On the Classification of Finite Boolean Functions up to Fairness

Nikolaos Makriyannis

Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra, Spain
nikolaos.makriyannis@upf.edu

Abstract. Two parties, P_1 and P_2, wish to jointly compute some function $f(x, y)$ where P_1 only knows x, whereas P_2 only knows y. Furthermore, and most importantly, the parties wish to reveal only what the output suggests. Function f is said to be computable with *complete fairness* if there exists a protocol computing f such that whenever one of the parties obtains the correct output, then both of them do. The only protocol known to compute functions with complete fairness is the one of Gordon et al (STOC 2008). The functions in question are finite, Boolean, and the output is shared by both parties. The classification of such functions up to fairness may be a first step towards the classification of all functionalities up to fairness. Recently, Asharov (TCC 2014) identifies two families of functions that are computable with fairness using the protocol of Gordon et al and another family for which the protocol (potentially) falls short. Surprisingly, these families account for almost all finite Boolean functions. In this paper, we expand our understanding of what can be computed fairly with the protocol of Gordon et al. In particular, we fully describe which functions the protocol computes fairly and which it (potentially) does not. Furthermore, we present a new class of functions for which fair computation is outright impossible. Finally, we confirm and expand Asharov's observation regarding the fairness of finite Boolean functions: almost all functions $f : X \times Y \to \{0, 1\}$ for which $|X| \neq |Y|$ are fair, whereas almost all functions for which $|X| = |Y|$ are not.

Keywords: Complete Fairness, Secure Two-Party Computation

1 Introduction

Assume that k parties wish to jointly compute some functionality on k inputs where each party holds exactly one of the inputs. *Secure Multi-Party Computation* explains how, given certain security requirements and under certain assumptions, this task may be accomplished. For instance, using the standard convention, in the presence of an adversary that is malicious (may deviate from a protocol arbitrarily), computationally bounded, and only corrupts a strict minority of parties, it is well known [9] that for any functionality there exist protocols that guarantee at the same time, *privacy* - parties learn only what their (personal) inputs/outputs suggest, *correctness* - parties' outputs are distributed according to the prescribed functionality, *independence of inputs* - no party can choose his input as a function of another party's input, *complete fairness* - if a certain party learns the correct output then all of them do. On the other hand, if the adversary corrupts more than half of the parties, then protocols are known to exist that satisfy the first three requirements but not the last. Whether or not there exist protocols that guarantee complete fairness (*fairness* for short) is an open problem. In fact, the problem remains open even for the case where only two parties are involved and the functionality in question is finite, Boolean and deterministic. The aim of the present paper is to shed some light on this particular case: fairness of finite Boolean functions in 2PC.

It was long thought that non-trivial functions are *not* computable with fairness. Cleve's seminal paper [8], while not explicitly about fairness, has some clear implications regarding the topic in question. Loosely speaking, any function that can be used for coin-tossing is *not* computable with fairness. This includes, among others, the exclusive-or operation XOR : $(x, y) \mapsto x \oplus y$. Arguably, the implications of Cleve's paper may have deterred others from pursuing the question.

In a surprising turn of events, Gordon et al [10] showed that the folklore is false. In particular, all finite Boolean functions f that *do not* contain embedded XORs (i.e. $\exists x_1, y_1, x_2, y_2$ such that $f(x_1, y_1) = f(x_2, y_2) \neq f(x_1, y_2) = f(x_2, y_1)$) are computable with fairness. These functions roughly correspond to the Millionaire's Problem (i.e. $f(x, y) = 0 \Leftrightarrow x \geq y$). The authors also show that even certain XOR-embedded functions are fair. Using the real/ideal world paradigm, they design a protocol (that we henceforth refer to as the GHKL-protocol) and show that for certain functions, any real world adversary can be simulated in the ideal model. Thus proving that in an execution of the GHKL-protocol, the adversary cannot gain any advantage in learning the output over the honest party. On the other hand, in [5], Asharov et al give a complete characterization of finite Boolean functions that are not fair due to the coin-tossing criterion. In other words, there exists a class of functions (*strictly balanced* functions) that cannot be computed fairly, since the realisation of any such function yields a completely fair coin-toss. What's more, they show that functions that are not in that class *cannot* be reduced to the coin-tossing problem, meaning that any new negative results with respect to fairness must rely on different criteria.

Another major advancement towards characterizing all finite Boolean functions up to fairness appears recently in [3]. The author shows that there exists an extensive amount of functions that can be computed fairly using a slightly generalised version of the GHKL-protocol. Thus proving that these functions are inherently fair. Surprisingly, the author shows that taking a random function where players have input domains of different size will result, with overwhelming probability, in a fair function. On the flip side, Asharov shows that the GHKL-protocol does not account for all functions whose fairness remains unknown (In fact, almost all functions $f : X \times Y \to \{0, 1\}$ for which $|X| = |Y|$). These functions are not reducible to coin-tossing, nor are they computable with fairness using the GHKL-protocol, leaving the problem of determining whether they are fair or not wide open. Finally, the author also considers asymmetric and non-binary functions.

Preventing one party from gaining an advantage over the other seems like a desirable security requirement for 2PC. Fairness guarantees exactly that. In addition, apart from its obvious appeal as a security requirement, fairness is important to the theoretical foundation of Multi-Party Computation. The characterization of finite Boolean functions that are computable with fairness may be a first step towards the classification of all functionalities up to fairness.

Our Results. In the present paper, we take another step towards characterizing finite Boolean functions up to fairness. Ideally, we would like to find a universal criterion to determine whether a *deterministic single-output Boolean function* is computable with fairness. We remind the reader that certain functions are known to be fair (using the GHKL-protocol) and certain functions are not (due to the coin-tossing criterion). We contribute to both of these fronts. First, we give a definite answer to the following question: Given a finite Boolean function $f : X \times Y \to \{0,1\}$, is f computable with fairness using the GHKL-protocol? In [3], Asharov answers the question for three particular families of functions. We settle the question using linear algebra. Next, we expand our knowledge of functions which are provably unfair. We show that certain functions are reducible to the sampling problem: P_1 and P_2 generate separate random bits according to some joint probability distribution. The sampling problem is a natural generalisation of coin-tossing and was shown *not* to be computable with fairness by Agrawal and Prabhakaran [2]. Finally, by using the same argument, we confirm and expand Asharov's observation regarding the fairness of finite Boolean functions: *almost all functions $f : X \times Y \to \{0,1\}$ for which $|X| \neq |Y|$ are fair, whereas almost all functions for which $|X| = |Y|$ are not.*

Outline of the Paper. The next section contains notation and definitions. In particular, we introduce secure computation and the ideal model paradigm. In Section 3 we give an informal description of the GHKL-protocol (in its generalised version proposed by Asharov [3]) and a brief overview of the simulation strategy. We conclude with a sufficient criterion for fairness, as it appears in [10]. Our contributions begin in Section 4 where we find an equivalent criterion involving the linear dependence of the columns of a certain associated matrix. Section 5 contains the classification of all functions with respect to these criteria. Finally, the last section contains our negative result. We present a class of functions for which fair computation is impossible.

