On Fair Exchange, Fair Coins and Fair Sampling[⋆]

Shashank Agrawal and Manoj Prabhakaran

University of Illinois, Urbana-Champaign
`sagrawl2@illinois.edu, mmp@illinois.edu`

Abstract. We study various classical secure computation problems in the context of fairness, and relate them with each other. We also systematically study fair sampling problems (i.e., inputless functionalities) and discover three levels of complexity for them.
Our results include the following:

– Fair exchange cannot be securely reduced to the problem of fair coin-tossing by an r-round protocol, except with an error that is $\Omega(\frac{1}{r})$.
– Finite fair *sampling* problems with rational probabilities can all be reduced to fair coin-tossing and unfair 2-party computation (or equivalently, under computational assumptions). Thus, for this class of functionalities, fair coin-tossing is complete.
– Only sampling problems which have fair protocols without any fair setup are the trivial ones in which the two parties can sample their outputs independently. Others all have an $\Omega(\frac{1}{r})$ error, roughly matching an upperbound for fair sampling from [21].
– We study communication-less protocols for sampling, given another sampling problem as setup, since such protocols are inherently fair. We use spectral graph theoretic tools to show that it is impossible to reduce a sampling problem with *common information* (like fair coin-tossing) to a sampling problem without (like "noisy" coin-tossing, which has a small probability of disagreement).

The last result above is a slightly sharper version of a classical result by Witsenhausen from 1975. Our proof reveals the connection between the tool used by Witsenhausen, namely "maximal correlation," and spectral graph theoretic tools like Cheeger inequality.

1 Introduction

Despite wide interest in the problem of fairness, our understanding of some of the most fundamental questions about it is greatly lacking. In this work, we study fair-exchange, fair coin-flipping, and more generally fair sampling, to understand the relation between these primitives. In the process, we also obtain a sharper version of a classical information theory result from the 70's on common information of correlated random variables.

[⋆] Research supported in part by NSF grants 1228856 and 0747027.

Fair coin-flipping and fair-exchange are two classical problems in cryptography, with a long history of results, both positive and negative. The most influential, and perhaps the most important negative result, dates back to the work of Cleve [8], in which a deceptively simple argument was used to prove a result of great consequence: irrespective of what computational assumptions are used, any 2-party coin-flipping protocol is vulnerable to a simple attack which can produce a bias that is inversely proportional to the number of rounds of the protocol (rather than being negligible, as one would have preferred).

Our first result relates fair coin-flipping to fair-exchange. It is easy to see that a fair exchange functionality can be directly used to obtain a fair coin-flipping protocol (and thus Cleve's impossibility for fair coin-flipping implies impossibility of fair-exchange as well). We ask if fair coin-flipping and fair-exchange are *equivalent*, possibly under some computational assumption. That is, *given access to a fair coin-flipping functionality, can we implement a fair-exchange protocol?*

The answer turns out to be negative: we show that an efficient attack can break the security of any fair-exchange protocol that has access to fair coin-flipping, in the same way Cleve's attack could break the security of any fair coin-flipping protocol. Our attack, like Cleve's, is a simple fail-stop attack, and does not rely on any computation other than running the steps of the protocol itself. However, it differs from Cleve's in some essential ways, in order to handle the presence of the fair coin-flipping functionality. (In particular, one of our attacks requires the adversary to run a particular round of the protocol twice, to "look-ahead" before actually accessing the coin-flipping functionality.)

Our other results relate to the problem of *fair sampling*. This is a generalization of the fair coin-flipping problem, in which it is not necessary that Alice and Bob output the same bit, but instead they are required to produce outputs that are correlated in a specified manner. While somewhat more subtle than the problem of fair coin-flipping, one can use a natural (standalone) simulation based security definition to get the right definition of fair and secure sampling. Surprisingly, we show that fair coin-flipping is at least as "complex" as generating correlated outputs from various distributions like noisy coin-flipping (where each party gets an unbiased coin, but with probability say, 0.1, their coins do not agree), random-OT (where Alice gets two random bits (x_0, x_1) and Bob gets (b, x_b) for a random b). That is, all these fair sampling problems can be solved with access to a fair coin-flipping functionality (under standard computational assumptions, or alternately, with access to *unfair* 2-party computation functionalities). On the other hand, we believe the converse does not hold in general. We give results (including one of independent interest) that show that somewhat restricted protocols cannot give fair coin-flipping from fair sampling functionalities if the distribution does not provide any *common information* (as formalized by [11]) to the two parties.

Two points are worth highlighting here. In standard (unfair) secure 2-party computation, the "complexity" of coin-flipping and that of say, noisy coin-flipping are inverted. Indeed, noisy coin-flipping is a *complete* functionality for unfair 2-

party computation in the information-theoretic setting, whereas coin-flipping is not (see for e.g. [18]). The second point is that, noisy coin-flipping and random OT, though possibly strictly simpler than coin-flipping itself, still turn out to be impossible to fairly and securely implement, irrespective of any computational assumptions or setups. We emphasize that these are sampling problems, and should not be confused with (automatically fair) functionalities like OT (with inputs). We generalize the proof of Cleve to show that unless a 2-party distribution is trivial (i.e., the outputs for the two parties are independent of each other), it does not have a fair protocol. In fact, our proof leads to a slight simplification of Cleve's argument, but without yielding any quantitative improvements.

Finally, an important contribution of our work on fair sampling is a deeper understanding of *common information*, a concept introduced by Gács and Körner [11], and since then widely studied in the information theory literature. Roughly speaking, common information of a 2-party distribution is a piece of information two parties can agree on, after they obtain a sample from the 2-party distribution (with each party obtaining only its part of the output). Distributions like noisy coin-flipping, and random OT have no common information. Gács and Körner showed that, even if a large number of samples from such a distribution are given to the two parties, if they must agree on a common output without further communication, then the *entropy rate* of their outputs must be zero. Our interest in this setting, where the parties have to agree on an output without any communication, is because such a protocol is inherently fair (provided the access to the samples are fair). The original proof of Gács and Körner used tools from ergodic theory to show that the number of independent random bits that Alice and Bob can agree on is $o(n)$ if they access n samples from a distribution with zero common information. Witsenhausen used maximal correlation [12,22] to show that they cannot agree on an output with any positive entropy (not entropy rate) except by suffering a constant probability of disagreement [23]. We reprove this result using tools from spectral graph theory. Technically, our proof is quite similar to that in [19] who refined Witsenhausen's proof that used maximal correlation. However, by identifying the connection with spectral graph theory, we are able to obtain a slightly sharper result, afforded to us by Cheeger's inequality [5].

