protocol is indeed the majority protocol (which tolerates* $\Theta(\sqrt{n})$ *faults).*

The theorem above already shows some strong separation between static and adaptive adversaries. Recall that the result of Ajtai and Linial [AL93] says that there are $\Omega(n/\log^2 n)$-resilient functions. In other words, there are $\Omega(n/\log^2 n)$-resilient coin-flipping protocols where each player sends one bit (even in a single round) which are secure against static adversaries. The above result says that no function (e.g., the function of Ajtai and Linial) $f : \{0,1\}^n \to \{0,1\}$, even if we spread it in any way over n rounds, can be more than $O(\sqrt{n})$-resilient against adaptive adversaries!

However, Theorem 1 supports Conjecture 1 much less than it seems to. Indeed, restricting each player to send at most 1 bit seems like a huge limitation. We saw that it was very limiting *even for statically secure protocols* (recall, no function can be more than $O(n/\log n)$-resilient by the result of [KKL89], and there are general $n/2$-resilient statically secure protocols [BN, ORV94, RZ98, F99]). For adaptively secure protocols, sending at most one bit seems particularly restrictive since last players typically have much more "influence" in this case, and it seems quite conceivable that this unproportional influence can be mitigated by having players send many bits (e.g., in many round-robin cycles).

To summarize, adaptively secure coin-flipping is much less understood than its static counterpart, there seems to be some indication that adaptive adversaries are much more powerful than static adversaries, but there is little formal evidence supporting this claim.

2 Our Approach and Main Impossibility Result

Black-Box Reductions. We look at the problem of constructing adaptively secure coin-flipping protocols from a different perspective. Namely, assume we are given a protocol Π which is known be "good" against *static* adversaries (we will be more precise in a second). We ask the question if it is possible to transform Π in a "black-box" way so as to obtain a "somewhat good" *adaptively secure* protocol Φ. To capture the intuition that we are really obtaining Φ from Π, we do not allow the player to send any messages outside those they send in Π, but allow them to run Π sequentially as many times as they wish. Of course, one might try to let the players run some sub-protocols in between running Π, but then it is very hard to say that we are really using Π and do not, say, run a brand new protocol in the middle and ignore everything that happens in Π. Thus, Φ can run Π any number of times times D, and get some coins $x_1, \ldots, x_D$, some of which might not be very fair since we ran Π against an adaptive adversary. To correct against this, players in Φ try to apply some function $f : \{0,1\}^D \to \{0,1\}$ to $x_1, \ldots, x_D$ to produce the final coin. This leads us to the following natural definition.

Definition 1 *Let D be any integer and $f : \{0,1\}^D \to \{0,1\}$ be any function. We let $\Phi(D, f, \Pi)$ (often we omit Π) be the protocol where players sequentially run the protocol Π D times, obtain coins $x_1, \ldots, x_D$, and output $f(x_1, \ldots, x_D)$ as the resulting coin. The class $\{\Phi(D, f, \Pi) \mid D \geq 1, f : \{0,1\}^D \to \{0,1\}\}$ is called the class of* black-box transformations *of* Π.

The (False) Hope. The intuitive reason why black-box transformations look very promising is the following. Assume that Π is $b(n)$-resilient and we wish to construct an adaptively $b(n)$-resilient $\Phi(D, f, \Pi)$. Ignoring the question of efficiency, we can make D arbitrarily large compared to $b(n)$ and n (e.g., 2^{2^n} if we so wish). Assume now $\mathcal{A}$ can adaptively corrupt up to $b(n)$ players. Let us take the worst case, and assume that whenever $\mathcal{A}$ corrupts even a single player in the middle of Π_i (the i-th run of Π), he controls x_i. *But this can happen at most $b(n) \ll D$ times.* And if $\mathcal{A}$ does not corrupt a player in the middle of Π, we know from the static security of Π that the coin is at least slightly random. Thus, at most $b \ll D$ of the x_i's are really biased, the remaining $D - b$ of x_i's are at least slightly random (maybe even almost random). So it seems like there should not be a big problem to design a function f that would be able to "ignore" this "miniscule" number b of "fixed" bits, and extract just a single somewhat random bit from the remaining $(D - b)$ "good" bits. We will show, perhaps even surprisingly, that this hope is unfortunately false for any interesting setting of parameters. In particular, one cannot beat the simple majority protocol in this way.

Adaptive Adversary for a Black-Box Transformation. The definition of a black-box transformation views the protocol Π as "one piece" that is simply being run several times. Even though given a particular Π (and D and f), we will end up with a particular protocol $\Phi(D, f, \Pi)$ and can talk about it being adaptively $b(n)$-resilient, it is more natural to let the adaptive adversary $\mathcal{A}$ for Φ perform "meta-operations" on the entire run of each Π (consistent with the static security of Π). Namely, (1) $\mathcal{A}$ can decide not to corrupt any players during the run of Π, and then the fairness of the resulting coin is what is achieved by Π, or (2) $\mathcal{A}$ can decide to corrupt one or more player during the run of Π, and then we do not know anything about the resulting coin, and, therefore, have to assume the worst (i.e., $\mathcal{A}$ can fix the coin). We make this more formal.