2 Preliminaries

We begin with notation and definitions. Throughout the paper we focus on functions of the form $f : X \times Y \to \{0,1\}$, where X and Y are finite and ordered sets, that is $X = \{x_1, \ldots, x_\ell\}$ and $Y = \{y_1, \ldots, y_k\}$. To every such function we associate a matrix $M \in \mathbb{R}^{\ell \times k}$, where $M_{i,j} = f(x_i, y_j)$. The ith row and jth column of M will be denoted row_i and col_j, respectively. Thus,

$$M = \begin{pmatrix} \mathsf{row}_1 \\ \vdots \\ \mathsf{row}_\ell \end{pmatrix} = \begin{pmatrix} \mathsf{col}_1 \,\big|\, \ldots \,\big|\, \mathsf{col}_k \end{pmatrix}.$$

More generally, we use capital letters for matrices $(M, P, \ldots)$. We denote the kernel and image of M by $\ker(M)$ and $\mathrm{im}(M)$ respectively, and its transpose by M^T. Vectors will be denoted by lower case letters $(u, v, \ldots)$ or bold letters $(\mathbf{p}, \mathbf{q}, \mathbf{x}, \ldots)$. In particular, $\mathbf{1}_\ell$ and $\mathbf{0}_\ell$ represent the all-1 and all-0 vector respectively, these elements will also be called monochromatic. We say that $\mathbf{p}^T = (p_1, \ldots, p_k)$ is a probability vector if $\sum_i p_i = 1$ and $p_j \geq 0$, for every j.

Let $\mathbb{R}^k$ denote the real vector space of dimension k and let $\langle u \,|\, v \rangle = u^T v$ denote the standard inner product. As usual, two vectors are orthogonal if their inner product is equal to 0. Similarly, two subsets of $\mathbb{R}^k$ are orthogonal if every element of the first is orthogonal to every element of the second. Recall the fundamental theorem of linear algebra: for a given matrix $M \in \mathbb{R}^{\ell \times k}$, $\ker(M)$ and $\mathrm{im}(M^T)$ are orthogonal and their sum spans the entire space. Furthermore, let $I^{\ker(M)}$ denote the *orthogonal* projection of $\mathbb{R}^k$ onto $\ker(M)$ i.e. $I^{\ker(M)}$ is the unique linear map from $\mathbb{R}^k$ onto itself that annihilates the elements of $\mathrm{im}(M^T)$ and corresponds to the identity when restricted to $\ker(M)$. Finally, given a vector space $\mathcal{V} \subseteq \mathbb{R}^k$ and a $k \times k$ matrix P, we say that P defines an endomorphism of $\mathcal{V}$ if $Pv \in \mathcal{V}$, for every $v \in \mathcal{V}$.

2.1 Two-Party Computation

Let $n \in \mathbb{N}$ denote the security parameter. A function $\mu(\cdot)$ is *negligible* if it vanishes faster than any (positive) inverse-polynomial. A distribution ensemble $X = \{X(a, n)\}_{a \in \Delta_n, n \in \mathbb{N}}$ is an infinite sequence of random variables indexed by Δ_n and $\mathbb{N}$. Two distribution ensembles, X and Y, are *computationally indistinguishable* if for every non-uniform polynomial-time algorithm D, there exists a negligible function μ such that for every a and n

$$|\Pr[D(X(a, n)) = 1] - \Pr[D(Y(a, n)) = 1]| \leq \mu(n).$$

Furthermore, we say that X and Y are *statistically close* if for all a and n, the following sum is upper-bounded by a negligible function in n:

$$\frac{1}{2} \cdot \sum_s |\Pr[X(a, n) = s] - \Pr[Y(a, n) = s]|,$$

where s ranges over the support of either $X(a, n)$ or $Y(a, n)$. We write $X \overset{c}{\equiv} Y$ when the ensembles are computationally indistinguishable and $X \overset{s}{\equiv} Y$ when they are statistically close.

Let P_1, P_2 denote the parties. A *two-party functionality* $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, is a sequence of random processes such that each f_n maps pairs of inputs (one for each party) to pairs of random variables (one for each party). The domain of f_n is denoted $X_n \times Y_n$ and the output (f_n^1, f_n^2). A *two-party protocol* π for computing a functionality $\mathcal{F}$, is a polynomial-time protocol such that on inputs $x \in X_n$ and $y \in Y_n$, the joint distribution of the outputs of any honest execution of π is statistically close to $f_n(x,y) = (f_n^1(x,y), f_n^2(x,y))$.

The Adversary. Following the usual convention, we introduce an adversary $\mathcal{A}$ given auxiliary input z corrupting one of the parties. The adversary is assumed to be malicious and computationally bounded. For a protocol π computing $\mathcal{F}$, let $(\mathrm{Out}^{\mathsf{Real}}_{\mathcal{A}(z),\pi}, \mathrm{View}^{\mathsf{Real}}_{\mathcal{A}(z),\pi})(x,y,n)$ denote the joint distribution of the honest party's output and the adversary's view during an execution of π, where x and y are the prescribed inputs and n is the security parameter.

2.2 The Ideal Model Paradigm

Let $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$ be a two-party functionality and let π be a protocol for computing $\mathcal{F}$. Further assume that an adversary is corrupting one of the parties. Security in two-party computation is defined via an *ideal model*. Namely, we assume that parties have access to a trusted party that performs the computation for them, and we attempt to show that protocol π *emulates* this ideal scenario. We will now describe three ideal models and give definitions of security for each of them. The first corresponds to complete fairness which is the topic of the present paper. Our goal is to show there exist (or not) protocols which are secure with respect to this model. The other two are stepping stones which will help us achieve that goal. We now describe the *ideal model with complete fairness* (Figure 1).

