2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Tighter Bounds on Multi-Party Coin Flipping via Augmented Weak Martingales and Differentially Private Sampling

Amos Beimel
Department of Computer Science, Ben-Gurion University
amos.beimel@gmail.com

Iftach Haitner
School of Computer Science, Tel Aviv University
iftachh@cs.tau.ac.il

Nikolaos Makriyannis
School of Computer Science, Tel Aviv University
n.makriyannis@gmail.com

Eran Omri
Department of Computer Science, Ariel University
omrier@ariel.ac.il

***Abstract*—In his seminal work, Cleve [STOC '86] has proved that any r-round coin-flipping protocol can be efficiently biased by $\Theta(1/r)$. This lower bound was met for the two-party case by Moran, Naor, and Segev [Journal of Cryptology '16], and the three-party case (up to a $\mathrm{polylog}$ factor) by Haitner and Tsfadia [SICOMP '17], and was approached for n-party protocols when $n < \mathrm{loglog}\, r$ by Buchbinder, Haitner, Levi, and Tsfadia [SODA '17]. For $n > \mathrm{loglog}\, r$, however, the best bias for n-party coin-flipping protocols remains $O(n/\sqrt{r})$ achieved by the majority protocol of Awerbuch, Blum, Chor, Goldwasser, and Micali [Manuscript '85].**

Our main result is a tighter lower bound on the bias of coin-flipping protocols, showing that, for every constant $\varepsilon > 0$, an r^{ε}-party r-round coin-flipping protocol can be efficiently biased by $\widetilde{\Omega}(1/\sqrt{r})$. As far as we know, this is the first improvement of Cleve's bound, and is only $n = r^{\varepsilon}$ (multiplicative) far from the aforementioned upper bound of Awerbuch et al.

We prove the above bound using two new results that we believe are of independent interest. The first result is that a sequence of ("augmented") weak martingales have large gap: with constant probability there exists two adjacent variables whose gap is at least the ratio between the gap between the first and last variables and the square root of the number of variables. This generalizes over the result of Cleve and Impagliazzo [Manuscript '93], who showed that the above holds for strong martingales, and allows in some setting to exploit this gap by efficient algorithms. We prove the above using a novel argument that does not follow the more complicated approach of Cleve and Impagliazzo. The second result is a new sampling algorithm that uses a differentially private mechanism to minimize the effect of data divergence.

***Keywords*-Coin Flipping; Bias; Lower Bounds; Cryptography; Martingales; Differential Privacy**

1. Introduction

In a coin-flipping protocol, introduced by Blum [7], the parties wish to output a common (close to) unbiased bit, even though some of the parties may be corrupted and try to bias the output. More formally, such protocols should satisfy the following two properties: first, when all parties are honest (i.e., follow the prescribed protocol), they all output the *same* unbiased bit. Second, even when some parties are corrupted (i.e., collude and arbitrarily deviate from the protocol), the remaining parties should still output the same bit, and this bit should not be too biased (i.e., its distribution should be close to being uniform over $\{0,1\}$). We emphasize that the above requirements stipulate that the honest parties should *always* output a common bit, regardless of what the corrupted parties do, and in particular they are not allowed to abort if a cheat was detected.[1] Coin flipping is a fundamental primitive with numerous applications, and thus lower bounds on coin flipping protocols imply analogous bounds on many other basic cryptographic primitives, including other inputless primitives and secure computation of functions that have input (e.g., XOR).

In his seminal work, Cleve [1] showed that for *any* efficient two-party r-round coin-flipping protocol, there exists an efficient adversarial strategy that biases the output of the honest party by $\Theta(1/r)$, and his bound extends to the multi-party case with no honest majority, via a simple reduction. The above lower bound on coin-flipping protocols was met for the two-party case by Moran, Naor, and Segev [2] and for the three-party case (up to a $\mathrm{polylog}$ factor) by Haitner and Tsfadia [3], and was approached for n-party coin-flipping protocols when $n < \mathrm{loglog}\, r$ by Buchbinder, Haitner, Levi, and Tsfadia [4]. For $n > \mathrm{loglog}\, r$, however, the smallest bias for n-party coin-flipping protocol remains $\Theta(n/\sqrt{r})$, achieved by the majority protocol of Awerbuch, Blum,

[1] Such protocols are typically addressed as having *guaranteed output delivery*, or, abusing terminology, as *fair*.

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00084

Chor, Goldwasser, and Micali [5].

A. Our Results

Our main result is the following lower bound on the security of coin-flipping protocols.

Theorem 1.1 (Main result, informal)**.** *For any n-party r-round coin-flipping protocol with $n^k \geq r$ for some $k \in \mathbb{N}$, there exists a fail-stop[2] adversary running in time n^k, corrupting all parties but one, that biases the output of honest party by $1/(\sqrt{r} \cdot \log(r)^k)$.*

As a concrete example, assume the number of parties is $n = r^{1/100}$. The above theorem yields an attack of bias $\widetilde{\Omega}(1/\sqrt{r}) = \widetilde{\Omega}(1/r^{0.5})$, to be compared to the $n/\sqrt{r} = 1/r^{0.49}$ upper bound of Awerbuch et al. [5]. As far as we know, Theorem 1.1 is the first improvement over the $\Omega(1/r)$ bound of Cleve [1].

Theorem 1.1 is only applicable when the adversary is able to corrupt all parties but one. However, by grouping parties together, we note that any n-party protocol is a $\lfloor n/s \rfloor$-party protocol, for any $s < n$, and thus the following theorem dealing with more versatile corruption strategies follows by simple reduction.

Theorem 1.2 (Main result, fewer corruptions variant, informal)**.** *For n-party r-round coin-flipping protocol with $(n/s)^k \geq r$ for some $s < n/2$ and $k \in \mathbb{N}$, there exists an adversary running in time $(n/s)^k$, corrupting all parties but a subset of size s, that biases the output of honest parties by $1/(\sqrt{r} \cdot \log(r)^k)$.*

For instance, if $n^k > r$, by corrupting all parties but a subset of size $n^{1/2}$, the adversary achieves a bias of $1/(\sqrt{r} \cdot \log(r)^{2k})$. That is, up to a factor of $1/\log(r)^k$ in the bias, we derive the same result as Theorem 1.1, but with fewer corrupted parties (only all parties but a subset of size $n^{1/2}$ instead of all parties but one).

We prove the above theorems using the following two results that we believe to be of independent interest.

1) Augmented Weak Martingales have Large Gap: A sequence $X_1, \ldots, X_r$ of random variables is a *(strong) martingale*, if $\mathbf{E}[X_i \mid X_{\leq i-1}] = X_{i-1}$ for every $i \in [r]$ (letting $X_{\leq j} = (X_1, \ldots, X_j)$). Cleve and Impagliazzo [6] showed that any strong martingale sequence with $X_1 = \frac{1}{2}$ and $X_r \in \{0, 1\}$ has a $1/\sqrt{r}$ gap with constant probability: with constant probability, $|X_i - X_{i-1}| \geq \Omega(1/\sqrt{r})$ for some $i \in [r]$. This result is the core of their proof showing that there exists an *inefficient* (fail-stop) attack for any coin-flipping protocol that yields a bias of order $1/\sqrt{r}$ (see Section 1-B). The result of [6] is used with respect to the *Doob martingale* sequence defined by $X_i = \mathbf{E}[f(Z) \mid Z_{\leq i}]$, for random variables $Z = (Z_1, \ldots, Z_r)$ and a function f of interest. To be applicable in a computational setting, we require that $X_i = \mathbf{E}[f(Z) \mid Z_{\leq i}]$ is an efficiently computable function of $Z_{\leq i}$. In many cases however, including the one considered by [6], $\mathrm{Supp}(Z_{\leq i})$ is huge, resulting in X_i not being efficiently computable.

Weak martingales, introduced by Nelson [8], is a relaxation of strong martingales where it is only required that $\mathbf{E}[X_i \mid X_{i-1}] = X_{i-1}$. Namely, the conditioning is only on the value of the preceding variable, and not on the whole "history". As in the case of (strong) martingales, for arbitrary $Z = (Z_1, \ldots, Z_r)$ and a function f of interest, we can consider the Doob-like sequence $X_i = \mathbf{E}[f(Z) \mid Z_i, X_{i-1}]$. The support size of the function for computing X_i is only of size $|\mathrm{Supp}(Z_i) \times \mathrm{Supp}(X_{i-1})|$, and we can use discretization to further reduce the support size of X_i (i.e., we let X_i be a *rounding* of $\mathbf{E}[f(Z) \mid Z_i, X_{i-1}]$). Hence, if the support of Z_i is small, the computation of the X_i's can be done efficiently. (Discretization is not useful for the (strong) Doob martingale described above, since, even if the support of each individual Z_1 is small, even 2, the domain of $Z_1, \ldots, Z_r$ is typically huge). Unfortunately, it is unclear whether weak martingales have large gaps, and thus we are unable to apply the attack of Cleve and Impagliazzo [6] using such a sequence.