Our Results. We provide a collection of results on fair 2-party computation (all of which also extend to the case of multi-party computation without honest majority). Our focus is on studying certain important and representative tasks, rather than attempting to exhaustively characterize fairness of all the tasks. The following three canonical problems can be used to explain our main results.

- Fair exchange $\mathcal{F}_{\text{EXCH}}$. Alice and Bob exchange a single bit.
- Fair coin-tossing $\mathcal{F}_{\text{COIN}}$. Alice and Bob obtain a common coin.
- Fair sampling of a random instance of oblivious transfer $\mathcal{F}_{\text{ROT}}$: Alice gets a pair of random bits (x_0, x_1) and Bob gets (b, x_b) where b is a random bit.

We show that these three functionalities have decreasing complexity in the context of fairness:

1. $\underline{\mathcal{F}_{\text{EXCH}} > \mathcal{F}_{\text{COIN}}}$: In Section 3, we show that $\mathcal{F}_{\text{EXCH}}$ cannot be reduced to $\mathcal{F}_{\text{COIN}}$, irrespective of what computational assumptions are made. We show that for any r-round protocol for $\mathcal{F}_{\text{EXCH}}$ using $\mathcal{F}_{\text{COIN}}$, there is an efficient fail-stop adversary for which the simulation error is $\Omega(\frac{1}{r})$. On the other hand, it is well-known that $\mathcal{F}_{\text{COIN}}$ can be reduced to $\mathcal{F}_{\text{EXCH}}$.
2. $\underline{\mathcal{F}_{\text{COIN}} \geq \mathcal{F}_{\text{ROT}}}$: In Section 4 we show that $\mathcal{F}_{\text{ROT}}$ can be reduced to $\mathcal{F}_{\text{COIN}}$ (and an unfair 2-party computation problem). This protocol involves no communication between the two parties, except for them both accessing an unfair sampling functionality, and $\mathcal{F}_{\text{COIN}}$. This protocols extends beyond $\mathcal{F}_{\text{ROT}}$, and shows that $\mathcal{F}_{\text{COIN}}$ is complete with respect to fair and secure reductions at least for a class of "nice" sampling tasks (including $\mathcal{F}_{\text{ROT}}$).
3. $\underline{\mathcal{F}_{\text{COIN}} > \mathcal{F}_{\text{ROT}}?}$ We do not completely rule out a reduction of $\mathcal{F}_{\text{COIN}}$ to $\mathcal{F}_{\text{ROT}}$. However, we present important partial negative results in Section 5. In particular, we show that there is no *logarithmic round reduction* from coin flipping to a distribution with zero common information. (Also see below.)
4. $\underline{\mathcal{F}_{\text{ROT}} \text{ non-trivial}}$: Though $\mathcal{F}_{\text{ROT}}$ is at the bottom of this list, in Section 6 we show that it cannot be fairly sampled either (irrespective of the computational assumptions used). Here we have a tight characterization: only distributions that can be fairly sampled are the ones in which there is no correlation between Alice's and Bob's outputs. Our result is also tight in that the bias we obtain closely matches a positive result from [21].

In Section 5 we investigate a sub-class of protocols for fair sampling, in which the two parties access samples from a setup functionality, and then, without any communication, produce their outputs. Such protocols are inherently fair. The question of when such protocols are possible presents interesting combinatorial and information-theoretic questions.

Using tools from spectral graph theory to analyze an appropriately defined graph product (or rather, bipartite graph product), we show that even with an unbounded number of samples from $\mathcal{F}_{\text{ROT}}$, any such protocol for $\mathcal{F}_{\text{COIN}}$ will have a *constant amount* of error. Specifically, we give a tight bound on the second eigenvalue of the normalized Laplacian of $G_1 \boxtimes G_2$ in terms of that of G_1 and G_2, where $\boxtimes$ is a natural bipartite graph product that we define. Our result sharpens a classical result on "common information" from information theory, originally proven by Gács and Körner [11] using techniques from ergodic theory and subsequently improved by Witsenhausen [23] using the "maximal correlation" measure [12,22]. As it turns out, our spectral graph theoretic proof is very similar to the one using maximal correlation as reformulated in [19], but we can make a slightly sharper statement relating the error when using one sample from $\mathcal{F}_{\text{ROT}}$ versus using an unbounded number of samples, thanks to Cheeger inequality.

This result also goes beyond $\mathcal{F}_{\text{ROT}}$, and in fact gives a tight characterization of which sampling problems allow $\mathcal{F}_{\text{COIN}}$ to be reduced to them, and which ones do not: any 2-party distribution (i.e., a pair of correlated random variables) which has non-zero "common information" can be used to implement $\mathcal{F}_{\text{COIN}}$, where common information is as was defined in the seminal work of [11].

We believe the explicit connection between spectral graph theory and tools in information theory is of independent interest, and holds promise for other problems.

An Emerging Picture. While the focus on this work has been to study specific functionalities, our results suggest a certain hierarchy of "complexity" of functionalities. Firstly, in general fair functions with input (like XOR) can be strictly more complex than fair sampling problems. We leave it open to study distinctions within functions with input (e.g., both parties having input vs. only one party having input). Our other results have explored variations among fair sampling problems. There are three apparent classes here: trivial problems (which can be sampled trivially, by both parties independently generating their outputs), non-trivial problems with zero common information (which includes $\mathcal{F}_{\text{ROT}}$ and noisy coin-flip), and problems with non-zero common information (which includes $\mathcal{F}_{\text{COIN}}$). Indeed, the last class is complete for all sampling problems with rational probabilities. The qualitative separation between problems with and without common information is formalized in the setting of protocols without communication.

Related Work. The problem of fairness in multi-party computation goes back to the work of Even and Yacobi [10] where exchange of digital signatures is informally proved to be impossible. The first rigorous proof of the impossibility of fairly computing a functionality comes from Cleve's work [8]. He showed that a very basic functionality, that of tossing a coin, cannot be realized fairly. Subsequent works like that of [9] which considered stronger attacks, relied on computationally unbounded adversaries.

A recent series of results has renewed interest in the area of fairness. Starting with the work of Gordon et al. [14], where they show that several functionalities of interest can be realized with complete fairness, there has been a series of results in this area. In [21], Moran et al. solve a long standing open problem in fairness. They show that Cleve's lower bound on the bias of coin tossing protocols can be achieved (up to a factor of 2) by a protocol. Beimel et al. [3] extend their results to the multi-party model when less that 2/3 of the parties are corrupt. In [15], Gordon et al. study the question of reductions among fair functionalities. They show that no *short* primitive is complete for fairness. They also establish a fairness hierarchy for simultaneous broadcast. Further in [13], a definition of partial fairness is proposed, and it is shown that any two-party functionality, at least one of whose domains or ranges is polynomial in size, can be realized fairly under this definition. Beimel et al. [2] study partial fairness in the multi-party setting. Asharov et al. [1] provide a complete characterization of functions that imply fair coin tossing, and hence cannot be computed fairly due to Cleve's impossibility result. The negative results in this work relied on computationally unbounded adversaries.