Assume that given a fixed set B of faulty players, Π produces a $\gamma_\Pi(B)$-fair coin for any static adversary who corrupts B at the beginning, and let $\gamma_\Pi(b) = \min_{|B|=b} \gamma_\Pi(B)$ be the best that a b-bounded static adversary can achieve. Let us denote by Π_i the i-th run of Π, and by x_i the resulting coin. As before, $\mathcal{A}$ is called b-bounded if he corrupts at most b players overall. However, now we assume that $\mathcal{A}$ (the adversary for $\Phi(D, f, \Pi)$) has the following capabilities:

(A) If at the beginning of Π_i the set of corrupted players is B and $\mathcal{A}$ decides not to corrupt new players during Π, the resulting coin x_i is $\gamma_\Pi(B)$-fair, but $\mathcal{A}$ can set the probability of $x_i = 0$ anywhere in the interval $[\gamma_\Pi(B), 1 - \gamma_\Pi(B)]$.

(B) If $\mathcal{A}$ decides to corrupt at least one new player during the execution of Π_i, he can set the resulting coin x_i to any value.

We justify assumptions (A) and (B) in two ways. First of all, we are talking about *black-box reductions*. In other words, we do not know and do not want to assume anything more about Π than what is given to us by the function $\gamma_\Pi(B)$. Thus, if $\mathcal{A}$ does not corrupt new players inside Π_i, we know that $\Pr(x_i = 0) \in [\gamma_\Pi(B), 1 - \gamma_\Pi(B)]$, but we cannot assume anything more, so we assume that $\mathcal{A}$ can set $\Pr(x_i = 0)$ anywhere in this interval. Similarly, once $\mathcal{A}$ corrupts a player inside Π_i, nothing can be said about the behavior of the resulting coin, so we again have to assume the worst case.

The other justification comes from the fact that all best non-adaptively secure coin-flipping protocols (e.g., [AN93, ORV94, RZ98, F99]) essentially satisfy both of these assumptions.[5] Assumption (B) because they always elect the leader, so corrupting the leader allows the adversary to control the coin. And assumption (A) because these protocols are actually symmetric in 0 and 1 and by making faulty players be "less and less faulty", they can indeed achieve essentially any probability inside the specified interval.

Main Result. Our main result is the following theorem, which states that using black-box reductions one cannot significantly beat the simple majority protocol, giving further support to Conjecture 1.

Theorem 2 *For any family of coin-flipping protocols Π, there is no black-box transformation resulting in an adaptively $\omega(\sqrt{n})$-resilient family of protocols $\Phi(D, f, \Pi)$.*

We also remark that the adaptive adversaries we will use to prove this result satisfy considerably weaker assumptions than (A) and (B). For example, we will only use the extremes $\gamma_\Pi(B)$ and $(1 - \gamma_\Pi(B))$ (even for some particular B) for assumption (A).[6] As for assumption (B), we will only use the fact that if $\mathcal{A}$ wants to completely control the coin, he can do so by corrupting just some (rather than any) one player. Some further relaxations will be clear from the proof we present, but the point we are making is that our main result is somewhat surprising and certainly non-trivial even without any of these relaxations. Indeed, in our informal intuition above (of why black-box reductions look very promising), assumptions (A) and (B) did not seem to create any problems, so even with these assumptions it is quite interesting to see why the intuition was wrong.

[5] In fact, it is easy to check that our main Theorem 2 holds on a "concrete level" if we replace Π with any of these protocols.

[6] Essentially, we are just ruling out the possibility that the static adversary can influence the bit towards 0, but cannot do (almost) the same for 1.

3 Reduction to Imperfect Random Sources

We reduce the proof of Theorem 2 to the analysis of a novel *imperfect random source* (IRS). Assume $\Phi(D, f, \Pi)$ is adaptively $2b(n)$-resilient. We construct the following $2b(n)$-bounded adversary for Φ satisfying properties (A) and (B). Let $b = b(n)$, $\gamma = \gamma_\Pi(b)$ and let B be the set of players of cardinality b achieving $\gamma_\Pi(B) = \gamma_\Pi(b) = \gamma$. Before Π_1 starts, $\mathcal{A}$ corrupts all the players in B. Therefore, from now on in each of the D invocations of Π, $\mathcal{A}$ can set the 0-probability of x_i anywhere in at least the interval $[\gamma, 1-\gamma]$. As $\mathcal{A}$ will later corrupt more players, this interval can only expand, but our particular $\mathcal{A}$ will not use it.[7] If $\mathcal{A}$ decides to follow rule (B), he will corrupt a single player and set the corresponding bit x_i to the value he wants. Therefore, since Φ claims to be $2b$-resilient, $\mathcal{A}$ can use rule (B) exactly b times.

Hence, we reduced the behavior of $\mathcal{A}$ to the following. For $i = 1 \dots D$, the adversary $\mathcal{A}$ can generate x_i given $x_1, \dots, x_{i-1}$ using one of the following rules:

(A') Set x_i to 0 with any probability inside the interval $[\gamma, 1-\gamma]$.