We prove that a slightly different variant of the Doob construction results in a sequence that is efficiently computable and has a large gap, at the same time. A sequence $X_1, \ldots, X_r$ of random variables is a *sum-of-squares-augmented weak martingales*, if $\mathbf{E}\left[X_i \mid X_{i-1}, \sum_{j \in [i-1]} (X_j - X_{j-1})^2\right] = X_{i-1}$. Namely, X has the "martingale property" when conditioning on some small amount of information about the past. For such a sequence, we prove the following result:

Theorem 1.3 (Informal)**.** *Let $X_1, \ldots, X_r$ be a sequence of sum-of-squares-augmented weak martingale with $X = 1/2$ and $X_r \in \{0, 1\}$, then*

$$\Pr\left[\exists i \in [r] \colon |X_i - X_{i-1}| \geq 1/\sqrt{r}\right] \in \Omega(1).$$

We prove that the above holds for a *rounded* variant of X_i, i.e., X_i are rounded to the closest multiplicative of some $\delta > 0$.

Consider the sequence of sum-of-squares-augmented weak martingales defined by the Doob-like sequence $X_i = \mathbf{E}\left[f(Z) \mid Z_i, X_{i-1}, \sum_{j \in [i-1]} (X_j - X_{j-1})^2\right]$, for arbitrary $Z = (Z_1, \ldots, Z_r)$ and a function f of interest. If the support of the Z_i's small, the computation

[2]Acts honestly, but might abort prematurely.

of the (rounding of) X_i's can be done *efficiently*. This efficiency plays a critical role in our attack on coin-flipping protocols, allowing us, in some cases, to mount an efficient variant of the attack of [6].

Our proof actually yields the following stronger statement.

Theorem 1.4 (Informal)**.** *Let $X_1, \ldots, X_r$ be a sequence of sum-of-squares-augmented weak martingales with $X = 1/2$ and $X_r \in \{0, 1\}$, then*

$$\Pr\Big[\sum_{i\in[r]} (X_i - X_{i-1})^2 \geq 1\Big] \in \Omega(1).$$

Namely, the sum-of-squares is constant with constant probability. In particular, the probability that $|X_i - X_{i-1}| \geq 1/\sqrt{r}$, for some i, is also constant, implying Theorem 1.3. But Theorem 1.4 yields a stronger result: if we are guaranteed that all gaps are at most $1/\sqrt{r}$ (i.e., $|X_i - X_{i-1}| \in O(1/\sqrt{r})$ for all i), then Theorem 1.4 implies that, with constant probability, the sequence has a *linear* number of $1/\sqrt{r}$-gaps (as opposed to only one such gap guaranteed by [6]).

Our proof for Theorem 1.4 is surprisingly simple, and does not follow the more complicated approach of Cleve and Impagliazzo [6].[3]

2) Oblivious Sampling via Differential Privacy: Consider the following r-round game in which the goal is to maximize the revenue of the chosen party: in the beginning, a party H is drawn uniformly from $\mathcal{H}$ (for $\mathcal{H}$ being a finite set of parties). In each round i, values $\left\{s_i^h \in [0,1]\right\}_{h\in\mathcal{H}}$ are assigned to the parties of $\mathcal{H}$, and the values of all parties but H, i.e., $\left\{s_i^h\right\}_{h\in\mathcal{H}\setminus\{H\}}$, are published. Seeing the published values, you can either decide to *abort*, and then party H is rewarded with (the unseen) value s_i^H, or to continue to the next round. If you never choose to abort, then party H is rewarded with s_r^H (the value of the last round). Your goal is to get a reward as close to the optimal value $\gamma = \max_i \left\{s_i := \mathbf{E}_{h\leftarrow\mathcal{H}}\left[s_i^h\right]\right\}$. To make the game reasonable, it is guaranteed that the values assigned to the parties in each round are *similar*: $\left|s_i^h - s_i\right| \leq \sigma$ for every $h \in \mathcal{H}$. Namely, the individual values are σ-close to the mean.

We will be interested in a distributional variant of the above game in which the values of $\left\{s_i^h\right\}$ are not fixed, but rather drawn from some underlying distribution (in our setting, the values of $\left\{s_i^h\right\}$ will be induced by the randomness of the attacked coin-flipping protocol), while satisfying the above guarantees with regards to γ and σ with good enough probability. We refer to the resulting game as an *oblivious sampling game* with parameters r, $|\mathcal{H}|$, γ, and σ. An aborting strategy for the above game can only depend on the game parameters (i.e., $r, |\mathcal{H}|, \gamma, \sigma$) and the values published online.

The simplest aborting strategy for such a game is to abort if the average of all other parties, i.e., $\{s_h\}_{h\in\mathcal{H}\setminus\{H\}}$, is larger than (roughly) $\gamma-\sigma$. The reward of such a strategy is roughly $\gamma - \sigma$, which is useless if $\sigma \geq \gamma$. As we show next, this linear loss in σ is inherent for this strategy; consider a deterministic threshold strategy that aborts if $s_i^{\setminus h} = \mathbf{E}_{h'\leftarrow\mathcal{H}\setminus h}\left[s_i^{h'}\right] \geq \mathsf{tsh}$ for some threshold $\mathsf{tsh} \in [0, \gamma]$. Namely, an aborts occurs if the average value at hand in a given round is greater than tsh. Consider the game defined by $\mathcal{H} = [r-1]$, $s_r^h = \gamma$ for all h, and for $i \in [r-1]$: $s_i^h = \mathsf{tsh} - \sigma$ if $i = h$, and tsh otherwise. It follows that for every value of h, the strategy seeing the values of $\left\{s_i^{\setminus h}\right\}$ aborts at round h, and gets reward $\mathsf{tsh} - \gamma$. Hence, the reward of this strategy is $\mathsf{tsh} - \sigma \leq \gamma - \sigma$.

We show that using a *differentially private* mechanism, and in particular adding Laplace noise to the estimated revenue $s_i^{\setminus h} = \mathbf{E}_{h'\leftarrow\mathcal{H}\setminus h}\left[s_i^{h'}\right]$, significantly improves upon the above deterministic strategy. By introducing such noise, the aborting decision is less correlated to the choice of the random party H. More accurately, the value of H is *σ-differentially private*, according to the definition of Dwork, McSherry, Nissim, and Smith [9], from the aborting decision, and thus we avoid the pitfalls caused by strong correlation between H and the aborting round, as illustrated by the above example for the deterministic threshold strategy. We exploit this "privacy" guarantee to prove the following improvement in the expected reward.

Theorem 1.5 (Informal)**.** *For every oblivious sampling game, the randomized strategy that adds Laplace noise in every round (whose magnitude depends on the game parameters) to $s_i^{\setminus h}$, and aborts if the result is greater than $\gamma/2$, achieves expected reward $\gamma/2 - \sigma^2$.*

Namely, the penalty for having imperfect similarity is reduced from σ to $\gamma/2 + \sigma^2$, a significant improvement when $\gamma < \sigma < 1$. We also prove a generalization of the above theorem where each party has a different similarity guarantee.

B. Our Techniques

Below, we describe the approach for proving Theorem 1.1 using Theorems 1.3 and 1.5. We do not discuss here the proofs of these theorems, but we do explain

[3] To be fair, Cleve and Impagliazzo [6] derive their result by proving an Azuma-like tail inequality for bounded strong martingales that have large gap with only small probability, a bound that we do not prove here.

in Section 1-B5 why the weak martingale used by the attack is computable by an efficient uniform algorithm.

Let Π be an r-round n-party coin-flipping protocol and let out denote the (always common) output of the parties in a random honest execution. By definition, $\mathsf{out} \in \{0,1\}$ and $\mathbf{E}[\mathsf{out}] = 1/2$. Our goal is to obtain an efficient attacker that, by controlling $n-1$ of the parties, biases the honest parties' output by $1/\sqrt{r}$ (we ignore $\log$ factors). We start by describing the $1/\sqrt{r}$ inefficient attack of Cleve and Impagliazzo [6].