Inputs: P_1 holds 1^n and $x \in X_n$, P_2 holds 1^n and $y \in Y_n$. The adversary is given an auxiliary input $z \in \{0,1\}^*$.

Parties send inputs: The honest party sends his input to $\mathcal{T}$, the corrupted party sends a value of the adversary's choice. Write (x', y') for the pair of inputs received by $\mathcal{T}$.

Trusted party performs computation: If either x' or y' are not in the appropriate domain, then $\mathcal{T}$ reassigns the aberrant input to some default value. Write (x', y') for the pair of inputs after (possible) reassignment. The trusted party then chooses a random string r and computes $(f_n^1, f_n^2) = f_n(x', y'; r)$.

Trusted party sends outputs: Party P_1 receives f_n^1, P_2 receives f_n^2.

Outputs: The honest party outputs whatever $\mathcal{T}$ sent him, the corrupted party outputs nothing and the adversary outputs a probabilistic polynomial-time function of its view.

Fig. 1: Ideal model with complete fairness

Let $\mathcal{S}$ be an adversary given auxiliary input z corrupting one of the parties. Write $(\mathrm{Out}^{\mathsf{IM\ fair}}_{\mathcal{S}(z),f}, \mathrm{View}^{\mathsf{IM\ fair}}_{\mathcal{S}(z),f})(x, y, n)$ for the joint distribution of the honest party's output and the adversary's view, where x and y are the prescribed inputs and n is the security parameter. We now define security with respect to the ideal model with complete fairness.

Definition 2.1. *Let π be a protocol for computing $\mathcal{F}$. We say that π* securely *computes $\mathcal{F}$* with complete fairness *(or $\mathcal{F}$ is fair for short) if for every non-uniform polynomial time adversary $\mathcal{A}$ in the real model, there exists a non-uniform polynomial time adversary $\mathcal{S}$ in the ideal model such that*

$$\left\{\left(\mathrm{Out}^{\mathsf{Real}}_{\mathcal{A}(z),\pi}, \mathrm{View}^{\mathsf{Real}}_{\mathcal{A}(z),\pi}\right)(x, y, n)\right\}_{\substack{(x,y)\in X_n\times Y_n,\\ z\in\{0,1\}^*, n\in\mathbb{N}}} \overset{c}{\equiv} \left\{\left(\mathrm{Out}^{\mathsf{IM\ fair}}_{\mathcal{S}(z),\mathcal{F}}, \mathrm{View}^{\mathsf{IM\ fair}}_{\mathcal{S}(z),\mathcal{F}}\right)(x, y, n)\right\}_{\substack{(x,y)\in X_n\times Y_n,\\ z\in\{0,1\}^*, n\in\mathbb{N}}}.$$

In effect, showing that the above distribution ensembles are computationally indistinguishable implies that, in the plain model, the information acquired by the adversary together with his influence over the honest party's output is no worse than what can be achieved in an idealized situation. Moving on, we also define the *ideal model with abort* (Figure 2) and define security with respect to this model. We note that all functionalities are securely computable with abort (Definition 2.2). Once again, let $\mathcal{S}$ be an adversary given auxiliary input z corrupting one of the parties, and write $(\mathrm{Out}^{\mathsf{IM\ abort}}_{\mathcal{S}(z),f}, \mathrm{View}^{\mathsf{IM\ abort}}_{\mathcal{S}(z),f})(x, y, n)$ for the joint distribution of the honest party's output and the adversary's view, where x and y are the prescribed inputs and n is the security parameter.

Definition 2.2. *Let π be a protocol for computing $\mathcal{F}$. We say that π* securely *computes $\mathcal{F}$* with abort *if for every non-uniform polynomial time adversary $\mathcal{A}$ in the real model, there exists a non-uniform polynomial time adversary $\mathcal{S}$ in the ideal model such that*

$$\left\{\left(\mathrm{Out}^{\mathsf{Real}}_{\mathcal{A}(z),\pi}, \mathrm{View}^{\mathsf{Real}}_{\mathcal{A}(z),\pi}\right)(x, y, n)\right\}_{\substack{(x,y)\in X_n\times Y_n,\\ z\in\{0,1\}^*, n\in\mathbb{N}}} \overset{c}{\equiv} \left\{\left(\mathrm{Out}^{\mathsf{IM\ abort}}_{\mathcal{S}(z),\mathcal{F}}, \mathrm{View}^{\mathsf{IM\ abort}}_{\mathcal{S}(z),\mathcal{F}}\right)(x, y, n)\right\}_{\substack{(x,y)\in X_n\times Y_n,\\ z\in\{0,1\}^*, n\in\mathbb{N}}}.$$

The Hybrid Model. In secure computation, the hybrid model is a tool that allows us to break some cryptographic task into subtask and, assuming these subtasks can be implemented securely, prove the security of the overlying task.

Inputs: P_1 holds 1^n and $x \in X_n$, P_2 holds 1^n and $y \in Y_n$. The adversary is given an auxiliary input $z \in \{0,1\}^*$.

Parties send inputs: The honest party sends his input to $\mathcal{T}$, the corrupted party sends a value of the adversary's choice. Write (x', y') for the pair of inputs received by $\mathcal{T}$.

Trusted party performs computation: If either x' or y' are not in the appropriate domain, then $\mathcal{T}$ reassigns the aberrant input to some default value. Write (x', y') for the pair of inputs after (possible) reassignment. The trusted party then chooses a random string r and computes $(f_n^1, f_n^2) = f_n(x', y'; r)$.

Trusted party sends outputs: Corrupted party P_i receives f_n^i from the trusted party. The adversary then decides either to *abort* or *continue*. In the first case, $\mathcal{T}$ sends $\perp$ to the honest party. In the second case, P_{3-i} receives f_n^{3-i}.

Outputs: The honest party outputs whatever $\mathcal{T}$ sent him, the corrupted party outputs nothing and the adversary outputs a probabilistic polynomial-time function of its view.

Fig. 2: Ideal model with abort

Let $\mathcal{F}_1, \ldots, \mathcal{F}_m$ and $\mathcal{F}$ be two-party functionalities. A protocol π for computing $\mathcal{F}$ in the $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$-hybrid model consists in an protocol computing $\mathcal{F}$ that proceed in rounds such that at any given round:

- Parties exchange information as in the real model *or*
- Parties invoke a trusted party computing $\mathcal{F}_i$ according to a specified ideal model.