(B') Set x_i to any value $\mathcal{A}$ desires. However, this rule can be used at most b times.

Thus, we can view our adversary $\mathcal{A}$ as an *imperfect random source* that emits D history dependent weakly random bits according to rules (A') and (B'), and can view our function $f : \{0,1\}^D \to \{0,1\}$ as the bit-extraction procedure trying to extract a single slightly random bit for *any* such source $\mathcal{A}$.

Definition 2 *Call any $\mathcal{A}$ obeying rules (A') and (B') above a* (γ, b, D)-imperfect random source, *or* (γ, b, D)*-IRS. Given* $f : \{0,1\}^D \to \{0,1\}$, *we let*

- $q(\gamma, b, D, f, \mathcal{A})$ *be the fairness of the coin* $f(x)$, *where* $x = x_1, \dots, x_D$ *was produced by* $\mathcal{A}$.
- $q(\gamma, b, D, f) = \min_{\mathcal{A}} q(\gamma, b, D, f, \mathcal{A})$ *(taken over all* (γ, b, D)*-IRS* $\mathcal{A}$*).*
- $q(\gamma, b, D) = \max_f q(\gamma, b, D, f)$ *(taken over all* $f : \{0,1\}^D \to \{0,1\}$*).*

Thus, $q(\gamma, b, D)$ *is the best fairness of a coin that can be extracted from any* (γ, b, D)*-IRS. Similar to collective coin-flipping, we say that one can extract a slightly random bit if* $q(\gamma, b, D) = \Omega(1)$, *and an almost perfect bit if* $q(\gamma, b, D) = \frac{1}{2} - o(1)$.

We will talk more about the relation of our IRS to two classical IRS of [SV86, LLS89], but let us right away state one of our main impossibility results for our IRS.

Theorem 3
$$q(\gamma, b, D) \leq \frac{2}{(2-2\gamma)^b} \tag{1}$$

In particular, if $b \cdot (\frac{1}{2} - \gamma) = \omega(1)$, *then* $q(\gamma, b, D) = o(1)$, *i.e. it is impossible to extract a slightly random bit.*

The amazing fact about Equation (1) is that it does not depend on the number of generated bits D! In other words, more generated bits do not help for a given γ and b. Tracing back to the adaptive coin-flipping, once we decided to achieve adaptive $2b(n)$-resilience, there is fundamental limitation on how fair we can make the resulting coin, irrespective of how many times we run the black-box protocol Π. In other words, our informal intuition was wrong, when we claimed that we

[7] In fact, $\mathcal{A}$ that we construct will always set the 0-probability of x_i to either γ, or to $(1-\gamma)$, and no other values.

should be able to "overcome" any number b of completely biased bits when having an overwhelming majority of $(D-b)$ slightly random bits.

Before moving back to our imperfect random source, we right away apply Theorem 3 to establish the impossibility of black-box reductions given by Theorem 2. Recall that we concluded that it is impossible to obtain a weakly adaptively $2b$-resilient $\Phi(b, D, \Pi)$ if it is impossible to extract a slightly random bit from a (γ, b, D)-IRS, where $\gamma = \gamma_\Pi(b)$. From the upper bound of Ben-Or and Linial [BL90] that we mentioned in Section 1, we know that for any coin-flipping protocol Π and any b, some b players can bias the coin to have fairness at most $\frac{1}{2} - \Omega(\frac{b}{n})$. Thus, $\gamma = \gamma(b) \leq \frac{1}{2} - \Omega(\frac{b}{n})$, i.e. $b(\frac{1}{2} - \gamma) = \Omega(b^2/n)$. By Theorem 3, it is impossible to extract a slightly random bit whenever $b^2 = \omega(n)$, i.e. $b = \omega(\sqrt{n})$, establishing Theorem 2.[8]

4 Analysis of the Imperfect Random Source

In the remainder of the paper, we discuss our new random source, relate it to earlier imperfect random sources, and analyze its properties (in particular, prove Theorem 3), which are of independent interest.

4.1 Bit-Fixing Source of Lichtenstein, Linial and Saks [LLS89]

Lichtenstein, Linial and Saks [LLS89] considered the case of $\gamma = \frac{1}{2}$, i.e. essentially $\mathcal{A}$ can only use rule (B'). Thus, there is a sequence of D *truly* random bits, b of which can be deterministically fixed by $\mathcal{A}$. This source is called *bit-fixing*. As usual, the question is whether we can extract at least one slightly random random bit from this source. Notice, that if we let f to be the majority function, we can tolerate $b = O(\sqrt{D})$ since any $c\sqrt{D}$ bits (for small enough constant c) do not influence the resulting majority with probability $1 - o(1)$. Remarkably enough, Lichtinstein, Linial and Saks [LLS89] actually showed that this is the best bit extraction possible. Namely,

Theorem 4 ([LLS89]) *$q(\frac{1}{2}, c_1\sqrt{D}, D) = \frac{1}{2} - o(1)$, while $q(\frac{1}{2}, c_2\sqrt{D}, D) = o(1)$ (for some c_1 and c_2). Moreover, majority is the best bit-extraction function f.*