1) Cleve and Impagliazzo's Inefficient Attack: Let $n = 2$ and let $(\mathsf{P}_0, \mathsf{P}_1)$ be the parties of Π. Let $T_1, \ldots, T_r$ denote the messages in a random execution of Π. Let $X_i = \mathbf{E}[\mathsf{out} \mid T_{\leq i}]$; namely, X_i is the expected outcome of the protocol given the first i messages $T_{\leq i} = T_1, \ldots, T_i$. It is easy to see that $X_1, \ldots, X_r$ is a (strong) martingale sequence. Hence, the result of [6] described in Section 1-A1 yields that (omitting absolute values and constant factors)

$$\text{Jump:} \quad \Pr\left[\ \exists i \in [r] \colon X_i - X_{i-1} \geq 1/\sqrt{r}\ \right] \in \Omega(1) \tag{1}$$

Backup values.: For $b \in \{0,1\}$, let the *backup value* Z_i^b denote the output of party P_b if party P_{1-b} aborts *after* the i^{th} message was sent, letting Z_r^b be the final output of P_b (if no abort occurs). Using this notation, $\mathbf{E}\left[Z_i^b \mid T_{\leq i}\right]$ is the expected outcome of P_b if P_{1-b} aborts after the i^{th} round. We can assume without loss of generality that

Backup value follows game value:

$$\Pr\left[\exists i \in [r] \colon \left|X_i - \mathbf{E}\left[Z_i^b \mid T_{\leq i}\right]\right| \geq 1/\sqrt{r}\right] \in o(1) \tag{2}$$

for both $b \in \{0,1\}$. Otherwise, the attacker controlling P_{1-b} that computes X_i and $\mathbf{E}\left[Z_i^b \mid T_{\leq i}\right]$ for each round i, and aborts if $X_i - \mathbf{E}\left[Z_i^b \mid T_{\leq i}\right] \geq 1/\sqrt{r}$, would bias P_b's output towards 0 by $1/\sqrt{r}$.[4]

The martingale attack.: The above two observations yield the following attack. From Equations (1) and (2), it follows that without loss of generality

Attack slot:

$$\Pr\left[\begin{array}{l} \exists i \in [r] \colon \mathsf{P}_b \text{ sends the } i^{\text{th}} \text{ message} \quad \wedge \\ \qquad X_i - \mathbf{E}\left[Z_{i-1}^{1-b} \mid T_{\leq i}\right] \geq 1/2\sqrt{r} \end{array}\right] \in \Omega(1) \tag{3}$$

This yields the following attack for party P_b to bias the output of party P_{1-b} towards zero. Before sending the i^{th} message T_i, party P_b aborts if $X_i - \mathbf{E}\left[Z_{i-1}^{1-b} \mid T_{\leq i}\right] \geq 1/2\sqrt{r}$. By Equation (3), under this attack, the output of P_{1-b} is biased towards zero by $\Omega(1/\sqrt{r})$.[5]

The clear limitation of the above attack is that, in many cases, the values of both $X_i = \mathbf{E}[\mathsf{out} \mid T_{\leq i}]$ and $\mathbf{E}\left[Z_i^{1-b} \mid T_{\leq i}\right]$ are *not* efficiently computable (given $T_{\leq i}$). Indeed (assuming the existence of oblivious transfer), the above $\Theta(1/\sqrt{r})$ lower bound does not hold for $n < \log\log r$ [4, 3, 2].

2) Towards an Efficient Attack via Augmented Weak Martingales: The first step towards making the above attack efficient is *not* to define the X_i's as a function of the transcript. Indeed, even given the first message T_1, computing $\mathbf{E}[\mathsf{out} \mid T_1]$ might involve inverting a one-way function. Our solution is to define $X_i^b = \mathbf{E}\left[\mathsf{out} \mid Z_{\leq i}^b\right]$; namely, the expected outcome given P^b's backup values. The immediate advantage is that the backup values are only bits. Thus, X_1^b has only two possible values, and computing it from Z_1 can be done efficiently. Yet, for large values of i, the computation of X_i^b (depending on $Z_1^b, \ldots, Z_i^b$) might still be infeasible.

Thankfully, our new result for sum-of-squares-augmented weak martingales (Theorem 1.3) circumvents this problem. Let $f(Z_1^b, \ldots, Z_r^b) = \mathbf{E}\left[\mathsf{out} \mid Z_{\leq r}^b\right]$. By definition, it holds that $f(Z_1^b, \ldots, Z_r^b) = Z_r^b \in \{0,1\}$, and thus $\mathbf{E}\left[f(Z_1^b, \ldots, Z_r^b)\right] = 1/2$. Theorem 1.3 yields that for the Doob-like sequence $X_i^b = \mathbf{E}\left[\mathsf{out} \mid Z_i^b, X_{i-1}^b, \sum_{j\in[i-1]}(X_j^b - X_{j-1}^b)^2\right]$, it holds that (again, omitting absolute values and constant factors)

$$\text{Jump:} \quad \Pr\left[\exists i \in [r] \colon X_i^b - X_{i-1}^b \geq 1/\sqrt{r}\right] \in \Omega(1) \tag{4}$$

Using a rounded variant of the X_i^b's, the value of X_i^b is only a function of $\left|\mathrm{Supp}(Z_i^b)\right| \cdot r^2 \in O(r^3)$ bits, and thus can be computed efficiently. Namely, the martingale attack of [6] (i.e., aborting in the event of an observed gap) with respect to this definition of X_i is now efficient. Similarly to [6], we obtain an $\Omega(1/\sqrt{r})$

[4] To be more precise, at least one of two attacks would succeed, depending on the aimed direction of the bias.

[5] In more detail, assume for simplicity that P_0 sends the messages $T_1, T_3, \ldots$ and P_1 sends the messages $T_2, T_4, \ldots$. For at least one party P_b, Equation (3) holds when limiting i to be a round where P_b is supposed to send the i^{th} message. The above attack is effective when executed by the relevant party.

attack if

Attack slot:

$$\Pr\left[\begin{array}{c}\exists i\in[r]\colon\\ X_i^b-\mathbf{E}\left[Z_{i-1}^{1-b}\mid Z_i^b, X_{i-1}^b, \sum_{j\le i-1}(X_j^b-X_{j-1}^b)^2\right]\\ \geq 1/2\sqrt{r}\end{array}\right] \in \Omega(1) \quad (5)$$

The coin-flipping protocols of [4, 3, 2] show that the equation above does not hold in general. Nevertheless, we show that (for a suitable variant of) the above inequality does hold for the case $n \geq r$, and thus the "martingale" attack achieves the desired bias for this case. The case $n^k \geq r$ for $k \geq 2$ is significantly more complex, but follows the same principle. Details below.

3) An Efficient Attack for $n = r$: Let $(\mathsf{P}_1, \ldots, \mathsf{P}_n)$ be the parties of Π. For $b \in [n]$, let $Z_i^b \in \{0,1\}$ be the output (backup value) party P_b outputs if *all* other parties abort right after the i^{th} round, and for $\mathcal{S} \subseteq [n]$ let $Z_i^{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \cdot \sum_{s\in\mathcal{S}} Z_i^s$. For a subset $\mathcal{S} \subseteq [n]$, consider the sequence of augmented weak martingales $X_i^{\mathcal{S}} = \mathbf{E}\left[\mathsf{out} \mid Z_i^{\mathcal{S}}, X_{i-1}^{\mathcal{S}}, \sum_{j\in[i-1]}(X_j^{\mathcal{S}} - X_{j-1}^{\mathcal{S}})^2\right]$. As before, with constant probability $X_i^{\mathcal{S}} - X_{i-1}^{\mathcal{S}} \geq 1/\sqrt{r}$ for some $i \in [r]$. Hence, without loss of generality,

$$\text{Jump:}\quad \Pr\left[\exists i\in[r]\colon X_i^{\mathcal{S}} - Z_{i-1}^{\mathcal{S}} \geq 1/2\sqrt{r}\right] \in \Omega(1) \quad (6)$$

A crucial observation, and the reason why considering a number of parties that is *linear* in the round complexity is rewarding, is that, with high probability over the choice of $\mathcal{S}$ of size $n/2$, it holds that

$$\text{Similar backup values:}\quad \forall i\in[r]\colon\quad Z_i^{\mathcal{S}} = Z_i^{\overline{\mathcal{S}}} \pm 1/3\sqrt{r} \quad (7)$$

Namely, $Z_i^{\mathcal{S}}$ is a good estimation for $Z_i^{\overline{\mathcal{S}}}$, for all rounds $i \in [r]$ *simultaneously*.[6]

Indeed, since $\mathcal{S}$ is chosen at random, $Z_i^{\mathcal{S}}$ $(= \frac{1}{|\mathcal{S}|} \cdot \sum_{s\in\mathcal{S}} Z_i^s)$ is a $1/3\sqrt{r}$ approximation of $Z_i^{[n]}$ and thus of $Z_i^{\overline{\mathcal{S}}}$. Fix such a good set $\mathcal{S}$. The following martingale attack biases the output of a random party P_h not in $\mathcal{S}$ (i.e., $h \leftarrow \overline{\mathcal{S}}$) towards zero. In the i^{th} round, the attacker aborts all parties but P_h if $X_i^{\mathcal{S}} - Z_{i-1}^{\mathcal{S}} \geq 1/6\sqrt{r}$. Equations (6) and (7) implies that the above adversary biases the output of P_h towards zero by $\Omega(1/\sqrt{r})$.

[6]Actually, this requires $n = r \log r$, but we ignore such log factors in this informal discussion.

4) An Efficient Attack for $n^k \geq r$ via Differentially Private Sampling: We describe the attack for $n^2 \geq r$, and then briefly highlight the extension for $k \geq 3$.