Separations of the kind we consider (impossiblity of reducing XOR to coin-flipping) was also considered in the context of security with abort, but in the computationally unbounded setting [20]. We remark that such a result does not hold in the computationally bounded setting.

The notion of common information was introduced by [11], and further developed in [24,16,23] and many later works. The problem of obtaining isoperimetric inequalities of graph products has been studied, but for notions of graph products different from the bipartite product we study (e.g. [7], also see [6]).

2 Preliminaries

2.1 Secure two-party computation with complete fairness

We are interested in (possibly randomized) two-party secure function evaluation with complete fairness (in contrast to security with abort). The functionalities we consider are all finite and their domains and ranges remain constant, irrespective of the security parameter. All entities considered are probabilistic polynomial time (PPT). They are given the security parameter n as an auxiliary input, and their total running time is polynomial in n.

Fairness is modeled by specifying the ideal functionality $\mathcal{F}$ to be fair: it delivers the output to both parties together. A corrupt party controlled by the adversary may explicitly instruct the ideal functionality to abort (or, provide $\perp$ as its input) without receiving any information from the functionality; but if both parties provide valid inputs, then the functionality will evaluate a specified function of the inputs and provide the results to the parties. Let $\textsc{ideal}_{\mathcal{F},\mathcal{S}}(n) = (\textsc{view}_{\mathcal{F},\mathcal{S}}(n), \textsc{out}_{\mathcal{F},\mathcal{S}}(n))$ be the random variable that denotes the output of the adversary and the output of the honest party in the ideal world.

In the real world, instead of outsourcing the computation, parties run a protocol π which enables them to compute $\mathcal{F}$. While the honest party follows the instructions of π, the corrupt party controlled by the adversary may deviate arbitrarily. Let $\textsc{real}_{\pi,\mathcal{A}}(n) = (\textsc{view}_{\pi,\mathcal{A}}(n), \textsc{out}_{\pi,\mathcal{A}}(n))$ be the random variable that denotes the view of the adversary and the output of the honest party.

A more detailed description of the ideal and real executions can be found in the full version of the paper. In proving our negative results, we use the following weak simulation based security definition.

Definition 1 (Weak Security). *A protocol π is said to be a weak ϵ-secure realization of a two party functionality $\mathcal{F}$ if for every PPT adversary $\mathcal{A}$ in the real world, there exists a PPT adversary $\mathcal{S}$ in the ideal world such that*

$$\Delta\left(\textsc{ideal}_{\mathcal{F},\mathcal{S}}(n), \textsc{real}_{\pi,\mathcal{A}}(n)\right) \le \epsilon(n).$$

We say that π is a weak secure realization of $\mathcal{F}$, if it is a weak ϵ-secure realization of $\mathcal{F}$ for a negligible function $\epsilon(n)$.

Our definition is similar to the one given in [13], except that we do not require security to hold in the presence of auxiliary information, which makes our definition weaker. Note that using a weaker security definition only strengthens the impossibility results. On the other hand, we remark that our positive results, i.e., constructions, are in fact UC secure [4].

2.2 Normal Form of a Protocol

We shall use the following normal form for a 2-party protocol π between Alice and Bob. The number of rounds of the protocol will be denoted by $r(n)$, where n is the security parameter. In this protocol, parties may also have access to a *setup functionality* $\mathcal{G}$. We shall often refer to such a setup functionality as an *oracle*. Without loss of generality, we assume that the i^{th} round in π consists of the following steps, for $1 \leq i \leq r(n)$:

- Alice sends a message to Bob; if Alice aborts without sending this message, Bob produces an output, denoted by the random variable Y_{i-1}.
- the functionality $\mathcal{G}$ is invoked; if this invocation is aborted, Alice and Bob would produce outputs.
- then Bob sends a message to Alice; if Bob aborts without sending this message, Alice produces an output, denoted by the random variable X_i.
- $\mathcal{G}$ is invoked once again; again, if this invocation is aborted, Alice and Bob would produce outputs.

In all our results, the functionality $\mathcal{G}$ will be an inputless function, and the particular attacks we use do not involve aborting its invocation. So we have not given any names for the random variables corresponding to the outputs if $\mathcal{G}$'s invocation is aborted. If multiple setups, say $\mathcal{G}_1$ and $\mathcal{G}_2$, are available, they will be invoked one after the other in every round.

We remark that what makes proving our impossibility results harder is that the protocol π can access $\mathcal{F}_{\text{COIN}}$ throughout its execution, rather than only in a pre-processing phase. Indeed, it has been observed before by Ishai et al. [17] that the impossibility results for fair deterministic function evaluation in the plain model continue to hold in a pre-processing model.[1]

3 Fair Exchange from Fair Coin-Flipping

In this section, our goal is to show that two parties cannot exchange their bits fairly, even when given access to fair coin-flipping functionality. The $\mathcal{F}_{\text{COIN}}$ func-

[1] The observation in [17] considers not just deterministic function evaluation. However in the general case, the impossibility of fairness there holds only under a stricter requirement, that the correctness of the protocol should hold conditioned on the randomness of the pre-processing phase. In particular, $\mathcal{F}_{\text{COIN}}$ does not reduce to $\mathcal{F}_{\text{COIN}}$ in such a pre-processing model. Our results are not restricted to the pre-processing model, nor depend on such a security requirement.

tionality does not take any input and provides a bit uniformly distributed in $\{0,1\}$ to the two parties. The $\mathcal{F}_{\text{EXCH}}$ functionality is also simple to state: if $x, y \in \{0,1\}$, then $\mathcal{F}_{\text{EXCH}}(x,y) = (y,x)$; but if one of the parties aborts or sends an invalid input to it, the functionality substitutes its input by a default value, say 0.[2] Recall our convention that this is a fair functionality, so the adversary cannot prevent the delivery of output to the honest party.