Notice that this result implies Theorem 1 we mentioned earlier. Indeed, in the coin-flipping protocols honest player send truly unbiased coin flips, while dishonest players send arbitrary bits. Thus, we have exactly the source in the above theorem, except adversary $\mathcal{A}$ cannot make *arbitrary interventions*, he can only intervene if the player is faulty. However, when each player sends at most 1 bit (i.e, n bits are sent overall) $\mathcal{A}$ can indeed intervene arbitrarily and we get Theorem 1. Unfortunately, the reasoning does not extend when players send more than 1 bit. Thus, using completely different reasoning, our approach and that of [LLS89] coincidentally reduced different problems at hand about adaptive coin-flipping to similar looking IRS.

As a side note, a *random* function $f : \{0,1\}^D \to \{0,1\}$ is a terrible bit-extraction function for the bit-fixing source even for $b = \omega(1)$, since with high probability the first $(D-b)$ bits do not fix f, so $\mathcal{A}$ can simply wait and set the last b bits to fix f to either 0 or 1. Another terrible function (even for $b = 1$) is any parity function: $\mathcal{A}$ can fix it by fixing the last bit of this parity.

To summarize, when $\gamma = \frac{1}{2}$ we can tolerate $b = O(\sqrt{D})$, and the majority is the best such function. However, a random function will not do the job even if $b = \omega(1)$.

[8] If we want to extract *almost* random bit, it is impossible to do it if $b = \Omega(\sqrt{n})$.

4.2 Slightly-Random Source of Sántha and Vazirani [SV86]

Sántha and Vazirani [SV86] looked at the case $b = 0$, i.e. $\mathcal{A}$ can only use rule (A'). Thus, $\mathcal{A}$ can set $\Pr(x_i = 0 \mid x_1 \dots x_{i-1})$ anywhere within $[\gamma, 1-\gamma]$. This source is sometimes referred to as the *slightly-random source* or also *SV-source*.

On a negative side, Sántha and Vazirani showed that one cannot extract $\tilde{\gamma}$-nontrivial bits for any $\tilde{\gamma} > \gamma$. Thus, the adversary $\mathcal{A}$ can always make sure that the resulting bit $f(x_1, \dots, x_D)$ is not better than any of the individual bits x_i. On the positive side, there are many f's that produce γ-fair bits, for example $f(x_1, \dots, x_D) = x_i$ (for any i), or, more generally, any non-trivial parity function of the input bits. Thus,

Theorem 5 ([SV86]) *$q(\gamma, 0, D) = \gamma$. Thus, one can extract a slightly random bit iff $\gamma = \Omega(1)$.*

Notice, similarly to our Theorem 3, the number of bits D does not help. However, it is completely trivial to extract a slightly random bit (just output x_1) if $\gamma = \Omega(1)$. In fact, Boppana and Narayanan [BN96], following the ideas of Alon and Rabin [AR89] and elegantly extending their techniques, showed much more.

Theorem 6 ([AR89, BN96]) *For any (constant) $\gamma > 0$ there exists a constant $\gamma_0 > 0$ such that with probability exponentially close to 1, a* random *function $f : \{0,1\}^D \to \{0,1\}$ satisfies $q(\gamma, 0, D, f) \geq \gamma_0$.*

Thus, a vast majority of functions extract a slightly random bit from any *SV*-source. Unfortunately, majority is not one of these functions. Indeed, if the adversary always sets the 1-probability of the next bit to be $1 - \gamma$, the resulting bit will be 1 with probability $1 - o(1)$. In fact, Alon and Rabin [AR89] showed that *majority is the worst* bit-extracting function. Namely, $q(\gamma, 0, D, majority) \leq q(\gamma, 0, D, f)$, for any f.

Hence, if $b = 0$, a random function is a good bit extractor, while the majority is the worst.

4.3 Our Combined Source

We see that (γ, b, D)-source generalizes both of the bit-fixing and the SV-sources (which roughly correspond to using only one of rules (A') or (B')). While for the interesting settings of parameters (e.g., $b = O(\sqrt{D})$ for bit-fixing, and constant $\delta > 0$ for SV), we can extract slightly random bits from both of these sources, the functions achieving this are drastically different. For the bit-fixing source the best function was majority, and a random function (or any parity function) was terrible, while for the SV-source a random function was good (and any parity function is optimal), while the majority was the worst. So best extractor becomes the worst and vice versa! One may wonder if it is indeed possible to combine "the best of two worlds" and extract a slightly random bit from our combined source. Unfortunately, Theorem 3 says that this is impossible for essentially any interesting setting of parameters. The most striking such setting, perhaps, is $b = \omega(1)$ and any constant $\gamma < \frac{1}{2}$. If we interpret $b = \omega(1)$ as $b \to \infty$, this says that no matter how large we make D (given b), it is still impossible to extract even a single slightly random bit when $\gamma < \frac{1}{2}$.