A critical part of the above attack for $n = r$ (stated in (7)) is that for a random (and thus for some) subset $\mathcal{S} \subseteq [n]$ of size $n/2$, it holds that $Z_i^{\mathcal{S}}$ is at most $O(1/\sqrt{r})$-far from $Z_i^{\overline{\mathcal{S}}}$. This is not the case for $n^2 = r$, where we are only guaranteed that $Z_i^{\mathcal{S}}$ is at most $O(1/\sqrt{n}) = O(1/\sqrt[4]{r})$-far from $Z_i^{\overline{\mathcal{S}}}$, a too-rough approximation for our needs, since the error is larger than the potential gain of $O(1/\sqrt{r})$).

Our solution is to consider the *joint* backup values for *pairs* of parties. That is, the joint output of such a pair given that all other parties abort. Considering the pairs' backup values, however, raises a different problem. The adversary can no longer examine the values of a random large subset $\mathcal{P} \subsetneq \binom{[n]}{2}$ of backup values, as we described in the case $n = r$, since *each* party in $[n]$ (and, in particular, the honest party) takes part in $\Theta(1/n)$ fraction of $\mathcal{P}$, with high probability. Rather, we let the attacker examine the backup values of the pairs $\binom{\mathcal{S}}{2}$, for some subset $\mathcal{S} \subsetneq [n]$. If (the average of) these backup values are a good approximation for the backup value of pairs that contain the honest party, then the previous aborting strategy results in a bias of suitable magnitude. If not, then we can employ a different type of attack using differentially private sampling (Theorem 1.5). More details follow.

For a pair $p = (j_1, j_2) \in \binom{[n]}{2}$, let $Z_i^p \in \{0,1\}$ be the joint output (backup value) of the parties P_{j_1} and P_{j_2}, if *all* parties but them abort right after the i^{th} round. For $\mathcal{P} \subseteq \binom{n}{2}$, let $Z_i^{\mathcal{P}} = \frac{1}{|\mathcal{P}|} \cdot \sum_{p\in\mathcal{P}} Z_i^p$. Consider the sequence of augmented weak martingales $X_i^{\mathcal{S}} = \mathbf{E}\left[\mathsf{out} \mid Z_i^{\binom{\mathcal{S}}{2}}, X_{i-1}^{\mathcal{S}}, \sum_{j\in[i-1]}(X_j^{\mathcal{S}} - X_{j-1}^{\mathcal{S}})^2\right]$, for some subset $\mathcal{S} \subseteq [n]$. As before, with constant probability $X_{i+1}^{\mathcal{S}} - X_i^{\mathcal{S}} \geq 1/\sqrt{r}$ for some i. Assuming that

$$\text{Similar backup values:}\quad Z_i^{\binom{\mathcal{S}}{2}} = Z_i^{\mathcal{S}\times\overline{\mathcal{S}}} \pm o(1/\sqrt{r}) \quad (8)$$

for every i. Namely, the average backup values of pairs of parties seen by an attacker controlling all parties in $\mathcal{S}$, is very close to the average of the backup values of pairs containing one party in $\mathcal{S}$ and one party not in $\mathcal{S}$. Similarly to the case $n = r$, the above assumption enables the following martingale attack biasing the output of a random party P_h not in $\mathcal{S}$ (i.e., $h \leftarrow \overline{\mathcal{S}}$) towards zero by $\Omega(1/\sqrt{r})$. In the i^{th} round, if $X_i^{\mathcal{S}} - X_{i-1}^{\mathcal{S}} > 1/2\sqrt{r}$ the attacker aborts all parties but $(\mathsf{P}_h, \mathsf{P}_s)$, for a random $s \leftarrow \mathcal{S}$.

Unlike the case $n = r$, Equation (8) might be false (for any set $\mathcal{S}$). Fortunately, if this happens, we can mount a different attack, described below.

The differentially private sampling attack.: Assume for simplicity that

Non-similar backup values:

$$\Pr\left[\exists i \in [r]\colon Z_i^{\binom{\mathcal{S}}{2}} - Z_i^{\mathcal{S}\times\overline{\mathcal{S}}} > 1/\sqrt{r}\right] \in \Omega(1) \quad (9)$$

This calls for the following attack biasing the output of a random honest party P_h, for $h \leftarrow \mathcal{S}$, towards zero. For a pair-subset $\mathcal{P} \subseteq \binom{[n]}{2}$, let $\mathcal{P} \setminus h$ stand for all pairs in $\mathcal{P}$ that do not include h. In the i^{th} round, the attacker checks whether $G_i^{\setminus h} = Z_i^{\binom{\mathcal{S}}{2}\setminus h} - Z_i^{(\mathcal{S}\times\overline{\mathcal{S}})\setminus h} > 1/2\sqrt{r}$. If so, it aborts all parties but $(\mathsf{P}_h, \mathsf{P}_s)$ for a random $s \leftarrow \mathcal{S}$. The attack performs well if the backup values of the corrupted parties are a good approximation of the expected value of the honest party output. In particular, if for every h and i:

$$\text{Strong gap similarity:} \quad G_i^{\setminus h} = G_i^h \pm o(1/\sqrt{r}), \quad (10)$$

for $G_i^h = Z_i^{\{h\}\times(\mathcal{S}\setminus\{h\})} - Z_i^{\{h\}\times\overline{\mathcal{S}}}$. Unfortunately, Equation (10) might be false, without yielding any useful consequences. Rather, we can only assume the weaker guarantee that

Weak gap similarity:

$$G_i^{\setminus h} = G_i^h \pm o(1/\sqrt{n}) = G_i^h \pm o(1/\sqrt[4]{r}) \quad (11)$$

for every h and i. Indeed, if Equation (11) does not hold, then (w.l.o.g) for some party $h' \in \mathcal{S}$ it holds that $Z_i^{\{h\}\times\overline{\mathcal{S}}} \neq Z_i^{\{h'\}\times\overline{\mathcal{S}}} \pm \Theta(1/\sqrt[4]{r})$. That is, when restricting our attention to the $n-1$ pairs containing h', the gap is $\Theta(1/\sqrt{n})$. Such gap yields that an attack in the spirit of the one used for the $n = r$ case induces a large bias on an honest party chosen randomly from $\overline{\mathcal{S}}$.

The guarantee of Equation (11) does not suffice for the simple attack described below Equation (9) to go through. Roughly, the reason is that the approximation error (i.e., $o(1/\sqrt[4]{r})$) is larger than the expected gain of $\Omega(1/\sqrt{r})$. Indeed, we find ourselves in the setting of the oblivious sampling game considered in Section 1-A2, letting $\gamma = 1/\sqrt{r}$, $s_i^h = G_i^h$ and $s_i = G_i = \mathbf{E}_{h\leftarrow\mathcal{H}}\left[s_i^h\right]$, and $\sigma = \left|s_i - s_i^h\right| \in o(1/\sqrt[4]{r}) > 1/\sqrt{r}$. As explained in Section 1-A2, a threshold deterministic attack for this sampling game, i.e. seeing the values of $\left\{s_i^h\right\}_{i\in[r]}$ one by one, until one decides to abort, might achieve no reward for random h, and neither for any fixed h. In particular, it might hold that $s_i^h = 0$ for i being the aborting round.

Fortunately, since we are in the setting of the oblivious sampling game, Theorem 1.5 yields that randomized aborting online strategy that adds noise to its halting decision, in every round, performs significantly better. Specifically, the strategy that adds the right Laplace noise to $s_i^{\setminus h} = \mathbf{E}_{h'\leftarrow\mathcal{H}\setminus h}\left[s_i^{h'}\right]$, and aborts if the result is greater than $\gamma/2$, achieves expected revenue $\gamma/2 - \sigma^2 = 1/2\sqrt{r} - o(1/\sqrt[4]{r})^2 > 1/4\sqrt{r}$. The above holds for a random h, and thus for some fixed h as well. It follows that the induced attacker on the protocol, controlling all parties by P_h, and applying the same randomized aborting strategy, obtains $Z_i^{\{h\}\times(\mathcal{S}\setminus\{h\})} - Z_i^{\{h\}\times\overline{\mathcal{S}}} \geq 1/4\sqrt{r}$ where i denotes the aborting round. This translates, using the same means as in the deterministic threshold attack for choosing the non aborting pair, into a $1/4\sqrt{r}$-bias of P_h's output.

Intuitively, the point of using the differentially private sampling mechanism is to avoid identifying the choice of honest party. That is, the effect of the backup values of pairs containing any single party on the adversary's decision to trigger the attack is diminished (Equation (11) keeps the sensitivity of this decision small).