We define another functionality $\mathcal{F}_{\text{XOR}}$ which takes inputs x and y from the two parties. If $x, y \in \{0,1\}$, then $\mathcal{F}_{\text{XOR}}(x,y) = (x \oplus y, x \oplus y)$; but if one of the parties aborts or sends an invalid input, the functionality substitutes its input by a default value, say 0 (similar to what $\mathcal{F}_{\text{EXCH}}$ does above). The functionality $\mathcal{F}_{\text{XOR}}$ is "isomorphic" to $\mathcal{F}_{\text{EXCH}}$: that is, each functionality can be (UC) securely reduced to the other using a protocol that involves no other communication other than a single invocation of the latter functionality. Then it is easy to see that the fair functionality $\mathcal{F}_{\text{EXCH}}$ can be (weakly) securely realized (using any set up) if and only if the fair functionality $\mathcal{F}_{\text{XOR}}$ can be (weakly) securely realized (using the same set up). This allows us to prove that $\mathcal{F}_{\text{EXCH}}$ cannot be reduced to $\mathcal{F}_{\text{COIN}}$ by showing instead that $\mathcal{F}_{\text{XOR}}$ cannot be reduced to $\mathcal{F}_{\text{COIN}}$.

The result of this section follows. A formal proof can be found in the full version of the paper. Here we provide a sketch which describes the main ideas involved in the proof. We point out that the result is tight up to a constant, since [13] shows that $\mathcal{F}_{\text{XOR}}$ can be computed ϵ-securely in $O(1/\epsilon(n))$ rounds even without access to $\mathcal{F}_{\text{COIN}}$.

Theorem 1. *For any weakly ϵ-secure protocol $\pi^{\mathcal{F}_{\text{COIN}}}$ that realizes the functionality $\mathcal{F}_{\text{XOR}}$ and runs in $r(n)$ rounds, $r(n) \in \Omega(\frac{1}{\epsilon(n)})$.*

Proof sketch: Similar to Cleve's approach [8], we shall consider a collection of fail-stop adversaries that corrupt either Alice or Bob. We shall also consider the case when neither party is corrupt. We seek to argue that at least for one of these adversaries, the outcome in the real experiment cannot be simulated within a $\Omega(\frac{1}{r})$ error by any simulator in the ideal world, where r is the number of rounds in $\pi^{\mathcal{F}_{\text{COIN}}}$. ($r$ is a function of the security parameter n, but for the sake of readability, we write r instead of $r(n)$.)

We start off along the same lines as Cleve: we note that at the end of the protocol, the parties will agree on their outcome (except with at most ϵ probability). On the other hand, in the beginning of the protocol, the variables Y_0 and X_1 are independent of each other; also, by considering an adversary who forces an abort right at the beginning, each of Y_0 and X_1 should be close to uniformly random. So Y_0 and X_1 are equal with probability only about half. Thus there must be a round i such that $\mathsf{Pr}[X_i = Y_i] - \mathsf{Pr}[X_i = Y_{i-1}] = \Omega(\frac{1}{r})$ (or

[2] An alternate formulation would be that if $(x,y) \notin \{0,1\}^2$, then $\mathcal{F}_{\text{EXCH}}(x,y) = (\perp,\perp)$ where $\perp$ is a special symbol indicating abort. It can be easily seen that these formulations are "isomorphic" to each other (see following text).

$\mathsf{Pr}[X_{i+1} = Y_i] - \mathsf{Pr}[X_i = Y_i] = \Omega(\frac{1}{r})$; w.l.o.g, we can consider the former to be the case).

In Cleve's case, where the protocol he considers does not have access to any setup, we can consider two adversaries that corrupt Alice, and selectively abort at round i as follows. (See Section 2 for the numbering of the rounds.) The first adversary forces Bob to output Y_{i-1} if $X_i = 0$ and otherwise forces him to output Y_i; the second adversary does the same for $X_i = 1$. To ensure that Bob's output is unbiased under these two attacks requires that

$$\mathsf{Pr}[Y_{i-1} = 0 \wedge X_i = 0] \approx \mathsf{Pr}[Y_i = 0 \wedge X_i = 0],$$
$$\mathsf{Pr}[Y_{i-1} = 1 \wedge X_i = 1] \approx \mathsf{Pr}[Y_i = 1 \wedge X_i = 1].$$

This contradicts the assumption that $\mathsf{Pr}[X_i = Y_i] - \mathsf{Pr}[X_i = Y_{i-1}] = \Omega(\frac{1}{r})$. A crucial element in this proof is that Alice can compute X_i first and then selectively force Bob to output Y_{i-1}.

Unfortunately, but not surprisingly, this breaks down when the parties have access to the $\mathcal{F}_{\text{COIN}}$ oracle as in our case. To compute X_i, Alice must obtain the first coin in round i. But after that she cannot force Bob to output Y_{i-1}: he will output only Y_i (note that aborting an access to $\mathcal{F}_{\text{COIN}}$ cannot help, because w.l.o.g, the protocol can instruct a party to substitute it with a coin it generates). Indeed, we cannot expect Cleve's argument to go through when an $\mathcal{F}_{\text{COIN}}$ oracle is present, because fair coin-flipping is trivially possible given access to $\mathcal{F}_{\text{COIN}}$.

The reason we can expect to have an attack nevertheless, has to do with the fact that there is an additional correctness requirement in the case of $\mathcal{F}_{\text{XOR}}$ that is not present in the case of coin-flipping. For instance, if the parties were to output a coin they obtain in round i as their final output, while none of the attacks can bias this outcome, when the execution is carried out without any corruption, the output will be different from the XOR of the input with probability $\frac{1}{2}$.

We leverage this fact in a somewhat non-obvious manner. Suppose we want to run Cleve's attacks as well as we can. The two adversaries described above can proceed right up to the point before accessing $\mathcal{F}_{\text{COIN}}$ in round i. Then, without invoking $\mathcal{F}_{\text{COIN}}$, the attacker can check what the value of X_i would be for each of the two possible outcomes from $\mathcal{F}_{\text{COIN}}$ (by feeding one value of the coin to the honest protocol execution, then rewinding it, and feeding the other value of the coin). If in both cases the outcome is the same, then the adversary manages to find X_i without invoking $\mathcal{F}_{\text{COIN}}$ at all. Let E_i^A denote this event that in an honest execution of the protocol, at the point before invoking the first access to $\mathcal{F}_{\text{COIN}}$ in round i, the value of X_i already gets determined.

But what happens if the complement event $\overline{E}_i^A$ occurs? In this case, X_i is a 0 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$. Further, this happens independently of Y_{i-1}. Note that Y_i could very well be correlated with X_i, since it is influenced by the same coin that decides X_i as well as messages sent by Alice after determining X_i. On the other hand, the final output of Bob, Y_r must be (almost) independent of X_i, since it must (mostly) equal the XOR of the inputs, which is fixed well before the coin from $\mathcal{F}_{\text{COIN}}$ is accessed. Thus, X_i is correlated almost the same way (i.e., uncorrelated) with both Y_{i-1} and Y_r.