We say that π computes $\mathcal{F}$ securely in the $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$-hybrid model if for any adversary corrupting one of the parties in the hybrid model, there exists a simulator $\mathcal{S}$ in the ideal model such that the joint distribution of the adversary's view and honest party's output is indistinguishable in both worlds. Hence, applying the composition theorem of [7], and assuming there exists secure protocols $\rho_1, \ldots, \rho_m$ computing $\mathcal{F}_1, \ldots, \mathcal{F}_m$, protocol $\pi^{\rho_1 \cdots \rho_m}$ securely computes $\mathcal{F}$, where $\pi^{\rho_1 \cdots \rho_m}$ is obtained by replacing ideal calls to the trusted party with the appropriate protocols.

Reactive Functionalities. A reactive functionality $\mathcal{G}$ is a cryptographic task that proceeds in several phases, where the input of one phase may depend on the output of previous phases. With respect to the ideal model with abort, every reactive functionality can be computed securely. Let π be a protocol for computing $\mathcal{F}$ that does not involve any direct exchange of information between the parties. Rather, at any given round, parties make a single call to a trusted party computing $\mathcal{G}$ with abort. The composition theorem still holds, that is if the joint distribution of the adversary's view and honest party's output is indistinguishable in the $\mathcal{G}$-hybrid and ideal model *with complete fairness*, then protocol π^ρ securely computes $\mathcal{F}$ with complete fairness, where ρ securely computes $\mathcal{G}$ with abort. In fact, following Asharov [3], the GHKL-protocol in the next section is defined by means of a reactive functionality.

3 Computing Fair Functions

In this section, we focus on the GHKL-protocol [10], the only protocol that provably computes (certain) functions with complete fairness. We follow the description and generalisation proposed by Asharov in [3]. The protocol is defined in the hybrid model where parties have access to a trusted party computing a certain reactive functionality with abort. We give an informal description of the protocol and a brief overview of the simulation strategy. The complete description and simulation strategy can be found in [11] and [4], in their respective appendices.

3.1 Informal Description of the Generalised GHKL-Protocol

Let $f : X \times Y \to \{0,1\}$ be a finite Boolean function, suppose that the prescribed inputs of P_1, P_2 are x and y respectively. The GHKL-protocol for computing f is parametrized by three values: a real number $\alpha \in (0,1)$, a probability vector $\mathbf{p} \in \mathbb{R}^\ell$ and the security parameter n. Prior to the execution of the protocol, the number of rounds is fixed at $r = \alpha^{-1} \cdot \omega(\ln(n))$.

- **Compute Backup Outputs:** P_1 chooses $\widetilde{y} \in Y$ according to the uniform distribution and computes $a_0 = f(x_i, \widetilde{y})$, P_2 chooses $\widetilde{x} \in X$ according to distribution $\mathbf{p}$ and computes $b_0 = f(\widetilde{x}, y_j)$.
- **Preliminary Phase:** Parties are instructed to send their inputs to the trusted party. If either input is not in the correct domain, the trusted party responds with an abort symbol to both parties and halts. Otherwise, write (x', y') for the pair of inputs received by the trusted party. The trusted party is instructed to choose an integer i^* according to the geometric distribution with parameter α, and constructs the following bits:
  - For $i = 1 \dots i^* - 1$, set $a_i = f(x', \widetilde{y}^{(i)})$ and $b_i = f(\widetilde{x}^{(i)}, y')$ where $\widetilde{y}^{(i)}$ and $\widetilde{x}^{(i)}$ are chosen according to the uniform distribution and distribution $\mathbf{p}$, respectively.
  - For $i = i^* \dots r$, set $a_i = b_i = f(x', y')$.
- **Online Phase:** For $i = 1 \dots r$
  1. P_2 sends proceed to the trusted party, P_1 then receives a_i.
  2. P_1 sends proceed to the trusted party, P_2 then receives b_i.

  If either party sends abort, then both receive $\perp$ and the trusted party halts.
- **Outputs:** Parties are instructed to output the last a_i/b_i they successfully constructed/received.

3.2 Security

In order to prove security, we need to show that any adversary in the hybrid model can be simulated in the ideal model with complete fairness. On an intuitive level, note that a corrupted P_2 cannot affect the protocol's fairness. No matter what, P_1 always receives the correct input first. In fact, a corrupted P_2 can be simulated regardless of the parameters, for *any* function (see [10] and [3]). On the other hand, precisely because P_1 receives the correct input first, an adversary $\mathcal{A}$ controlling P_1 can potentially affect the protocol's fairness. We focus on the latter case and give an incomplete description of the simulator $\mathcal{S}$ *in the ideal model.*

Suppose that $\mathcal{A}$ hands $x \in X$ to $\mathcal{S}$ for the computation of f. The simulator chooses a value i^* according to the geometric distribution with parameter α. Now, for $i = 1 \dots i^* - 1$, simulator $\mathcal{S}$ hands $a_i = f(x, \widetilde{y}^{(i)})$ to $\mathcal{A}$, where $\widetilde{y}^{(i)}$ is chosen according to the uniform distribution. If at any point the adversary decides to abort, $\mathcal{S}$ chooses x' according to probability distribution $\mathbf{x}_x^{(a_i)}$ (that we define below), sends x' to the trusted party, outputs whatever $\mathcal{A}$ outputs, and halts. Otherwise, at $i = i^*$, the simulator sends x to the trusted party and receives $a_{\text{out}} = f(x, y)$ which it hands to $\mathcal{A}$. If $\mathcal{A}$ aborts, then $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs, and halts. Finally, for $i = i^* + 1 \dots r$, the simulator hands a_{out} to $\mathcal{A}$. Once again, if $\mathcal{A}$ aborts, then $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs, and halts.