We now state our results more precisely. In what follows from here on, γ will never change, so we omit it from all the notation. Note that given the extraction function f, the optimal adversary does the following. First $\mathcal{A}$ tries (in his mind) to minimize the probability that the resulting coin $\sigma = 0$, then he does the same with $\sigma = 1$, and then chooses the smaller of the above. Therefore, it is more convenient for us to analyze $\mathcal{A}$ that, given f, tries to avoid a particular σ, say $\sigma = 0$. In this case, however, the success of $\mathcal{A}$ will crucially depend on how biased towards 1 the function f

is: if $f \equiv 0$, nothing could be done, while if $f \equiv 1$, nothing needs to be done. This motivates the following definition.

Definition 3 *Given $f : \{0,1\}^D \to \{0,1\}$, denote by $Ones(f) = |\{x \in \{0,1\}^D \text{ s.t. } f(x) = 1\}|$. We let*

$$p(t, D, b) = \max_{f} \min_{\mathcal{A}} \Pr(f(x) = 0)$$

where the maximum is taken over all $f : \{0,1\}^D \to \{0,1\}$ with $Ones(f) = t$, and the minimum is taken over all adversaries $\mathcal{A}$ producing $x = x_1 \dots x_D$ and satisfying rules (A') and (B'). In other words, we restrict ourselves to extracting functions having t preimages of 1, and see how biased towards 1 the adversary of our source can make the resulting coin.

In the terminology of [LLS89], we can define the *language* L associated with f as $L = \{x \mid f(x) = 1\}$. Then we can view our adversary as trying to force $x \in L$. The quantity $p(t, D, b)$ tells us how the probability of failure ($x \notin L$, i.e. $f(x) = 0$) of the adversary over the worst possible languages L (over D-bit strings) of cardinality t.

Theorem 7

$$p(t, D, b) \leq \frac{2^D}{t} \cdot \frac{1}{(2 - 2\gamma)^b} \tag{2}$$

We notice that $t/2^D$ is simply the fraction of x such that $f(x) = 1$. Thus, Equation (2) says for any $f : \{0,1\}^D \to \{0,1\}$, we can upper bound the probability of adversary's failure to fix $f(x) = 1$ by a function depending only on the $density(f) \stackrel{\text{def}}{=} Ones(f)/2^D$, i.e. only the fraction of "ones" of f matters! Since any function either has a majority of "ones" or "zeros", by replacing, if necessary, 0 and 1 we can assume that $Ones(f) \geq 2^{D-1}$, i.e. $2^D/t \leq 2$. This immediately implies Theorem 3. In fact, to make the coin not ε-fair, it suffices for the adversary to have the number of interventions $b = O(\frac{1}{1-2\gamma}) \cdot \log(\frac{1}{\varepsilon})$. We now prove Theorem 7.

Proof: The statement is true for $\gamma = \frac{1}{2}$ or $b = 1$, since $p(\cdot, \cdot, \cdot) \leq 1 \leq 2^D/t$, so assume $\gamma < \frac{1}{2}$ and $b \geq 1$. Let $a = t/2^D$ be the fraction of "ones" of f, and define $g(a, b) = \frac{1}{a(2-2\gamma)^b}$. We need to show that $p(t, D, b) \leq g(a, b)$ for any $D \geq 1$, $1 \leq b \leq D$ and $0 \leq t \leq 2^D$. We prove this by induction on D. For $D = 1$, $p(0, 1, b) = 1 < \infty = g(0, b)$, and $p(1, 1, b) = p(2, 1, b) = 0 \leq g(a, b)$ (here we used $b \geq 1$, so that we can take the branch leading to 1). Assume now the claim is true for $(D - 1)$ and we want to show it for D.

Take any f such that $Ones(f) = t$. Let $f_0 : \{0,1\}^{D-1} \to \{0,1\}$ be the restriction of f when $x_0 = 0$. Similarly for f_1. Let $\ell = Ones(f_0)$ and $r = Ones(f_1)$. Clearly, $\ell + r = t$. Without loss of generality assume $\ell \geq r$ (if not, we reverse ℓ and r everywhere in the proof). Given such f, our particular adversary $\mathcal{A}$ will consider two options: either he will use rule (B') (he can do it since we assumed $b \geq 1$) and fix $x_0 = 0$, reducing the question to that of analyzing the function f_0 with $Ones(f_0) = \ell$ on $D - 1$ variables and also reducing b by 1, or he will use rule (A') making the 0-probability of x_0 equal to $1 - \gamma$ and leaving the same b. By the definition of function $p(t, D, b)$, we know that in the first case the failure probability of $\mathcal{A}$ will be at most $p(\ell, D - 1, b - 1)$, and in the second case it will be at most $\gamma \cdot p(r, D - 1, b) + (1 - \gamma) \cdot p(\ell, D - 1, b)$. Given f, our adversary will choose the best (i.e., the smallest) of these two quantities. Since the choice of $\ell \geq r$ such that $\ell + r = t$ is outside of our control, we will take the maximum over all such choices and obtain the following recurrence.