This gives us a way to emulate the effect of forcing the outcome to be Y_{i-1} when X_i comes out a particular way, provided $\overline{E}_i^A$ occurs: instead of trying to force Bob to output Y_{i-1} (for which it is too late), let the protocol run to completion and force his outcome to be Y_r.

Somewhat surprisingly, this intuition can be turned into a concrete argument. We employ adversaries for each round which check if the event E_i^A occurs, and adopt one of the above strategies. Note that the adversary can efficiently determine if the event E_i^A occurs (without accessing the corresponding instance of $\mathcal{F}_{\text{COIN}}$). $\square$

In the full version, we actually prove a generalization of Theorem 1. We show that no δ-balanced function [1] can be securely reduced to $\mathcal{F}_{\text{COIN}}$. The XOR function, which is δ-balanced with $\delta = 1/2$, is therefore not reducible to $\mathcal{F}_{\text{COIN}}$ either.

Remark. The proof and the result readily extends to the case when the protocol has access to other *unfair* non-reactive functionalities as well as $\mathcal{F}_{\text{COIN}}$, since in that case Alice can determine whether the event E_i^A occurs (using an unfair access to the functionalities) and act accordingly. Also, a corollary of the generalization of Theorem 1 is that access to any fair functionality that can be securely realized using access to (polynomially many invocations of) a δ-balanced function is not sufficient to obtain a secure fair XOR protocol.

4 Fair sampling from Fair Coin-flipping

We shall say that a functionality $\mathcal{F}$ is complete for fair function evaluation if for any fair function evaluation task there is an (information theoretically) secure protocol that uses $\mathcal{F}$ *and optionally, some unfair functionality* $\mathcal{G}$. Allowing access to an unfair functionality eliminates the need to base the completeness result on computational assumptions. (Equivalently, one could define it in terms of a reduction to $\mathcal{F}$ that is secure in the probabilistic polynomial time setting, and assume the existence of oblivious transfer protocol.)

In [15] it was shown that no finite functionality is complete for fair computation, even restricted to finite functionalities. We pose the same question, but restricted to finite sampling functionalities (i.e., functionalities without input).

Surprisingly, we show that fair coin-flipping functionality $\mathcal{F}_{\text{COIN}}$ is in fact complete for this class of problems. We mention a caveat in our result: our protocol for fair sampling requires that the probability values in the target distribution are rational numbers. Note that since the functionalities are finite, there is only a finite constant set of probabilities in question, independent of the security parameter. We say that such distributions are “nice.” If the target distribution involves probabilities that are not rational, then even though one could approximate them to negligible error using rational numbers, an initial unfair secure computation phase in our protocol would involve exponentially large outputs. However, even in this case, the number of accesses to $\mathcal{F}_{\text{COIN}}$ is still only polynomial.

Let p_{XY} denote a joint distribution over two random variables X and Y which take values in the finite domains $\mathcal{X}$ and $\mathcal{Y}$ respectively. Our goal is to construct an information-theoretic, UC secure protocol for the functionality which takes no input, but samples (X, Y) according to p_{XY} and gives X to Alice and Y to Bob. This functionality is modeled as a fair functionality as described in Section 2. The protocol has access to an arbitrary unfair 2-party computation problem (in fact, a 2-party sampling problem suffices) and the fair coin tossing function $\mathcal{F}_{\text{COIN}}$.

The basic idea of the protocol is fairly simple: first use an unfair secure computation phase to generate two lists A and B such that if a uniformly random i is picked then (A_i, B_i) will be distributed according to the target distribution. This computation will give the list A to Alice and B to Bob. Then, use (many accesses to) fair coins to sample an index i; if either party aborts mid-way, the other party simply tosses the remaining coins on its own. Alice's output will be A_i and Bob's output will be B_i. The list A could contain many indices i such that $A_i = x$ for some character x, such that not all of these indices have the same value of B_i. For security of this protocol, it is important that if Alice receives i such that $A_i = x$, she learns nothing more about Bob's output B_i, than what x reveals. This is ensured by randomly permuting the lists A and B.

It remains to describe how the lists A and B can satisfy the above requirements. For this, first, for each $(x, y) \in \mathcal{X} \times \mathcal{Y}$, express the probability $p(x, y)$ as rational number $P_{x,y}/Q$, where Q is the same for all (x, y). Note that this is where we assume that the distribution p_{XY} is nice. Then, for each pair (x, y), add $P_{x,y}$ copies of (x, y) to a list L. The size of this list will be Q. Then randomly permute L to obtain a list $((a_1, b_1), \cdots (a_Q, b_Q))$. A is defined to be the list $(a_1, \cdots, a_Q)$ and B the list $(b_1, \cdots, b_Q)$.

Simulation to prove the security of this protocol is straightforward and omitted.

Despite the restriction to nice distributions, we note that the consequences of this protocol are already quite powerful. The sampling problems mentioned in the introduction — noisy coin-flipping with noise probability 0.1 or random oblivious transfer distribution (in which one of 8 different possibilities occur with the probability 1/8 each) — are covered by this protocol.

We also point out another feature of this protocol: Alice and Bob do not interact with each other, except by accessing two sampling functions: the first one which produces the lists A and B (unfairly) and the second one which gives fair coins.

5 Impossibility of Fair Coin Flip from Fair Sampling

In this section we ask if it is possible to have a fair coin flipping protocol given access to a setup for fairly sampling from a 2-party distribution. As we shall see, this depends on whether the setup distribution gives *non-zero common information* to the two parties. Our definition of common information of a 2-party distribution, adapted from Gács and Körner [11], is best understood in terms of the characteristic bipartite graph representation of a 2-party distribution.

Characteristic Bipartite Graph. Consider a distribution which samples a pair of symbols $(u, v) \in U \times V$ with probability $p(u, v)$ and gives u to Alice and v to Bob. The characteristic bipartite graph (or simply the graph of a distribution) of this distribution is a weighted graph $G = (U, V, w)$ with U and V as the two partite sets, and with weight of the edge between $u \in U$ and $v \in V$ defined to be $w(u, v) = p(u, v)$. Edges with weight 0 are considered absent, and only nodes with at least one edge incident on them are retained in G.

Common Information. In the above setting, consider a function C which maps a sample (u, v) to the index of the connected component in G that contains the edge (u, v) (after removing 0-weight edges). We define the common information of a 2-party distribution as the entropy of the random variable $C(u, v)$ when (u, v) is sampled from the distribution. In particular, the distribution has zero common information iff G has a single connected component (after removing 0-weight edges and isolated nodes).