The case $n^k \geq r$ for $k \geq 3$.: To begin, assume that $k = 3$ (i.e., $n^3 \geq r$). For such value of n, it holds that $1/\sqrt{n} = 1/\sqrt[6]{r} \gg 1/\sqrt[4]{r}$. Thus, the promise $G_i = G_i^h \pm o(1/\sqrt{n})$ does not suffice for the differentially private based attack to go through. Rather, we need to assume that $G_i = G_i^h \pm o(1/\sqrt[4]{r}) = G_i^h \pm o(1/n^{3/4})$. We show that if the latter does not hold, the attacker can fix a party and never abort it (i.e., we restrict the subset of all backup values to those containing this party) we are essentially in the setting of $n^2 \geq r$. Namely, either we have differentially private based attack, or we have a martingale attack (both with respect to the above fixing of a never aborting party).

For larger values of k, we iterate the above, fixing non-aborting parties one after the other, until one of the differentially private based attacks or the martingale attack go through.

5) Computing Doob-like Weak Martingales: In Section 1-A1, we claimed that if the support size of the Z_i's is small, then the sum-of-squares-augmented weak martingales $X_1, \ldots, X_r$ defined by the rounded Doob-like sequence $X_i = \mathsf{rnd}(\mathbf{E}\left[f(Z) \mid Z_i, X_{i-1}, \sum_{j\in[i-1]}(X_j - X_{j-1})^2\right])$ can be efficiently computable, where rnd is a small support rounding function. We use this guarantee above to argue that our attack is efficient. While this claim trivially holds when considering non-uniform algorithms, the argument for *uniform* algorithms is more subtle, and

since we believe it to be of independent interest, we highlight it below.

For simplicity, we focus on the weak martingales defined by the Doob-like sequence $X_i = \mathsf{rnd}(\mathbf{E}[f(Z) \mid Z_i, X_{i-1}])$. Consider the mappings $\chi_1, \ldots, \chi_r$ inductively defined by $\chi_i(z) = \mathsf{rnd}(\mathbf{E}[f(Z) \mid Z_i = z_i, X_{i-1} = \chi_{i-1}(z)])$. It is easy to verify that $X_i = \chi_i(Z_{\leq r})$, and since each of these mappings has a small description, the sequence $X_0, \ldots, X_r$ can be computed from $Z_1, \ldots, Z_r$ by a small circuit holding these mappings. Arguing that the above can be performed by an efficient (uniform) *algorithm*, things get slightly more involved. While we can estimate well the mapping $\chi_1(z) = \mathsf{rnd}(\mathbf{E}[f(Z) \mid Z_1 = z_1])$ via sampling, even a small unavoidable error in the estimation might cause a larger error in the estimation of $\chi_2 = \mathsf{rnd}(\mathbf{E}[f(Z) \mid Z_2 = z_2, X_{i-1} = \chi_1(z)])$. This is since the dependency on χ_1 is in the conditioning, and thus estimating χ_2 using an estimate of χ_1 amplifies the error. This might lead to very large errors when trying to use the estimated mapping for calculating X_i's of large indices.

So rather, we consider the efficiently computable sequence $\widehat{X} = (\widehat{X}_1, \ldots, \widehat{X}_r)$ defined by $\widehat{X}_i = \mu(Z_{\leq i})$, for $\mu_1, \ldots, \mu_r$ being an *estimation* for $\chi_1, \ldots, \chi_r$ done via sampling.[7] Since $\widehat{X}$ is defined with respect to the approximated mappings, it is a weak martingale, even if the approximated mappings *wrongly approximate* the real ones. The reason is that the quality of $\widehat{X}_i$ as a "Doob-like sequence" — i.e. how well it approximates $\mathbf{E}\left[f(Z) \mid Z_i, \widehat{X}_{i-1}\right]$ — is not affected by the quality of $\mu_1, \ldots, \mu_{i-1}$, and thus errors do not accumulate. Taking the same approach for the sum-of-squares-augmented weak martingales, our construction yields that with high probability over the choice of the estimated mappings $\mu_1, \ldots, \mu_r$, the sequence $\widehat{X}$ satisfies all the properties required by Theorem 1.3, and thus we can invoke our attack using this sequence.

C. Related Work

1) Coin Flipping: A coin-flipping protocol is δ-fair, if no efficient attacker (controlling any number of parties) can bias the output (bit) of the honest parties by more than δ.

Upper bounds.: Blum [7] presented a two-party two-round coin-flipping protocol with bias $1/4$. Awerbuch et al. [5] presented an n-party r-round protocol with bias $O(n/\sqrt{r})$ (the two-party case appears also in Cleve [1]). This was improved to (almost) $O(1/\sqrt{r})$ in [10, 11], for the case where the fraction of honest parties is constant. Moran, Naor, and Segev [12] resolved the two-party case, presenting a two-party r-round coin-flipping protocol with bias $O(1/r)$. Haitner and Tsfadia [13] resolved the three-party case up to poly logarithmic factor, presenting a three-party coin-flipping protocol with bias $O(\mathrm{polylog}(r)/r)$. Buchbinder et al. [4] constructed an n-party r-round coin-flipping protocol with bias $\widetilde{O}(n^3 2^n / r^{\frac{1}{2} + \frac{1}{2^{n-1}-2}})$. In particular, their four-party coin-flipping protocol the bias is $\widetilde{O}(1/r^{2/3})$, and for $n = \log\log r$ their protocol has bias smaller than [5].

For the case where less than $2/3$ of the parties are corrupt, Beimel et al. [14] have constructed an n-party r-round coin-flipping protocol with bias $2^{2^k}/r$, tolerating up to $t = (n+k)/2$ corrupt parties. Alon and Omri [15] constructed an n-party r-round coin-flipping protocol with bias $\widetilde{O}(2^{2^n}/r)$, tolerating up to t corrupted parties, for constant n and $t < 3n/4$.

Lower bounds.: Cleve [1] proved that for every r-round two-party coin-flipping protocol there exists is an efficient adversary that can bias the output by $\Omega(1/r)$. Cleve and Impagliazzo [6] proved that for every r-round two-party coin-flipping protocol there exists an inefficient fail-stop adversary that biases the output by $\Omega(1/\sqrt{r})$. They also showed that a similar attack exists also if the parties have access to an ideal commitment scheme. All above bounds extend to multi-party protocol (with no honest majority) via a simple reduction.

A different line of work examines the minimal assumptions required to achieve an $o(1/\sqrt{r})$-bias two-party coin-flipping protocols. Dachman-Soled et al. [16] have shown that any fully black-box construction of $O(1/r)$-bias two-party protocols based on one-way functions with r-bit input and output needs $\Omega(r/\log r)$ rounds. Dachman-Soled et al. [17] have shown that there is no fully black-box and function *oblivious* construction of $O(1/r)$-bias two-party protocols from one-way functions (a protocol is function oblivious if the outcome of the protocol is independent of the choice of the one-way function used in the protocol). Very recently, Haitner et al. [18] have used an attack in the spirit of the one used in this paper, together with the dichotomy result of Haitner et al. [19], to prove that key-agreement is a necessary assumption for *two-party* r-round coin-flipping protocol of bias $o(1/\sqrt{r})$, as long as r is independent of the security parameter.

A different type of lower bound was given by Cohen et al. [20]. They focused on the communication model

[7] The mapping $\mu_1, \ldots, \mu_r$ are constructed iteratively. After constructing $\mu_1, \ldots, \mu_{i-1}$, the value of $\mu_i(z)$ is set by approximating via sampling (a rounding of) $\mathbf{E}[f(Z) \mid Z_i = z_i, \mu_{i-1}(Z) = \mu_{i-1}(z)]$.

required for fully secure computation, and in particular showed that in the setting where broadcast is impossible (e.g., peer-to-peer network with $1/3$ fraction of dishonest parties), there exists no many-party coin-flipping protocol with non-trivial bias (i.e., noticeably smaller then $1/2$).

2) $1/p$-Secure Protocols: Cleve [1] result implies that for many functions fully-secure computation without an honest majority is not possible. Gordon and Katz [21] suggested the notion of $1/p$-secure computation to bypass this impossibility result. Very informally, a protocol is $1/p$-secure if every poly-time adversary can harm the protocol with probability at most $1/p$ (e.g., with probability $1/p$ the adversary can learn the inputs of honest parties, get the output and prevent the honest parties from getting the output, or bias the output). Gordon and Katz [21] constructed for every polynomial $p(\kappa)$ (where κ is the security parameter) an efficient two-party $1/p(\kappa)$-secure protocol for computing a function f, provided that the size of the domain of at least one party in f or the size of the range of f is bounded by a polynomial. Beimel et al. [22] generalized this result to multi-party protocols when the number of parties is constant – for every function f with $O(1)$ inputs such that the domain of each party (or the size of the range of f) is bounded by a polynomial and for every polynomial $p(\kappa)$, they presented an efficient $1/p(\kappa)$-secure protocol for computing the function.