Recall that the protocol is secure if the joint distributions of the adversary's view and the honest party output in the ideal and hybrid model are computationally indistinguishable. The simulation strategy boils down to the existence of $\mathbf{x}_x^{(a)}$ that will do the trick. Using the fact that i^* is chosen according to a geometric distribution, it can be shown that the above simulation works if for all $a \in \{0, 1\}$, for all $x \in \{x_1, \dots, x_\ell\}$,

$$M^T \cdot \mathbf{x}_x^{(a)} = \mathbf{c}_x^{(a)},$$

where $M_{i,j} = f(x_i, y_j)$,

$$\mathbf{c}_x^{(0)}(j) \stackrel{\text{def}}{=} \begin{cases} p_{y_j} & \text{if } f(x, y_j) = 1 \\ \frac{\alpha \cdot p_{y_j}}{(1-\alpha)\cdot(1-p_x)} + p_{y_j} & \text{otherwise} \end{cases} ,$$

$$\mathbf{c}_x^{(1)}(j) \stackrel{\text{def}}{=} \begin{cases} \frac{\alpha \cdot (p_{y_j} - 1)}{(1-\alpha)\cdot p_x} + p_{y_j} & \text{if } f(x, y_j) = 1 \\ p_{y_j} & \text{otherwise} \end{cases} ,$$

and

$$(p_{y_1}, \dots, p_{y_k}) \stackrel{\text{def}}{=} \mathbf{p}^T \cdot M, \qquad (p_{x_1}, \dots, p_{x_\ell})^T \stackrel{\text{def}}{=} M \cdot (1/k, \dots, 1/k)^T.$$

Theorem 3.1. *Using the notation above, if for some probability vector* $\mathbf{p}$ *and* $\alpha \in (0, 1)$, *and for all* $i \in \{1, \dots, \ell\}$, *there exist probability vectors* $\mathbf{x}_{x_i}^{(0)}, \mathbf{x}_{x_i}^{(1)} \in \mathbb{R}^\ell$ *such that*

$$M^T \cdot \mathbf{x}_{x_i}^{(a)} = \mathbf{c}_{x_i}^{(a)},$$

then f *is computable with complete fairness using the GHKL-protocol.*

Arguably, a function that does not satisfy the criteria of the above theorem may still be computable with fairness using the GHKL-protocol. Of course, such a claim must be accompanied by a new valid simulation strategy, since the one described above will not work. With that in mind, and for the rest of the paper, a function will be said to be *GHKL-fair* if it satisfies the hypothesis of Theorem 3.1. In other words, a function is GHKL-fair if it is computable with fairness using the GHKL-protocol *and* the particular simulation strategy described above proves it.

4 Equivalent Conditions for GHKL-Fairness

The conditions of Theorem 3.1 depend heavily on parameters of the protocol, namely α, $\{p_{x_i}\}_i$ and $\{p_{y_j}\}_y$. A universal criterion for fairness would only depend on the function itself. Failing that, we aim to find equivalent conditions that are "easier" to verify. In this section, starting with the equations of Theorem 3.1, we present a new set of conditions that do not depend on α, nor $\{p_{x_i}\}_i$, but only $\{p_{y_j}\}_j$ and the function itself. Let f be a finite boolean function, and assume that f is GHKL-fair i.e. for some $\alpha \in (0,1)$ and probability vector $\mathbf{p}$, and all $i \in \{1,\dots,\ell\}$, there exist probability vectors $\mathbf{x}_{x_i}^{(0)}, \mathbf{x}_{x_i}^{(1)} \in \mathbb{R}^\ell$ such that $M^T \cdot \mathbf{x}_{x_i}^{(a)} = \mathbf{c}_{x_i}^{(a)}$. Define

$$\widetilde{\mathbf{x}}_{x_i}^{(0)} \overset{\text{def}}{=} \frac{(1-\alpha)(1-p_{x_i})}{\alpha}\left(\mathbf{x}_{x_i}^{(0)} - \mathbf{p}\right) \ ,$$
$$\widetilde{\mathbf{x}}_{x_i}^{(1)} \overset{\text{def}}{=} \frac{(1-\alpha)p_{x_i}}{\alpha}\left(\mathbf{x}_{x_i}^{(1)} - \mathbf{p}\right) \ ,$$

and note that

$$M^T \cdot \widetilde{\mathbf{x}}_{x_i}^{(a)} = \widetilde{\mathbf{c}}_{x_i}^{(a)} \ , \tag{1}$$

where

$$\widetilde{\mathbf{c}}_{x_i}^{(0)}(j) = \begin{cases} 0 & \text{if } f(x_i,y_j)=1 \\ p_{y_j} & \text{otherwise} \end{cases} , \qquad \widetilde{\mathbf{c}}_{x_i}^{(1)}(j) = \begin{cases} p_{y_j}-1 & \text{if } f(x_i,y_j)=1 \\ 0 & \text{otherwise} \end{cases} .$$

Furthermore,

$$\sum_j \widetilde{\mathbf{x}}_{x_i}^{(a)}(j) = 0 \ , \tag{2}$$
$$\forall j,\ \widetilde{\mathbf{x}}_{x_i}^{(0)}(j) \in \frac{(1-\alpha)(1-p_{x_i})}{\alpha}\cdot[-\mathbf{p}(j), 1-\mathbf{p}(j)] \ , \tag{3}$$
$$\forall j,\ \widetilde{\mathbf{x}}_{x_i}^{(1)}(j) \in \frac{(1-\alpha)p_{x_i}}{\alpha}\cdot[-\mathbf{p}(j), 1-\mathbf{p}(j)] \ . \tag{4}$$

Conversely, if for some probability vector $\mathbf{p}$ and $\alpha \in (0,1)$, and for all i, there exist $\widetilde{\mathbf{x}}_{x_i}^{(0)}$ and $\widetilde{\mathbf{x}}_{x_i}^{(1)}$ satisfying relations (1) to (4), then we can construct $\mathbf{x}_{x_i}^{(0)}$ and $\mathbf{x}_{x_i}^{(1)}$ satisfying Theorem 3.1. Specifically,

$$\mathbf{x}_{x_i}^{(0)} = \begin{cases} \frac{\alpha\cdot\widetilde{\mathbf{x}}_{x_i}^{(0)}}{(1-\alpha)(1-p_{x_i})} + \mathbf{p} & \text{if } p_{x_i} \neq 1 \\ \mathbf{p} & \text{otherwise} \end{cases} , \qquad \mathbf{x}_{x_i}^{(1)} = \begin{cases} \frac{\alpha\cdot\widetilde{\mathbf{x}}_{x_i}^{(1)}}{(1-\alpha)p_{x_i}} + \mathbf{p} & \text{if } p_{x_i} \neq 0 \\ \mathbf{p} & \text{otherwise} \end{cases} .$$

Simplifying Assumption. Probability vector $\mathbf{p}$ will be considered without zero-components i.e. $\forall j,\ \mathbf{p}(j) \neq 0$. Note that, in this case, the intervals defined in (3) and (4) grow arbitrarily as α tends to zero. Thus, we are only required to verify relations (1) and (2). On the other hand, we claim that this simplifying assumption does not incur any loss of generality: If there exists a "suitable" $\mathbf{p}$ with zero-entries, then we can construct another "suitable" probability vector that is strictly positive. For further details, see Appendix A. To sum up, we conclude with the following theorem.