For example, 2-party coin flipping has 1 bit of common information, whereas a noisy coin flipping which gives an unbiased coin each to Alice and Bob which are equal only with probability say 0.9, has zero common information.

Conjecture 1. For the class of finite 2-party distributions, the ones that are complete with respect to fair and secure reductions are exactly the ones that have positive common information.

We do not completely resolve this conjecture, but we provide the following results in its evidence:

1. In the positive direction, the conjecture is equivalent to stating that coin flipping is complete (since, as can be easily seen, any distribution with positive common information can be used to obtain fair coins). Our result in Section 4 proves this, restricted to the class of "nice" distributions.

2. In the negative direction, we show that there is no *logarithmic round reduction* from coin flipping to a distribution with zero common information. We present this proof in the full version of the paper.

3. We show that there is no reduction from coin flipping to a distribution with zero common information using a protocol that has (an unbounded number of) rounds which access the setup, followed by a polynomial number of communication rounds. (Our proof does not apply if the accesses to the setup are

interspersed with communication.) This can be shown using Theorem 2 below, which deals with the special case when the protocol involves no communication at all.

Theorem 2. *[23] Let p_{UV} be a 2-party distribution with zero common information. Then for every constant $\delta > 0$ there is a constant $\epsilon > 0$ (depending on p_{UV}) such that for any 2-party protocol in which the parties are given an arbitrary number of samples from p_{UV}, but they do not exchange any messages and the entropy of the output of at least one of the parties is at least δ, then with probability at least ϵ the output of the two parties will be different.*

In Appendix A we give a new proof for this result, originally due to [23]. Note that the error probability ϵ does not decrease with the number of coin samples the protocol is allowed access to. Further, Lemma 1 in Appendix A, implies that the error ϵ_0 achievable using *a single sample* from p_{UV} is (for the same δ) $O(\sqrt{\epsilon})$. That is, using more than one sample can decrease the error probability at most quadratically. To the best of our knowledge, this final result was not known previously.

6 Secure Sampling

In this section we consider the task of sampling from a joint distribution (X, Y), where X and Y are distributed over finite domains $\mathcal{X}$ and $\mathcal{Y}$ respectively. Two parties Alice and Bob wish to sample from this distribution such that Alice learns only the value of X and Bob learns only the value of Y. The functionality for secure sampling $\mathcal{F}_{\text{ss}}$ is very simple: it does not take any input and produces a sample from the distribution (X, Y).

Let $X \times Y$ denote the product distribution of X and Y, i.e., $\mathsf{Pr}[X \times Y = (x, y)] = \mathsf{Pr}[X = x] \cdot \mathsf{Pr}[Y = y]$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We show:

Theorem 3. *For any weakly ϵ-secure protocol π that realizes the functionality $\mathcal{F}_{\text{ss}}$ and runs in $r(n)$ rounds,*

$$r(n) \geq \frac{\Delta((X,Y), X \times Y) - 3\epsilon(n)}{2(|\mathcal{X}| + |\mathcal{Y}|)\epsilon(n)}, \tag{1}$$

$$r(n) \geq \frac{\alpha_{XY} - 3\epsilon(n)}{4\epsilon(n)}, \tag{2}$$

where $\alpha_{XY} = \mathsf{max}_{(x,y)\in(\mathcal{X},\mathcal{Y})} |\mathsf{Pr}[(X,Y) = (x,y)] - \mathsf{Pr}[X \times Y = (x,y)]|$.

In general, the two bounds are incomparable: the nature of joint distribution decides which one is stronger (for examples see the full version). Our first bound closely matches an upper bound from [21], who give an ϵ-secure sampling protocol with $\frac{\Delta((X,Y),X\times Y)}{2\epsilon(n)} + c$ rounds, where c is a positive constant. We prove this bound in the full version of the paper. Here we prove the second bound in the above theorem. Our proof is a natural generalization (and perhaps a slight simplification/clarification) of Cleve's proof for fair coin-tossing.

Proof of (2): Consider a weakly ϵ-secure protocol π for secure sampling that runs in $r(n)$ rounds. In a single round, Alice sends a message to Bob followed by Bob sending a message to Alice. Recall the definitions of X_i and Y_{i-1} for $1 \le i \le r(n)+1$ from Section 2. Since we are working in the plain model here (without any oracle set-up), Alice (resp. Bob) can compute the value of X_i (resp. Y_i) before sending the message for round i.

For simplicity in the following, we omit the security parameter n. We also assume that Alice and Bob always output a value from $\mathcal{X}$ and $\mathcal{Y}$ respectively when the other party aborts (for more discussion, see the full version). Fix a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and define four adversaries as shown in Figure 5 for each $1 \le i \le r$.

Adversary $\mathcal{A}_{i,x}$:

```
  Simulate Alice for $i-1$ rounds
  **if** $X_i = x$ **then**
    abort at round $i+1$
  **else**
    abort at round $i$
  **end if**
```

Adversary $\mathcal{B}_{i,y}$

```
  Simulate Bob for $i-1$ rounds
  **if** $Y_i = y$ **then**
    abort at round $i+1$
  **else**
    abort at round $i$
  **end if**
```

Adversary $\mathcal{A}_i$

```
  Simulate Alice for $i-1$ rounds
  Abort
```

Adversary $\mathcal{B}_i$

```
  Simulate Bob for $i-1$ rounds
  Abort
```

Fig. 5 Adversaries for round i, where $1 \le i \le r$.

Let us first consider the probability that Bob outputs y when Alice is corrupted by $\mathcal{A}_{i,x}$ in the real world.

$$\begin{aligned}\Pr[\text{OUT}_{\pi,\mathcal{A}_{i,x}} = y] &= \Pr[X_i = x \wedge Y_i = y] + \Pr[X_i \ne x \wedge Y_{i-1} = y] \\ &= \Pr[X_i = x \wedge Y_i = y] - \Pr[X_i = x \wedge Y_{i-1} = y] + \Pr[Y_{i-1} = y].\end{aligned} \tag{3}$$

When $\mathcal{A}_i$ corrupts Alice, $\Pr[\text{OUT}_{\pi,\mathcal{A}_i} = y]$ is simply $\Pr[Y_{i-1} = y]$.