Gordon and Katz [21] and Beimel et al. [22] also provided impossibility results explaining why their protocols require bounding the size of the domain or range of the functions. Specifically, Gordon and Katz [21] described a two-party function whose size of domain of each party and size of range is $\kappa^{\omega(1)}$ such that this function cannot be computed by any poly-round protocol achieving $1/3$-security. Beimel et al. [22] used this result to construct a function $f\colon \{0,1\}^{\omega(\log n)} \to \kappa^{\omega(1)}$ (i.e., a function with $\omega(\log n)$ parties where the domain of each party is Boolean) such that this function cannot be computed by any poly-round protocol achieving $1/3$-security. They also showed the same impossibility result for a function with $\omega(1)$ parties where the domain of each party is bounded by a polynomial is the security parameter. We emphasize that these impossibility results do not apply to coin-flipping protocols, where the parties do not have inputs.

3) Complete Fairness Without Honest Majority: Cleve [1] result was interpreted as saying that non-trivial functions cannot be computed with complete fairness without an honest majority. In a surprising result, Gordon et al. [23] have shown that the millionaire problem with a polynomial size domain and other interesting functions can be computed with complete fairness in the two-party setting. The two-party functions that can be computed with complete fairness were further studied in [24, 25, 26, 27]; in particular, Asharov et al. [27] characterized the Boolean functions that can be computed with complete fairness. Gordon and Katz [28] have studied complete fairness in the multi-party case and constructed completely-fair protocols for non-trivial functions in this setting.

4) Differential Privacy: Differential privacy, introduced by Dwork et al. [9], provides a provable guarantee of privacy for data of individuals. Assume there is a database containing private information of individuals and there is an algorithm computing some function of the database. We say that such randomized algorithm is differentially private if changing the data of one individual has small affect on the output of the algorithm. For example, if, for a database D, a function $f(D)$ returns a numerical value in $[0,1]$, then an algorithm returning $f(D) + \textit{noise}$, where *noise* is distributed according to the Laplace distribution (with suitable parameters), is a differentially private algorithm. Since the introduction of differential privacy in 2006, many algorithms satisfying differential privacy were introduced, see, e.g., Dwork and Roth [29]. In this work we use differential privacy (i.e., Laplace noise) not for protecting privacy, but rather to provide oblivious sampling. This is similar in spirit to the usage of differential privacy, by Dwork et al. [30], to enable adaptive queries to a database.

Open Questions

Our lower bound is only applicable if the number of parties n is greater than $\log(r)$, and it is very close to the $n/\sqrt{r}$-upper bound (protocol) of [5], if the number of parties is $n = r^{\varepsilon}$ for "small" $\varepsilon > 0$. For $n < \operatorname{loglog} r$ parties, the upper bounds of [2, 3, 4] tell us that no attack achieving $\widetilde{O}(1/\sqrt{r})$-bias exists. For $\operatorname{loglog}(r) < n \leq \log(r)$ parties, we know of no such limitation, yet our attack is either innaplicable, or it yields a bias that is smaller than $1/r$. Thus, for the latter choice of parameters, the only known bound remains the $1/r$-lower bound of [1], which is far from meeting the upper bound of [5].

Paper Organization

Basic definitions and notation used throughout the paper, are given in Section 2. Our result for augmented week martingales is stated in Section 3, and the oblivious sampling result is given in Section 4. The main theorem is given in Section 5. Detailed proofs can be found in the full version of our paper [31].

2. PRELIMINARIES

A. Notation

We use calligraphic letters to denote sets, uppercase for random variables and functions, lowercase for values, and boldface for vectors. All logarithms considered here are in base two. For a vector $\mathbf{v}$, we denote its i^{th} entry by $\mathbf{v}_i$ or $\mathbf{v}[i]$. For $a \in \mathbb{R}$ and $b \geq 0$, let $a \pm b$ stand for the interval $[a - b, a + b]$. Given sets $\mathcal{S}_1, \ldots, \mathcal{S}_k$ and k-input function f, let $f(\mathcal{S}_1, \ldots, \mathcal{S}_k) := \{f(x_1, \ldots, x_j) \colon x_i \in \mathcal{S}_i\}$, e.g., $f(1 \pm 0.1) = \{f(x) \colon x \in [.9, 1.1]\}$. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$ and $(n) := \{0, \ldots, n\}$. Given a vector $\mathbf{v} \in \{0,1\}^*$, let $w(\mathbf{v}) := \sum_{i \in [|\mathbf{v}|]} \mathbf{v}_i$. For $x, \delta \in [0,1]$ let $\mathsf{rnd}_\delta(x) = k\delta$, for $k \in \mathbb{Z}$ being the largest number with $k\delta \leq x$. For a function $f \colon \mathcal{A} \mapsto \mathcal{B}$, let $\mathrm{Im}(f) = \{f(a) \colon a \in \mathcal{A}\}$.

Let poly denote the set of all polynomials, let PPT stand for probabilistic polynomial time, let PPTM denote a PPT algorithm (Turing machine) and let $\text{PPTM}^{\mathsf{NU}}$ stands for a *non-uniform* PPTM. A function $\nu \colon \mathbb{N} \to [0,1]$ is *negligible*, denoted $\nu(n) = \mathrm{neg}(n)$, if $\nu(n) < 1/p(n)$ for every $p \in \mathrm{poly}$ and large enough n.

B. Coin-Flipping Protocols

Since the focus of this paper is showing the non-existence of coin-flipping protocols with small bias, we will only focus on the correctness and bias of such protocols. See [3] for a complete definition of such protocols.

Definition 2.1 (correct coin-flipping protocols)**.** *A multi-party protocol is a* correct coin-flipping protocol*, if*

- *When interacting with an efficient adversary controlling a subset of the parties, the honest parties* always *output the same bit, and*
- *The common output in a random* honest *execution of the protocol is a uniform bit.*

Definition 2.2 (Biassing coin-flipping protocols)**.** *An adversary* A *controlling a strict subsets of the parties of a correct coin-flipping protocol* biases *its output by $\delta \in [1/2, 1]$, if when interacting with the parties controlled by* A*, the remaining honest parties output some a priory fixed bit $b \in \{0,1\}$ with probability $\frac{1}{2} + \delta$.*

Such an adversary is called fail stop*, if the parties in its control honestly follow the prescribed protocol, but might abort prematurely. The adversary is a rushing adversary, that is, in each round, first the honest parties send their messages, then the adversary might instruct some of the parties to abort (that is, send a special "abort" message to all other parties), and finally, all corrupt parties that have not aborted send their messages.*

C. Basic Probability Facts

Given a distribution D, we write $x \leftarrow D$ to indicate that x is selected according to D. Similarly, given a random variable X, we write $x \leftarrow X$ to indicate that x is selected according to X. Given a finite set $\mathcal{S}$, we let $s \leftarrow \mathcal{S}$ denote that s is selected according to the uniform distribution on $\mathcal{S}$. Let D be a distribution over a finite set $\mathcal{U}$, for $u \in \mathcal{U}$, denote $D(u) = \Pr_{X \leftarrow D}[X = u]$ and for $\mathcal{S} \subseteq \mathcal{U}$ denote $D(\mathcal{S}) = \Pr_{X \leftarrow D}[X \in \mathcal{S}]$. Let the support of D, denoted $\mathrm{Supp}(D)$, be defined as $\{u \in \mathcal{U} : D(u) > 0\}$. The *statistical distance* between two distributions P and Q over a finite set $\mathcal{U}$, denoted as $\mathrm{SD}(P, Q)$, is defined as $\max_{\mathcal{S} \subseteq \mathcal{U}} |P(\mathcal{S}) - Q(\mathcal{S})| = \frac{1}{2} \sum_{u \in \mathcal{U}} |P(u) - Q(u)|$.

1) The Laplace Distribution:

Definition 2.3. *The Laplace distribution with parameter $\lambda \in \mathbb{R}^+$, denoted* $\mathsf{Lap}(\lambda)$*, is defined by the density function $f(x) = \exp(-|x|/\lambda)/2\lambda$.*

D. Martingales

In this section we define weaker variants of martingales.

Definition 2.4 (δ-martingales)**.** *Let $X_0, \ldots, X_r$ be a sequence of random variables. We say that the sequence is a δ-*strong *martingale sequence if $\mathbf{E}[X_{i+1} \mid X_{\leq i} = x_{\leq i}] \in x_i \pm \delta$ for every $i \in [r-1]$. We say that the sequence is a δ-*weak *martingale sequence if $\mathbf{E}[X_{i+1} \mid X_i = x_i] \in x_i \pm \delta$ for every $i \in [r-1]$. If $\delta = 0$, the above are just called* strong *and* weak *martingale sequence respectively.*

In plain terms, a sequence is a strong martingale if the expectation of the next point conditioned on the entire history is exactly the last observed point. Analogously, a sequence is a weak martingale if the expectation of the next point conditioned on the previous point is equal to the previous point.