$$p(t, D, b) \leq \max_{\substack{0 \leq r \leq t/2 \\ \ell = t - r}} \min \left[p(\ell, D - 1, b - 1) \ , \ \gamma \cdot p(r, D - 1, b) + (1 - \gamma) \cdot p(\ell, D - 1, b) \right] \tag{3}$$

Let $\ell/2^{D-1} = a+\beta$ and $r/2^{D-1} = a-\beta$, where $\beta \geq 0$ (since $\ell + r = t = a2^D$ and $\ell \geq r$). Using our inductive assumption on $(D-1)$, we get

$$p(t,D,b) \leq \max_{0\leq\beta\leq\min(a,1-a)} \min\left(g(a+\beta,b-1),\ \ \gamma g(a-\beta,b) + (1-\gamma)g(a+\beta,b)\right) \stackrel{?}{\leq} g(a,b) \quad (4)$$

Recalling the definition of g, it thus suffices to show that

$$\max_{0\leq\beta\leq\min(a,1-a)} \min\left(\frac{1}{(a+\beta)(2-2\gamma)^{b-1}}\ ,\ \frac{\gamma}{(a-\beta)(2-2\gamma)^b} + \frac{1-\gamma}{(a+\beta)(2-2\gamma)^b}\right) \leq \frac{1}{a(2-2\gamma)^b}$$

$$\iff \max_{0\leq\beta\leq\min(a,1-a)} \min\left(\frac{2-2\gamma}{a+\beta}\ ,\ \frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta}\right) \leq \frac{1}{a}$$

To show the last equation, we see when it is the case that $\frac{2-2\gamma}{a+\beta} = \frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta}$, i.e. the expressions under the min are equal. It is not hard to see that this happens when $\beta = (1-2\gamma)a$. We now consider two cases.

- Case 1. Assume $\beta \geq (1-2\gamma)a$. Then $\min\left(\frac{2-2\gamma}{a+\beta}\ ,\ \frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta}\right) = \frac{2-2\gamma}{a+\beta}$ and it suffices to show that $\frac{2-2\gamma}{a+\beta} \leq \frac{1}{a}$. But it is easy to see that the latter is *exactly equivalent* to our assumption on β, so it holds.
- Case 2. Assume $\beta \leq (1-2\gamma)a$. Then $\min\left(\frac{2-2\gamma}{a+\beta}\ ,\ \frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta}\right) = \frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta}$ and it suffices to show that $\frac{\gamma}{a-\beta} + \frac{1-\gamma}{a+\beta} \leq \frac{1}{a}$. But this is again *exactly equivalent* to our assumption on β, so it holds.

■

4.4 Expected Number of Interventions to Fix the Outcome

Finally, we analyze another property of our IRS. Assume that rather than having at most b applications of rule (B') and trying to minimize the fairness of the coin, the adversary tries to fix the coin to some value he desires (with probability 1) and wants to minimize the expected number of "interventions", i.e. applications of rule (B') (while rule (A') can be used "for free"). In other words, given an extraction function f, $\mathcal{A}$ computes the expected number of interventions to force 0, than does the same for 1, and chooses the smaller of the two. We let $v(\gamma, D)$ be this smallest expected number of interventions taken over the worst possible extraction function $f : \{0,1\}^D \to \{0,1\}$.

Theorem 8

$$v(\gamma, D) \leq O\left(\frac{1}{1-2\gamma}\right) \quad (5)$$

In particular, if $\gamma < \frac{1}{2}$, a constant expected number of interventions suffice irrespective of D!

We again see a similar trend to Theorem 5 and Theorem 3: large number of repetitions D does not help. In other words, our "combined" random source gives much more power to the adversary than one would imagine: if (constant) $\gamma < \frac{1}{2}$ and no matter how large is D, a super-constant number of interventions b makes it impossible to extract a slightly random bit, and a constant expected number of interventions suffices to fix the bit no matter what extraction function we use. We also remark that Theorems 3 and 8 about our IRS are complimentary to each other (i.e. one does not imply the other), even though both suffice to establish our main Theorem 2. Indeed, we already saw that Theorem 2 follows from the claim that $b(\frac{1}{2} - \gamma) = \omega(1) \Rightarrow q(\gamma, b, D) = o(1)$ (which

was immediate from Theorem 3). But this claim also follows from Theorem 8 by applying Markov's inequality and getting that $b = O(1/(\varepsilon(1-2\gamma)))$ suffices to make $q(\gamma, b, D) \leq \varepsilon$, which gives the needed $b(\frac{1}{2} - \gamma) = \omega(1) \Rightarrow q(\gamma, b, D) = o(1)$.

Similarly to the proof of Theorem 3, it is more convenient to analyze $\mathcal{A}$ that always forces a particular outcome (say, 1) with probability 1 and tries to minimize the number of interventions b. We again consider extraction functions f with $Ones(f) = t$ and omit γ from the notation below.