On the other hand, since the sampling functionality is inputless, no matter what strategy the adversary adopts in the ideal world, the output of Bob is distributed according to the marginal distribution Y. Therefore, for all $\mathcal{S}_{\mathcal{A}_{i,x}}, \mathcal{S}_{\mathcal{A}_i}$, we have

$$\Pr[\text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{A}_{i,x}}} = y] = \Pr[\text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{A}_i}} = y] = \Pr[Y = y], \tag{4}$$

where $\mathcal{S}_{\mathcal{A}}$ denotes the ideal world counterpart of a real world adversary $\mathcal{A}$. Hence,

$$\Delta\Big(\text{OUT}_{\pi,\mathcal{A}_{i,x}}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{A}_{i,x}}}\Big) + \Delta\big(\text{OUT}_{\pi,\mathcal{A}_i}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{A}_i}}\big) \geq \left|\Pr[X_i = x \wedge Y_i = y] - \Pr[X_i = x \wedge Y_{i-1} = y]\right|. \quad (5)$$

Similarly, by considering the real and ideal world outputs of Alice when Bob is corrupted by adversaries $\mathcal{B}_{i,y}$ and $\mathcal{B}_i$, for all $\mathcal{S}_{\mathcal{B}_i}, \mathcal{S}_{\mathcal{B}_{i,y}}$, we have

$$\Delta\Big(\text{OUT}_{\pi,\mathcal{B}_{i,y}}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{B}_{i,y}}}\Big) + \Delta\big(\text{OUT}_{\pi,\mathcal{B}_i}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}_{\mathcal{B}_i}}\big) \geq \left|\Pr[Y_i = y \wedge X_{i+1} = x] - \Pr[Y_i = y \wedge X_i = x]\right|. \quad (6)$$

Adding (5) and (6) for all $1 \leq i \leq r$, we have that the sum of $4r$ statistical difference terms is at least $|\Pr[X_{r+1} = x \wedge Y_r = y] - \Pr[X_1 = x \wedge Y_0 = y]|$. Hence, there exists an adversary $\mathcal{A}$ (among the $4r$ adversaries) such that for any ideal world adversary $\mathcal{S}$,

$$\Delta(\text{OUT}_{\pi,\mathcal{A}}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}}) \geq \frac{|\Pr[X_{r+1} = x \wedge Y_r = y] - \Pr[X_1 = x \wedge Y_0 = y]|}{4r} \quad (7)$$

We want to lower bound the above quantity in terms of X and Y. To this end, observe that when neither party is corrupt, the joint distribution of Alice and Bob's outputs in the ideal world is given by (X, Y), and in the real world it is given by (X_{r+1}, Y_r). Then, the ϵ-security of π implies that

$$|\Pr[(X, Y) = (x, y)] - \Pr[(X_{r+1}, Y_r) = (x, y)]| \leq \epsilon. \quad (8)$$

We can also obtain the following from the ϵ-security of π:

$$|\Pr[(X_1, Y_0) = (x, y)] - \Pr[X \times Y = (x, y)]| \leq 2\epsilon \quad (9)$$

(see the full version for a proof). Combining (7) with (8) and (9), we get:

$$\Delta(\text{OUT}_{\pi,\mathcal{A}}, \text{OUT}_{\mathcal{F}_{\text{SS}},\mathcal{S}}) \geq \frac{|\Pr[(X, Y) = (x, y)] - \Pr[X \times Y = (x, y)]| - 3\epsilon}{4r}. \quad (10)$$

However, since π is an ϵ-secure protocol, the above quantity can be at most ϵ. Choosing a pair (x, y) that maximizes $|\Pr[(X, Y) = (x, y)] - \Pr[X \times Y = (x, y)]|$, we obtain the desired bound:

$$r \geq \frac{\alpha_{XY} - 3\epsilon}{4\epsilon}. \quad (11)$$

Acknowledgments

We thank Vinod Prabhakaran for pointing out that Theorem 2 was proven in [23], and for pointing us to [19]. We also thank the anonymous referees for helpful suggestions and pointers.

References

1. G. Asharov, Y. Lindell, and T. Rabin. A full characterization of functions that imply fair coin tossing and ramifications to fairness. In *TCC*, pages 243–262, 2013. 5, 10
2. A. Beimel, Y. Lindell, E. Omri, and I. Orlov. $1/p$-secure multiparty computation without honest majority and the best of both worlds. In *CRYPTO*, pages 277–296, 2011. 5
3. A. Beimel, E. Omri, and I. Orlov. Protocols for multiparty coin toss with dishonest majority. In *CRYPTO*, pages 538–557, 2010. 5
4. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Revised version of [?]. 7
5. J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*, 625:195–199, 1970. 3, 18
6. F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1996. 6
7. F. R. K. Chung and P. Tetali. Isoperimetric inequalities for cartesian products of graphs. *Combinatorics, Probability & Computing*, 7(2):141–148, 1998. 6
8. R. Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In J. Hartmanis, editor, *STOC*, pages 364–369. ACM, 1986. 2, 5, 8
9. R. Cleve and R. Impagliazzo. Martingales, collective coin flipping and discrete control processes. Manuscript, 1993. http://www.cpsc.ucalgary.ca/~cleve/pubs/martingales.ps. 5
10. S. Even and Y. Yacobi. Relations among public key signature systems. Technical report, Technical Report 175, Technion, Haifa, Israel, 1980. 5
11. P. Gács and J. Körner. Common information is far less than mutual information. *Problems of Control and Information Theory*, 2(2):149–162, 1973. 2, 3, 4, 6, 12, 18
12. H. Gebelein. Das statistische problem der korrelation als variations und eigenwertproblem und sein zusammenhang mit der ausgleichungsrechnung. *Zeitschrift für angew. Math. und Mech.*, 21:364–379, 1941. 3, 4
13. S. Gordon and J. Katz. Partial fairness in secure two-party computation. *Journal of Cryptology*, 25(1):14–40, 2012. Preliminary version in EUROCRYPT 2010. 5, 7, 8
14. S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell. Complete fairness in secure two-party computation. *J. ACM*, 58(6):24:1–24:37, Dec. 2011. Preliminary version in STOC 2008. 5
15. S. D. Gordon, Y. Ishai, T. Moran, R. Ostrovsky, and A. Sahai. On complete primitives for fairness. In D. Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 91–108. Springer, 2010. 5, 10
16. R. M. Gray and A. D. Wyner. Source coding for a simple network. *Bell System Technical Journal*, 53:1681–1721, 1974. 6
17. Y. Ishai, R. Ostrovsky, and H. Seyalioglu. Identifying cheaters without an honest majority. In *TCC*, pages 21–38, 2012. 7
18. J. Kilian. More general completeness theorems for secure two-party computation. In F. F. Yao and E. M. Luks, editors, *STOC*, pages 316–324. ACM, 2000. 3
19. G. Kumar. Binary renyi correlation: A simpler proof of Witsenhausen's result and a tighter upper bound. Manuscript, available at http://www.stanford.edu/~gowthamr/research/binary_renyi_correlation.pdf, 2010. 3, 4, 15

20. H. K. Maji, P. Ouppaphan, M. Prabhakaran, and M. Rosulek. Exploring the limits of common coins using frontier analysis of protocols. In Y. Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2011. 6
21. T. Moran, M. Naor, and G. Segev. An optimally fair coin toss. In *TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 1–18. Springer, March 2009. 1, 4, 5, 13
22. A. Rényi. On measures of dependence. *Acta Math. Hung.*, 10:441–451, 1959. 3, 4
23. H. S. Witsenhausen. On sequences of pairs of dependent random variables. *SIAM Journal of Applied Mathematics*, 28:100–113, 1975. 3, 4, 6, 13, 15
24. A. D. Wyner. The common information of two dependent random variables. *IEEE Transactions on Information Theory*, 21(2):163–179, 1975. 6

A A Spectral Graph Theoretic Proof of Theorem 2

Firstly, we define a natural notion of bipartite graph product, to capture the bipartite characteristic graph resulting from multiple independent samples from a 2-party distribution.