Definition 2.5 (SoS-augmented δ-weak martingale)**.** *Let $X_0, \ldots, X_r$ be a sequence of random variables. We say that the sequence is a SoS-augmented δ-*weak *martingale sequence if $\mathbf{E}\left[X_{i+1} \mid X_i = x_i, \sum_{j<i}(X_{j+1} - X_j)^2 = \sigma\right] \in x_i \pm \delta$ for every $i \in [r-1]$.*

In a sense, a sequence is a SoS-augmented weak martingale if it satisfies the weak martingale property and it is "distance-oblivious", i.e. the expectation is unaffected by conditioning on the quantity $\sum_{j<i}(X_{j+1} - X_j)^2$,

which captures the distance the sequence has traveled thus far.

Definition 2.6 (Associated difference sequence)**.** *Let $X_0, \ldots, X_r$ be a an arbitrary sequence and define $Y_i = X_i - X_{i-1}$, for every $i \in [r]$. The sequence $Y_1 \ldots Y_r$ is referred to as the difference sequence associated with $X_0, \ldots, X_r$.*

By Definitions 2.4 and 2.6, it follows immediately that a sequence $Y_1 \ldots Y_r$ is a δ-strong martingale difference if and only if $\mathbf{E}[Y_i \mid Y_1, \ldots, Y_{i-1}] \in \pm\delta$, and that a sequence $Y_1 \ldots Y_r$ is a δ-weak martingale difference if and only if $\mathbf{E}\left[Y_i \mid \sum_{\ell<i} Y_\ell\right] \in \pm\delta$. By Definitions 2.5 and 2.6, a sequence $Y_1 \ldots Y_r$ is a SoS-augmented δ-weak martingale difference if and only if $\mathbf{E}\left[Y_i \mid \sum_{\ell<i} Y_\ell, \sum_{\ell<i} Y_\ell^2\right] \in \pm\delta$.

3. Augmented Weak Martingales have Large Gaps

In this section, we prove a result about sequences that satisfy a weaker version of the "martingale property". Namely, we show that for any sequence satisfying the SoS-augmented δ-weak martingale property, if $X_0 = 1/2$ and $X_r \in \{0,1\}$, then the quantity $\sum_{i=1}^r (X_i - X_{i-1})^2$ is greater than $1/16$ with constant probability. As a corollary, we obtain a generalization of the result of Cleve and Impagliazzo [6], who showed that (strong) martingales have large gap between consecutive points. We emphasize that our results extend immediately to the usual notion of (strong) martingale sequences. The reader is referred to Section 1-A1 for an informal discussion and motivation for the present section.

Recall (cf., Section 2-D) that a sequence $X_0, \ldots, X_r$ is a δ-weak martingale, if $\mathbf{E}[X_{i+1} \mid X_i = x_i] \in x_i \pm \delta$ for every $i \in [r-1]$ and $x_i \in \mathrm{supp}(X_i)$. Further recall that the difference sequence associated with $X_0, \ldots, X_r$ is the sequence $Y_1, \ldots, Y_r$ defined by $Y_i = X_i - X_{i-1}$, for every $i \in [r]$. We begin by extending to weak martingales a result of DasGupta [32] for strong martingales. We will use this result in the proof of our main theorem.

Lemma 3.1. *Let $X_0 \ldots X_r$ be a δ-weak martingale and let $Y_i = X_i - X_{i-1}$. If $X_i \in [0,1]$ for every $i \in [r]$, then $\mathbf{E}\left[X_r^2 - X_0^2\right] \in \mathbf{E}\left[\sum_{i\in[r]} Y_i^2\right] \pm 2r\delta$.*

Recall that a sequence $X_0 \ldots X_r$ is a SoS-augmented δ-weak martingale if $\mathbf{E}\left[X_{i+1} \mid X_i = x_i, \sum_{\ell\le i}(X_i - X_{i-1})^2 = \sigma\right] \in x_i \pm \delta$ for every $i \in [r-1]$, $x_i \in \mathrm{supp}(X_i)$ and $\sigma \in \mathrm{supp}(\sum_{\ell\le i}(X_i - X_{i-1})^2)$. Following is the main result of this section.

Theorem 3.2. *For $\delta < 1/100r$, let $X_0, \ldots, X_r$ be a SoS-augmented δ-weak martingale sequence such that $X_i \in [0,1]$ for every $i \in [r]$. Assuming $X_0 = 1/2$ and $\Pr[X_r \in \{0,1\}] = 1$, then $\Pr\left[\sum_{i\in[r]}(X_i - X_{i-1})^2 \ge 1/16\right] \ge 1/20$.*

We provide a sketch of the proof. Assume without loss of generality that $\Pr[X_r = 1] \ge 1/2$ (otherwise apply the argument to the sequence $X_0', \ldots, X_r'$ defined by $X_i' = 1 - X_i$, for every $i \in [r]$). Notice that if $\Pr\left[\sum_{i=1}^r (X_i - X_{i-1})^2 \ge \frac{1}{16}\right] = 0$ then $\mathbf{E}\left[\sum_{i=1}^r (X_i - X_{i-1})^2\right] \le \frac{1}{16}$, in contradiction with Lemma 3.1 which states that $\mathbf{E}\left[\sum_{i=1}^r (X_i - X_{i-1})^2\right] = \mathbf{E}\left[X_r^2 - X_0^2\right] \ge \frac{1}{4}$. We argue that a similar contradiction can be derived if $\Pr\left[\sum_{i=1}^r (X_i - X_{i-1})^2 \ge \frac{1}{16}\right] < 1/20$. Unfortunately, we cannot apply the same inequality as before because we have no control over the quantity $\sum_{i=1}^r (X_i - X_{i-1})^2$ when it is greater than $1/16$ (a crude upper bound is r which is utterly unhelpful). Our solution is to construct a weak martingale sequence $U_0, \ldots, U_r$ which is "coupled" with the X-sequence in the following way: U_i is equal to X_i as long as $\sum_{\ell=1}^{i-1}(X_\ell - X_{\ell-1})^2 \le \frac{1}{16}$, and $U_i = U_{i-1}$ otherwise. Then, we argue that $\mathbf{E}\left[U_r^2 - U_0^2\right] \ge \frac{1}{4} - \Pr\left[\sum_{i=1}^r (X_i - X_{i-1})^2 \ge \frac{1}{16}\right]$ by observing that $\Pr\left[\sum_{i=1}^r (X_i - X_{i-1})^2 \ge \frac{1}{16}\right]$ roughly corresponds to the probability that the two sequences diverge. We then upper bound the latter by applying Lemma 3.1 to the sequence $U_0, \ldots, U_r$ which we have a much better grasp on, since, by construction, $\sum_{i=1}^r (U_i - U_{i-1})^2$ can never exceed $1/16$ by much.

4. Oblivious Sampling via Differential Privacy

Consider the following r-round game in which your goal is to maximize the revenue of a random "party" $H \leftarrow \mathcal{H}$. In the beginning, a party H is chosen with uniform distribution from $\mathcal{H}$ (where $\mathcal{H}$ is a finite set of parties). In each round, values $\left\{s_i^h \in [0,1]\right\}_{h\in\mathcal{H}}$ are assigned to the parties of $\mathcal{H}$, but only the values $\{s_h\}_{h\in\mathcal{H}\setminus\{H\}}$ of the other parties are published. You can decide to *abort*, and then party H is rewarded by s_i^H, or to continue to the next round. If an abort never occurs, party H is rewarded by s_r^H (last round value). You have the *similarity* guarantee that $\left|s_i^h - s_i\right| \le \sigma$ for every $h \in \mathcal{H}$, letting $s_i = \mathbf{E}_{h\leftarrow\mathcal{H}}\left[s_i^h\right]$. You are also guarantee that $\max_i\{s_i\} \ge \gamma$.

In this section we analyze the following "differentially private based" approach for this task, which is described by the following experiment (the basic game described above is captured by the experiment for $p = 1/n$).

Experiment 4.1 (LapExp: Oblivious sampling experiment)**.**
Parameters: $\mathcal{H} = [n]$, $\mathcal{S} = \left\{s_i^h \in [-1,1]\right\}_{i\in[r],h\in\mathcal{H}}$, $p \in [0,1/2]$, $\gamma \in [0,1]$ *and* $\lambda \in \mathbb{R}^+$.
Notation: Let $s_i = \frac{1}{n}\sum_{h\in\mathcal{H}} s_i^h$ *and for* $h \in \mathcal{H}$ *let* $s_i^{\setminus h} = \frac{1}{1-p}(s_i - p \cdot s_i^h)$.
Description:

1) *Sample* $h \leftarrow \mathcal{H}$.
2) *For* $i = 1,\ldots,r-1$*:*
   a) *Sample* $\nu_i \leftarrow \mathsf{Lap}(\lambda)$.
   b) *If* $s_i^{\setminus h} + \nu_i \geq \gamma$*, output* s_i^h *and halt.*
3) *Output* s_r^h.

Let $\mathsf{LapExp}(\mathcal{H},\mathcal{S},\gamma,\lambda)$ denote the above experiment with parameters $\mathcal{H}$, $\mathcal{S}$, γ and λ. Theorem 4.2 analyzes the expected value of the output of $\mathsf{LapExp}(\mathcal{H},\mathcal{S},\gamma,\lambda)$.