Theorem 4.1. *Using the notation above, function f is GHKL-fair if and only if for some strictly positive $\mathbf{p}$, and all $i \in \{1, \ldots, \ell\}$, there exist $\widetilde{\mathbf{x}}_{x_i}^{(0)}$ and $\widetilde{\mathbf{x}}_{x_i}^{(1)}$ such that*

$$M^T \cdot \widetilde{\mathbf{x}}_{x_i}^{(a)} = \widetilde{\mathbf{c}}_{x_i}^{(a)} \qquad \textit{and} \qquad \sum_j \widetilde{\mathbf{x}}_{x_i}^{(a)}(j) = 0 \ .$$

Recall that $\mathbf{p}^T M = (p_{y_1}, \ldots, p_{y_k})$. Define $P(M, \mathbf{p}) = \mathsf{diag}(p_{y_1}, \ldots, p_{y_k})$ i.e. the diagonal matrix whose components are the p_{y_i}. If no confusion arises, $P(M, \mathbf{p})$ will be denoted P. In Figure 3, we express the conditions of Theorem 4.1 with respect to matrix P. In effect, function f is GHKL-fair if $(\mathbf{1}_k^T - \mathsf{row}_i)P$ and

For all $i \in \{1, \ldots, \ell\}$, for all $a \in \{0, 1\}$, there exists $(\mu_{i,1}^{(a)}, \ldots, \mu_{i,\ell}^{(a)})$ that satisfies

1. $\sum_j \mu_{i,j}^{(a)} = 0$,
2. $(\mu_{i,1}^{(0)}, \ldots, \mu_{i,\ell}^{(0)})M = (\mathbf{1}_k^T - \mathsf{row}_i)P$,
3. $(\mu_{i,1}^{(1)}, \ldots, \mu_{i,\ell}^{(1)})M = \mathsf{row}_i(P - \mathrm{Id}_k)$.

Fig. 3: GHKL-fairness conditions

$\mathsf{row}_i(P - \mathrm{Id}_k)$ belong to the image of M^T and admit a suitable pre-image. Let $\mathcal{V} = \{v \in \mathbb{R}^\ell \mid \sum_i v_i = 0\}$. By considering an appropriate basis of $\mathcal{V}$, we note that the image of $\mathcal{V}$ by M^T corresponds exactly to the image of M'^T, where

$$M' = \begin{pmatrix} \mathsf{row}_1 \\ \mathsf{row}_1 \\ \vdots \\ \mathsf{row}_1 \end{pmatrix} - \begin{pmatrix} \mathsf{row}_2 \\ \mathsf{row}_3 \\ \vdots \\ \mathsf{row}_\ell \end{pmatrix} .$$

Furthermore, a quick analysis shows that $\ker(M') = \left\{ v \in \mathbb{R}^k \,\middle|\, Mv = (\delta, \ldots, \delta)^T \right\}$. Going back to the conditions of Figure 3, function f is GHKL-fair if $(\mathbf{1}_k^T - \mathsf{row}_i)P$ and $\mathsf{row}_i(P - \mathrm{Id}_k)$ belong to the image of M'^T. By orthogonality,

$$\forall v \in \ker(M'), \qquad \begin{cases} (\mathbf{1}_k^T - \mathsf{row}_i)Pv & = 0 \\ \mathsf{row}_i(P - \mathrm{Id}_k)v & = 0 \end{cases} ,$$

both of which boil down to $\mathsf{row}_i Pv = \delta$. By letting i vary, we deduce the following Proposition.

Proposition 4.2. *Function f is GHKL-fair if and only if there exists $\mathbf{p}$ such that $Mv = MPv$, for every $v \in \ker(M')$.*

Already, we see that for a given $\mathbf{p}$, verifying that a function f is GHKL-fair is very easy. Take a basis of $\ker(M')$ and check that $M(\mathrm{Id}_k - P)v = \mathbf{0}_\ell$, for every element of the basis. Of course finding $\mathbf{p}$ is a non-trivial task, and we look into that later on. For the remainder of the section, we show that certain functions are not viable candidates for GHKL-fairness, regardless of the value of $\mathbf{p}$.

Lemma 4.3. *Assuming $Mv = MPv$, for every $v \in \ker(M')$, matrix P defines an endomorphism of $\ker(M)$ as well as $\mathrm{im}(M^T)$.*

Proof. First of all, if $v \in \ker(M)$, then $MPv = Mv = 0$ and thus $Pv \in \ker(M)$. On the other hand, let $u \in \mathrm{im}(M^T)$ and $v \in \ker(M)$. Since P is diagonal, we deduce that $\langle Pu \,|\, v\rangle = \langle u \,|\, Pv\rangle$, and thus $\langle Pu \,|\, v\rangle = 0$. Since v is arbitrary, $Pu \in \mathrm{im}(M^T)$. □

Lemma 4.4. *Suppose that $Mv = MPv$, for every $v \in \ker(M')$. Furthermore, assuming it exists, let $b \in \mathbb{R}^k$ denote the unique pre-image of $\mathbf{1}_\ell$ by M that is orthogonal to $\ker(M)$. Then, $b(j) \neq 0$ implies $\mathsf{col}_j = \mathbf{1}_\ell$.*

Proof. We know that $(\mathrm{Id}_k - P)b \in \ker(M)$. Now, since P is an endomorphism of $\mathrm{im}(M^T)$, and thus $(\mathrm{Id}_k - P)b \in \mathrm{im}(M^T)$, we deduce that $(\mathrm{Id}_k - P)b = 0$. Consequently $b(j) \neq 0$ implies $P_{j,j} = 1$, and since $P_{j,j} = \mathbf{p}^T\mathsf{col}_j$ and $\mathbf{p} > \mathbf{0}_\ell$, we conclude that $\mathsf{col}_j = \mathbf{1}_\ell$. □

Theorem 4.5. *Using the notation above, f is GHKL-fair if and only if*

1. *no linear combination of the non-monochromatic columns of M yields $\mathbf{1}_\ell$,*
2. *there exists a strictly positive $\mathbf{p}$ such that matrix P defines an endomorphism of $\ker(M)$.*

Proof. The second item is necessary because of Lemma 4.3. For the first item, suppose without loss of generality that the first k' columns contain $\mathbf{1}_\ell$ in their linear span, and none of them is monochromatic. We show that $b(j) \neq 0$, for some $j \in \{1, \dots, k'\}$, in contradiction with Lemma 4.4. Let $v = (v_1, \dots, v_{k'}, 0, \dots, 0)$, such that $Mv = \mathbf{1}_\ell$. We know that $v = b + v'$, where $v' \in \ker(M)$. Consequently, $\langle v \,|\, b\rangle = \langle b \,|\, b\rangle > 0$, and we conclude that at least one of the first k' entries of b is non-zero.