Definition 2. *If $G_1 = (U_1, V_1, w_1)$ and $G_2 = (U_2, V_2, w_2)$ are two weighted bipartite graphs, we define their* bipartite tensor product $G_1 \boxtimes G_2 = (U, V, w)$ *as a weighted bipartite graph with $U = U_1 \times U_2$, $V = V_1 \times V_2$ and $w((u_1, u_2), (v_1, v_2)) = w_1(u_1, v_1) \cdot w_2(u_2, v_2)$.*

Also, for all positive integers k we define $G^{\boxtimes k} = G^{\boxtimes k-1} \boxtimes G$, where $G^{\boxtimes 0} = K_{1,1}$ (a single edge with weight 1).

To prove Theorem 2, we shall see that it suffices to lowerbound the "Cheeger constant" of $G^{\boxtimes k}$ (for all $k \in \mathbb{N}$). Before defining the Cheeger constant, we note that we can consider a weighted bipartite graph $G = (U, V, w)$ as a general (not necessarily bipartite) weighted graph $G' = (T, w)$, where $T = U \cup V$, by extending its weight function (originally defined over $U \times V$) to cover all pairs of nodes in the graph, in a natural way: for $(v, u) \in V \times U$, $w(v, u) = w(u, v)$; for $(x, x') \in U^2 \cup V^2$, $w(x, x') = 0$. Also, as a matter of convenient notation, for every node $x \in T$, we define $w(x) = \sum_{y \in T} w(x, y)$. Also, for $S \subseteq T$, let $w(S) = \sum_{x \in S} w(x)$ and $w(S, \overline{S}) = \sum_{(x,y) \in S \times \overline{S}} w(x, y)$.

Definition 3 (Cheeger Constant). *For a weighted graph $G = (T, w)$, the Cheeger constant $h(G)$ is*

$$h(G) = \min_{S \subseteq T} \frac{w(S, \overline{S})}{\min(w(S), w(\overline{S}))}. \tag{12}$$

Lemma 1. *Given a weighted bipartite graph G, for all non-negative integers k, t, $h(G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}) \geq \frac{1}{2} h^2(G)$, where $K_{2,2}$ denotes the complete bipartite graph with weight $\frac{1}{4}$ on all four edges.*

While this does not give very tight bounds on $h(G^{\boxtimes k})$ in terms of $h(G)$, the proof of this lemma, given in the full version, in fact shows that the second eigenvalue of the normalized Laplacian matrix associated with $G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}$ (for $k > 0$) is *equal to* that of the normalized Laplacian matrix associated with G. This translates to the above bound via Cheeger inequality [5].

Now we sketch how this lemma can be used to prove Theorem 2. Let $G = (U, V, w)$ be the characteristic bipartite graph of p_{UV}. $G^{\boxtimes k} = (U^k, V^k, w^{(k)})$ denotes the graph corresponding to k independent samples from p_{UV}, where $w^{(k)}((u_1, \cdots, u_k), (v_1, \cdots, v_k)) = \prod_{i=1}^{k} w(u_i, v_i)$. Alice gets a node in U^k as her part of the sample from p_{UV}^k (i.e., k independent samples from p_{UV}), and Bob gets a node in V^k. Further, Alice and Bob may use private random coins, say t of them. The characteristic bipartite graph for t pairs of independent coins is ${K_{2,2}}^{\boxtimes t}$. Thus $G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}$ denotes the entire view of the two parties in the protocol. Now, the output of each party is a deterministic function of its view.

W.l.o.g., we assume that each party is outputting a single bit (if necessary, by partitioning the outputs into two appropriately chosen parts, while retaining a constant amount of entropy in the outputs). Let $A_0 \subseteq U^k \times \{0, 1\}^t$ be the set of views on which Alice outputs 0. Similarly define A_1, and also define the sets B_0 and B_1 for Bob. Let w^* be the weight function for $G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}$.

Then, the probability that Alice outputs 0 is $p_0^A = \sum_{a \in A_0} w^*(a)$. Similarly $p_1^A = \sum_{a \in A_1} w^*(a)$, and $p_0^B = \sum_{b \in B_0} w^*(b)$ and $p_1^B = \sum_{b \in B_1} w^*(b)$. W.l.o.g, assume that $p_0^A + p_0^B \le p_1^A + p_1^B$ (interchanging 0 and 1 if necessary). Then, let $S = A_0 \cup B_0$ and $\overline{S} = A_1 \cup B_1$. Since we required the output of at least one party to have constant entropy, it must be the case that $p_0^A + p_0^B \ge \alpha$ for some constant $\alpha > 0$. The probability that Alice and Bob disagree on their outputs, p^* is given by the weight of the edges that go across S and $\overline{S}$: i.e., $p^* = \sum_{x \in S, y \in \overline{S}} w^*(x, y)$. By definition of the Cheeger constant, we have

$$h(G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}) \le \frac{\sum_{x \in S, y \in \overline{S}} w^*(x, y)}{p_0^A + p_0^B} \le \frac{p^*}{\alpha}.$$

That is, $p^* \ge \alpha h(G^{\boxtimes k} \boxtimes {K_{2,2}}^{\boxtimes t}) \ge \alpha \frac{h^2(G)}{2}$. Since p_{UV} has zero common information, its bipartite characteristic graph G has a single connected component, and $h(G)$ is positive. Thus, we can set $\epsilon = \alpha \frac{h^2(G)}{2}$ to complete the proof of Theorem 2.

Remark. Gács and Körner [11] also considered the case when the setup distribution has non-zero common information. Our proof readily extends to this setting, showing that the entropy of a common output *conditioned on this common information* will have to be $o(1)$ (when the disagreement probability is required to be $o(1)$).