Definition 4 *We let*

$$e(t, D) = \max_{f} \min_{\mathcal{A}} \mathbf{E}[b]$$

where the maximum is taken over all $f : \{0,1\}^D \to \{0,1\}$ with $Ones(f) = t$, the minimum is taken over all adversaries $\mathcal{A}$ following rules (A') and (B') and necessarily producing $x = x_1 \ldots x_D$ satisfying $f(x) = 1$, and $\mathbf{E}[b]$ stands for the expected number of applications of rule (B') by $\mathcal{A}$ (taken over the random choices involved in using rule (A')). In other words, we restrict ourselves to extracting functions having t preimages of 1, and see how many interventions $\mathcal{A}$ needs on average to ensure $f(x) = 1$.

In the terminology of [LLS89], we can define the *language* L associated with f as $L = \{x \mid f(x) = 1\}$. Then we can view our adversary as trying to ensure that $x \in L$ with the smallest number of interventions. The quantity $e(t, D)$ tells us this expected number of interventions that $\mathcal{A}$ over the worst possible languages L (over D-bit strings) of cardinality t. In order to state a bound on $e(t, D)$, we need the following easily verified analytical lemma.

Lemma 1 *For any $0 < \gamma < \frac{1}{2}$ the equation*

$$z^{\frac{1}{\gamma}} + 1 = 2 \cdot z^{\frac{1}{\gamma}-1} \tag{6}$$

has a unique solution $z_\gamma \in (1,2)$. In addition, z_γ is a continuous decreasing function of γ such that $\lim_{\gamma \to 0} z_\gamma = 2$, $\lim_{\gamma \to \frac{1}{2}} z_\gamma = 1$, $\log_2 z_\gamma = \Theta(1-2\gamma)$, and for all $1 \leq w \leq z_\gamma$ we have $w^{1/\gamma} + 1 \leq 2 \cdot w^{1/\gamma - 1}$.

Theorem 9

$$e(t, D) \leq \log_{z_\gamma}\left(\frac{2^D}{t}\right) = \frac{\log_2(2^D/t)}{\log_2 z_\gamma} = O\left(\frac{1}{1-2\gamma}\right) \cdot \log(2^D/t) \tag{7}$$

Again, Equation (7) says for any $f : \{0,1\}^D \to \{0,1\}$, we can upper bound the expected number of interventions to force $f(x) = 1$ by a function depending only on the $density(f) = Ones(f)/2^D$, i.e. only the fraction of "ones" of f matters! Since any function either has a majority of "ones" or "zeros", by replacing, if necessary, 0 and 1 we can assume that $Ones(f) \geq 2^{D-1}$, i.e. $2^D/t \leq 2$. This immediately implies Theorem 8. We now prove Theorem 9 using almost the same technique we used in Theorem 7.

Proof: Let $a = t/2^D$ be the fraction of "ones" of f, $z = z_\gamma$ and define $h(a) = \log_z(1/a)$. We need to show that $e(t, D) \leq h(a)$ for any $D \geq 1$ and $0 \leq t \leq 2^D$. We prove this by induction on D. For $D = 1$, $e(0,1) = \infty = h(0)$, and $e(1,1) = 1 \leq \log_z 2 = h(\frac{1}{2})$ (since $z \leq 2$, and here $a = \frac{1}{2}$) and $e(2,1) = 0 = h(1)$. Assume now the claim is true for $(D-1)$ and we want to show it for D.

Let f, f_0, f_1, r, ℓ have the same meaning they had in the proof of Theorem 7. In fact, our adversary $\mathcal{A}$ will be the same as well! In other words, he will consider spending one intervention to set $x_0 = 0$ versus saving the intervention and making the 0-probability of x_0 equal to $1 - \gamma$. The

only difference is that in the setting of Theorem 7 $\mathcal{A}$ could "run out" of his b interventions and also minimized a different quantity $p(t, D, b)$ with different initial conditions, while in our case $\mathcal{A}$ will use an extra intervention if this pays off. We get the following recurrence.

$$\begin{aligned} e(t,D) &\leq \max_{\substack{0\leq r\leq t/2 \\ \ell=t-r}} \min\left[e(\ell, D-1)+1\ ,\ \gamma\cdot e(r,D-1)+(1-\gamma)\cdot e(\ell,D-1)\right] &(8)\\ &= \max_{\substack{0\leq r\leq t/2 \\ \ell=t-r}} \left(\ e(\ell,D-1)+\min\left[\ 1\ ,\ \gamma\cdot\{e(r,D-1)-e(\ell,D-1)\}\ \right]\ \right) &(9)\end{aligned}$$

Let $\ell/2^{D-1} = a+\beta$ and $r/2^{D-1} = a-\beta$, where $\beta \geq 0$ (since $\ell + r = t = a2^D$ and $\ell \geq r$). Using our inductive assumption on $(D-1)$, we get

$$e(t,D) \leq \max_{0\leq\beta\leq\min(a,1-a)} \left(\ h(a+\beta)+\min\left[\ 1\ ,\ \gamma\cdot\{h(a-\beta)-h(a+\beta)\}\ \right]\ \right) \stackrel{?}{\leq} h(a) \qquad (10)$$