Conversely, we show that the two conditions imply Proposition 4.2. First, since P is an endomorphism of $\ker(M)$, then $0 = Mv = MPv$, for all $v \in \ker(M)$. It remains to show that $Mb = MPb$. Since the non-monochromatic columns of M do not span the all-1 vector, we deduce that $b(j) \neq 0$ if and only if $\mathsf{col}_i = \mathbf{1}_\ell$, and thus $P_{i,i} = 1$. Consequently, $Pb = b$, and thus f is GHKL-fair. It can also be shown that neither condition is sufficient on its own. □

5 Classification of GHKL-Fair Functions

We now present a classification of finite Boolean functions with respect to GHKL-fairness. We subdivide functions into four families, three of which can be accounted for almost immediately. First, we prove a couple of claims.

Proposition 5.1. *If f is GHKL-fair, then the same is true of* $1 - f$.

Proof. We prove the claim by applying Proposition 4.2. Assuming that M is the associated matrix of f, write $\overline{M}$ for the associated matrix of $1 - f$. Using the same $\mathbf{p}$, assuming that $Mv = MPv$, for every $v \in \ker(M')$, and noting that $\ker(\overline{M}') = \ker(M')$, we conclude that $\overline{M}v = \overline{M} \cdot P(\overline{M}, \mathbf{p}) \cdot v$, for every $v \in \ker(\overline{M}')$. □

Corollary 5.2. *Suppose that f is GHKL-fair. Further assume that the first k' columns of M are non-monochromatic. Then, $\sum_{i=1}^{k'} v_i \mathsf{col}_i = \mathbf{0}_\ell$ implies $\sum_{i=1}^{k'} v_i =$* 0.

Proof. If not, we obtain a contradiction with the above proposition and the first item of Theorem 4.5. □

Define $M_{0/1} \in \{0,1\}^{\ell \times k'}$ obtained from M by deleting all monochromatic columns. We distinguish 4 types of functions depending on the following properties of their associated matrix:

(Type 1) $\mathbf{1}_\ell \in \mathrm{im}(M_{0/1})$.
(Type 2) $\mathbf{1}_\ell \notin \mathrm{im}(M_{0/1})$ *and* $\ker(M_{0/1}) = \{\mathbf{0}_{k'}\}$.
(Type 3) $\mathbf{1}_\ell \notin \mathrm{im}(M_{0/1})$ *and* $\ker(M_{0/1}) \neq \{\mathbf{0}_{k'}\}$ *and* $\ker(M_{0/1}) \subseteq \{v \mid \sum_i v_i = 0\}$.
(Type 4) $\mathbf{1}_\ell \notin \mathrm{im}(M_{0/1})$ *and* $\ker(M_{0/1}) \neq \{\mathbf{0}_{k'}\}$ *and* $\ker(M_{0/1}) \not\subseteq \{v \mid \sum_i v_i = 0\}$.

Theorem 5.3. *Let f be a finite Boolean function.*

- *If f is of type 1 or 4, then it is not GHKL-fair.*
- *If f is of type 2, then it is GHKL-fair.*

Proof. Functions of type 1 do not satisfy the first item of Theorem 4.5. Similarly, functions of type 4 are not in accordance with Corollary 5.2. On the other hand, and by applying Theorem 4.5, we note that functions of type 2 and 3 do not contain $\mathbf{1}_\ell$ in the linear span of the non-monochromatic columns. It remains to show that there exists a strictly positive $\mathbf{p}$ such that matrix P defines an endomorphism of the kernel. If f is of type 2, then columns of M that are linearly dependent are all monochromatic. Knowing that $(p_{y_1}, \ldots, p_{y_k}) = \mathbf{p}^T M$, a quick analysis shows that matrix P defines an endomorphism of $\ker(M)$, for any $\mathbf{p}$. Finally, for functions of type 3, the existence of $\mathbf{p}$ depends on the linearly dependent columns of $M_{0/1}$, and so a general rule cannot be extrapolated at this point. □

The theorem above confirms the findings of Asharov. Namely, in [3], the author identifies two families of GHKL-fair functions, both of which are of type 2. On the flip side, the author finds a family of functions for which the simulation strategy we consider falls short. Indeed, these are type 1 functions.

5.1 Finding the Probability Vector

Theorem 5.3 does not account for functions of type 3. Indeed, the existence of $\mathbf{p}$ depends on the linearly dependent columns of $M_{0/1}$. In the remainder of this section, we show that the existence of $\mathbf{p}$ is subject to the resolution of a linear program. We note that the program is defined in the broadest possible terms, meaning that it's a "one size fits all" way to find $\mathbf{p}$. To illustrate, consider a matrix $M \in \{0,1\}^{\ell \times k}$ of type 2. Construct a new matrix by concatenating a column of M to itself. Now, M is of type 2 and thus GHKL-fair, whereas the other is of type 3. It is easy to see however that it is also GHKL-fair, for any $\mathbf{p}$. Moving on, recall that $I^{\ker(M)}$ denotes the orthogonal projection onto the kernel of M. Furthermore, for all i, let I^{row_i} denote the diagonal matrix $I^{\mathsf{row}_i}_{j,j} = \mathsf{row}_i(j)$.

Theorem 5.4. *Let f be a finite Boolean function of type 3. Function f is GHKL-fair if and only if*

$$\begin{pmatrix} I^{\ker(M)} \cdot I^{\mathsf{row}_1} \\ \vdots \\ I^{\ker(M)} \cdot I^{\mathsf{row}_\ell} \end{pmatrix} M^T \begin{pmatrix} r_1 \\ \vdots \\ r_\ell \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5}$$

admits a strictly positive solution.