Theorem 4.2 (Quality of the oblivious sampling experiment)**.** *Let* $\mathcal{H}$*,* $\mathcal{S}$*,* γ*,* λ *and* p *be as in Experiment 4.1, with* $s_r^h = s_r$ *for every* $h \in \mathcal{H}$*. Let* $\sigma^h = \max_i\left\{\left|s_i - s_i^h\right|\right\}$*, let* $\mathcal{S}\mathsf{imilar} = \left\{h \in \mathcal{H}\colon \sigma^h \leq \lambda\cdot(1-p)/p\right\}$ *and* $\mathcal{N}\mathsf{on}\mathcal{S}\mathsf{imilar} = \mathcal{H} \setminus \mathcal{S}\mathsf{imilar}$.

Let H *be the value of* h *and* J *be the halting round (set to* r *if Experiment 4.1 does not halt in step (2b)) in a random execution of* $\mathsf{LapExp}(\mathcal{H},\mathcal{S},\gamma,\lambda)$*. Then* $\mathbf{E}\left[s_J^H\right] \geq \mathbf{E}\left[v_H\right] - r\cdot e^{-\gamma/2\lambda}$*, where*

$$v_h = \begin{cases} q_h^{\perp}\cdot\left(\frac{\gamma}{2} - \frac{40(\sigma^h)^2}{\lambda}\cdot\frac{p}{1-p}\right) & h \in \mathcal{S}\mathsf{imilar} \\ -4\sigma^h & h \in \mathcal{N}\mathsf{on}\mathcal{S}\mathsf{imilar} \end{cases},$$

and $q_h^{\perp} = \Pr\left[J \neq r \mid H = h\right]$*. If* $s_i \geq \gamma$ *for some* $i \in [r-1]$*, then* $q_h^{\perp} \geq 1/6$*, for every* $h \in \mathcal{S}\mathsf{imilar}$.

5. Biasing Coin-Flipping Protocols

Theorem 5.1 (Main theorem)**.** *There exists a fail-stop adversary* A *such that the following holds. Let* Π *be a correct* n*-party* r*-round coin-flipping protocol, and let* $k \in \mathbb{N}$ *be the smallest integer such that* $\binom{n}{k} \geq r\log(r)^{2k}$*. Then, there exists a party* P *in* Π *such that* A^{Π} *controlling all parties but* P *biases the output of* P *by* $\Omega(1/\sqrt{r}\log(r)^k)$*. The running time of* A^{Π} *is polynomial in the running time of* Π *and* n^k*, and it uses oracle only access to* Π*'s next-message function.*

Acknowledgment

Research of A.B. and E.O. was supported by ISF grant 152/17. Research of I.H. and N.M. was supported by ERC starting grant 638121.

References

[1] R. Cleve, "Limits on the security of coin flips when half the processors are faulty," in *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1986, pp. 364–369.

[2] T. Moran, M. Naor, and G. Segev, "An optimally fair coin toss," *Journal of Cryptology*, vol. 29, no. 3, pp. 491–513, 2016.

[3] I. Haitner and E. Tsfadia, "An almost-optimally fair three-party coin-flipping protocol," vol. 46, no. 2, pp. 479–542, 2017.

[4] N. Buchbinder, I. Haitner, N. Levi, and E. Tsfadia, "Fair coin flipping: Tighter analysis and the many-party case," in *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, pp. 2580–2600.

[5] B. Awerbuch, M. Blum, B. Chor, S. Goldwasser, and S. Micali, "How to implement Bracha's $O(\log n)$ byzantine agreement algorithm," 1985, unpublished manuscript.

[6] R. Cleve and R. Impagliazzo, "Martingales, collective coin flipping and discrete control processes (extended abstract)," http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.1797, 1993.

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

[8] P. I. Nelson, "A class of orthogonal series related to martingales," *Annals of Mathematical Statistics*, vol. 41, pp. 1684–1694, 1970.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Journal of Privacy and Confidentiality*, vol. 7, no. 3, p. 2, 2016.

[10] A. Beimel, E. Omri, and I. Orlov, "Protocols for multiparty coin toss with dishonest majority," in *Advances in Cryptology – CRYPTO 2010*, 2010, pp. 538–557.

[11] R. Cohen, I. Haitner, E. Omri, and L. Rotem, "From fairness to full security in multiparty computation," 2018, manuscript.

[12] T. Moran, M. Naor, and G. Segev, "An optimally fair coin toss," in *Proceedings of the 6th Theory of Cryptography Conference, TCC 2009*, 2009, pp. 1–18.

[13] I. Haitner and E. Tsfadia, "An almost-optimally fair three-party coin-flipping protocol," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, 2014, pp. 817–836.

[14] A. Beimel, E. Omri, and I. Orlov, "Protocols for multiparty coin toss with a dishonest majority," *Journal of Cryptology*, vol. 28, no. 3, pp. 551–

600, 2015.

[15] B. Alon and E. Omri, "Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious," in *Proceedings of the 14th Theory of Cryptography Conference, TCC 2016-B, part I*, 2016, pp. 307–335.

[16] D. Dachman-Soled, Y. Lindell, M. Mahmoody, and T. Malkin, "On the black-box complexity of optimally-fair coin tossing," in *Proceedings of the 8th Theory of Cryptography Conference, TCC 2011*, vol. 6597, 2011, pp. 450–467.

[17] D. Dachman-Soled, M. Mahmoody, and T. Malkin, "Can optimally-fair coin tossing be based on one-way functions?" in *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*, ser. Lecture Notes in Computer Science, Y. Lindell, Ed., vol. 8349. Springer, 2014, pp. 217–239.

[18] I. Haitner, N. Makriyannis, and E. Omri, "On the complexity of fair coin flipping," Electronic Colloquium on Computational Complexity, Tech. Rep. TR18-84, 2018.

[19] I. Haitner, K. Nissim, E. Omri, R. Shaltiel, and J. Silbak, "Computational two-party correlation," Electronic Colloquium on Computational Complexity, Tech. Rep. TR18-071, 2018.

[20] R. Cohen, I. Haitner, E. Omri, and L. Rotem, "Characterization of secure multiparty computation without broadcast," in *Proceedings of the 14th Theory of Cryptography Conference, TCC 2016-B, part I*, 2016, pp. 596–616.

[21] D. Gordon and J. Katz, "Partial fairness in secure two-party computation," in *Advances in Cryptology – EUROCRYPT 2010*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed., vol. 6110. Springer, 2010, pp. 157–176.

[22] A. Beimel, Y. Lindell, E. Omri, and I. Orlov, "1/p-secure multiparty computation without honest majority and the best of both worlds," in *Advances in Cryptology – CRYPTO 2011*, ser. Lecture Notes in Computer Science, P. Rogaway, Ed., vol. 6841. Springer, 2011, pp. 277–296.

[23] S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell, "Complete fairness in secure two-party computation," *J. ACM*, vol. 58, no. 6, pp. 24:1–24:37, 2011.

[24] G. Asharov, Y. Lindell, and T. Rabin, "A full characterization of functions that imply fair coin tossing and ramifications to fairness," in *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013*, ser. Lecture Notes in Computer Science, A. Sahai, Ed., vol. 7785. Springer, 2013, pp. 243–262.

[25] G. Asharov, "Towards characterizing complete fairness in secure two-party computation," in *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*, ser. Lecture Notes in Computer Science, Y. Lindell, Ed., vol. 8349. Springer, 2014, pp. 291–316.

[26] N. Makriyannis, "On the classification of finite boolean functions up to fairness," in *Security and Cryptography for Networks - 9th International Conference, SCN 2014, 2014.*, ser. Lecture Notes in Computer Science, M. Abdalla and R. D. Prisco, Eds., vol. 8642. Springer, 2014, pp. 135–154.

[27] G. Asharov, A. Beimel, N. Makriyannis, and E. Omri, "Complete characterization of fairness in secure two-party computation of boolean functions," in *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, 2015, Proceedings, Part I*, ser. Lecture Notes in Computer Science, Y. Dodis and J. B. Nielsen, Eds., vol. 9014. Springer, 2015, pp. 199–228.

[28] D. Gordon and J. Katz, "Complete fairness in multi-party computation without an honest majority," in *Proceedings of the 6th Theory of Cryptography Conference, TCC 2009*, 2009, pp. 19–35.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 34, pp. 211–407, 2014.

[30] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth, "Guilt-free data reuse," *Commun. ACM*, vol. 60, no. 4, pp. 86–93, 2017.

[31] A. Beimel, I. Haitner, N. Makriyannis, and E. Omri, "Tighter bounds on multi-party coin flipping, via augmented weak martingales and differentially private sampling," Electronic Colloquium on Computational Complexity, Tech. Rep. TR17-168, 2017.

[32] A. DasGupta, *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*, ser. Springer Texts in Statistics. Springer New York, 2011.