Recalling the definition of h, it thus suffices to show that

$$\max_{0\leq\beta\leq\min(a,1-a)} \left(\ \log_z \frac{1}{a+\beta} + \min\left[\ 1\ ,\ \gamma\cdot\log_z\frac{a+\beta}{a-\beta}\ \right]\ \right) \leq \log_z \frac{1}{a}$$

It will now be convenient to make change of variable and let $\beta = a\cdot\frac{c-1}{c+1}$ for some $c \geq 1$ (this is always possible because $0 \leq \beta \leq a$). Noticing that $a-\beta = 2/(c+1)$, $a+\beta = 2c/(c+1)$ and $1 = \log_z z$, we get that it suffices to show that

$$\begin{aligned} \max_{c\geq 1}\left(\ \log_z\left(\frac{c+1}{2c\cdot a}\right)+\min\left[\ \log_z z\ ,\ \gamma\cdot\log_z c\ \right]\ \right) &\leq \log_z\frac{1}{a} \quad\Longleftrightarrow\\ \max_{c\geq 1}\ \min\left[\frac{(c+1)z}{2c\cdot a}, \frac{(c+1)c^\gamma}{2c\cdot a}\right] &\leq \frac{1}{a} \quad\Longleftrightarrow\\ \max_{c\geq 1}\left(\ \frac{c+1}{2c}\cdot\min\left[z, c^\gamma\right]\ \right) &\leq 1\end{aligned}$$

We now make the final change of variable, letting $c = w^{1/\gamma}$. Then it suffices to show that

$$\max_{w\geq 1}\left(\ \frac{w^{1/\gamma}+1}{2w^{1/\gamma}}\cdot\min\left[z,w\right]\ \right) \leq 1 \qquad (11)$$

To show the last equation, we consider two cases.

- Case 1. Assume $w \leq z$. Then $\min[z,w] = w$ and it suffices to show $w^{1/\gamma}+1 \leq 2w^{1/\gamma-1}$, which follows from Lemma 1 since $1 \leq w \leq z$ by our assumption.
- Case 2. Assume $w \geq z$. Then $\min[z,w] = z$ and it suffices to show $(w^{1/\gamma}+1)z \leq 2w^{1/\gamma}$, which is the same as $w^{1/\gamma} \geq z/(2-z)$. But since $z = z_\gamma$ is the solution to Equation (6), it is easy to see that $z/(2-z) = z^{1/\gamma}$, so it suffices to show $w^{1/\gamma} \geq z^{1/\gamma}$, which is the same as our assumption $w \geq z$.

■

To summarize the properties of our "combined" imperfect random source, we have shown that it gives too much power to the adversary, perhaps more than one would expect.

5 Conclusions

We have seen that Theorems 1 and 2 give very different evidences in support of Conjecture 1. However, the status of coin-flipping with adaptive adversaries is still open and it would be very interesting to resolve it.

References

[AL93] M. Ajtai, N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[AN93] N. Alon, M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM J. Comput.*, 22(2):403-417, 1993.

[AR89] N. Alon, M. Rabin. Biased Coins and Randomized Algorithms. *Advances in Computing Research*, 5:499-507, 1989.

[BL90] M. Ben-Or, N. Linial. Collective Coin-Flipping. In Slivio Micali, editor, *Randomness and Computation*, pp. 91-115, Academic Press, New York, 1990.

[BN96] R. Boppana, B. Narayanan. The Biased Coin Problem. *SIAM J. Discrete Math.*, 9(1)29–36, 1996.

[BN] R. Boppana, B. Narayanan. Perfect-information Leader Election with Optimal Resilience. *SIAM J. Comput., to appear.*

[CL95] J. Cooper, N. Linial. Fast perfect-information leader-election protocols with linear immunity. *Combinatorica*, 15:319–332, 1995.

[F99] U. Feige. Noncryptographic Selection Protocols. In *Proc. of* 40*th FOCS*, pp. 142–152, 1999.

[GGL98] O. Goldreich, S. Goldwasser, N. Linial. Fault-Tolerant Computation in the Full Information Model. *SIAM J. Comput.*, 27(2):506–544, 1998.

[KKL89] J. Kahn, G. Kalai, N. Linial. The Influence of Variables on Boolean Functions. In *Proc. of* 29*th FOCS*, pp. 68–80, 1989.

[LLS89] D. Lichtenstein, N. Linial, M. Saks. Some Extremal Problems Arising from Discrete Control Processes. *Combinatorica*, 9:269–287, 1989.

[ORV94] R. Ostrovsky, S. Rajagopalan, U. Vazirani. Simple and Efficient Leader Election in the Full Information Model. In *Proc. of* 26*th STOC*, pp. 234–242, 1994.

[RSZ99] A. Russell, M. Saks, D. Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. In *Proc. of* 31*st STOC*, pp. 339–347, 1999.

[RZ98] A. Russell, D. Zuckerman. Perfect information leader election in $\log^* n + O(1)$ rounds. In *Proc. of* 39*th FOCS*, pp. 576–583, 1998.

[S89] M. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM J. Discrete Math.*, 2(2):240–244, 1989.

[SV86] M. Sántha, U. Vazirani. Generating Quasi-Random Sequences from Semi-Random Sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'