Proof. Let $\mathbf{p} = (r_1, \ldots, r_\ell)^T$. First, note that $P = \sum_i r_i I^{\mathsf{row}_i}$, and that P defines an endomorphism of $\ker(M)$ if and only if

$$\forall j \in \{1, \ldots, \ell\}, \quad \left(\sum_i r_i I^{\mathsf{row}_i} \right) \mathsf{row}_j^T \in \mathrm{im}(M^T) \ . \tag{6}$$

Since the matrices are all diagonal, and thus commute, we deduce that

$$\begin{aligned} \left(\sum_i r_i I^{\mathsf{row}_i} \right) \mathsf{row}_j^T &= \left(\sum_i r_i I^{\mathsf{row}_i} \right) I^{\mathsf{row}_j} \cdot \mathbf{1}_\ell = I^{\mathsf{row}_j} \left(\sum_i r_i \mathsf{row}_i^T \right) \\ &= I^{\mathsf{row}_j} \left(\sum_i r_i \mathsf{row}_i^T \right) = I^{\mathsf{row}_j} \cdot M^T (r_1, \ldots, r_\ell)^T \ . \end{aligned}$$

Consequently, (6) holds if and only if (5) admits a strictly positive solution, in which case the probability vector is obtained by normalisation. □

5.2 Complete Fairness and GHKL-Fairness

We take a moment to discuss the relationship between complete and GHKL-fairness. Recall that the latter depends on a specific protocol (and simulation strategy), and constitutes a sufficient criterion for complete fairness. Let f be a finite Boolean function and define $f^T : (y, x) \mapsto f(x, y)$. It might be the case that f^T is GHKL-fair, while f is not. It is easy to see however that f is indeed computable with complete fairness, the protocol in question being the GHKL-protocol with the players' roles interchanged. We conclude that for any f, if either f or f^T are GHKL-fair, then f is computable with complete fairness.

Another possibility is to consider *computationally equivalent functions* [10] i.e. functions obtained by adding/deleting redundant inputs. By applying Theorem 4.5 however, one can show that GHKL-fairness is preserved. In particular, adding/deleting redundant inputs for P_2 has no effect, and with a straightforward modification of the probability vector, the same can be shown for P_1. In summary, assuming a given function f is fair, if neither f nor f^T is GHKL-fair then to the best of our knowledge, there is no immediate way to design a completely fair protocol computing f from the one of Gordon, Hazay, Katz and Lindell.

On the other hand, assume that fair computation is impossible for a given function f. Further assume that the function is not reducible to coin-tossing (not strictly balanced [5]). Then we need a new argument to prove that it is indeed unfair. In the next section, we do exactly that.

6 A Class of Unfair Functions

In this section, we present a new class of functions for which fair computation is impossible. We begin with an example. Let $f : \{x_1, \dots, x_4\} \times \{y_1, \dots, y_4\} \to \{0, 1\}$ be the function with associated matrix

$$M = \begin{pmatrix} 0\,1\,0\,1 \\ 1\,1\,1\,0 \\ 0\,0\,1\,0 \\ 1\,0\,0\,0 \end{pmatrix} .$$

First, note that neither f nor f^T are GHKL-fair, since both of them are of type 1. Assuming there exists a completely fair realization of function f, consider the following: P_1 chooses among $\{x_1, x_3, x_4\}$ with equal probability, P_2 chooses y_4 with probability 2/5, or one of his other inputs with probability 1/5. Players compute f on the chosen inputs and obtain b. Define

$$\text{Out}_1 = b, \qquad \text{Out}_2 = \begin{cases} 1 - b & \text{if } P_2 \text{ chose } y_2 \\ b & \text{otherwise} \end{cases} .$$

A quick analysis shows that $\Pr[\text{Out}_1 = 1] = 1/3$, $\Pr[\text{Out}_2 = 1] = 2/5$, and that the two are equal with probability 4/5. If Out_1 and Out_2 were independent, they would be equal with probability 8/15, which is not the case. Finally, it is not hard to see that any malicious behaviour by either party does not affect the probability distribution of the other party's output.

In fact, we have just described a non-trivial instance of the sampling problem: parties P_1 and P_2 generate bits b_1 and b_2, respectively, according to some *joint* probability distribution. Secure sampling is said to be non-trivial if the outputs are not independent. In [2], Agrawal and Prabhakaran show that this problem cannot be realised with complete fairness. In what follows, we demonstrate that there exists a large class of functions that are reducible to the sampling problem, and are thus unfair. Formally, let $\mathcal{F}_{SS}$ denote the following functionality:

- **Parameters** $p, q \in [0,1]$ and $\chi \in [2\max(p+q-1,0)-2pq\,,\, 2\min(p,q)-2pq]$.
- **Inputs** Empty for both parties.
- **Outputs** P_1 and P_2 receive bits b_1 and $b_2 \in \{0,1\}$, respectively, such that
  1. $\Pr[b_1 = 1] = p$,
  2. $\Pr[b_2 = 1] = q$,
  3. $\Pr[b_1 = b_2] = pq + (1-p)(1-q) + \chi$.

Theorem 6.1. *Unless $\chi = 0$, functionality $\mathcal{F}_{SS}$ is not computable with complete fairness.*

Proof. The proof is a natural generalisation of Cleve's [8] original argument (where $p = q = 1/2$) and can be found in [2]. Note that if $\chi = 0$, then b_1 and b_2 are independent and $\mathcal{F}_{SS}$ is trivially fair. $\square$

In fact, $\mathcal{F}_{SS}$ is not computable with complete fairness even if we allow the adversary to learn the honest party's output. Formally, we augment the ideal model with complete fairness such that after sending the outputs, the trusted party divulges the honest party's output to the adversary. We claim that $\mathcal{F}_{SS}$ is not securely computable in this new augmented model. The claim is true because Cleve's argument makes no assumption regarding the privacy of the parties' outputs. As a corollary, we deduce that any finite Boolean function that can be reduced to an instance of the sampling problem is inherently unfair, even if the adversary learns the honest party's output.

6.1 Semi-Balanced Functions

In [3], the author identifies a class of functions that are not GHKL-fair. Here, we go one step further and show that they are inherently unfair. In particular, they are all reducible to a (non-trivial) instance of the sampling problem.

Definition 6.2. *Let f be a finite boolean function with matrix representation M and write $\overline{M}$ for the matrix representation of $1-f$. We say that f is right semi-balanced if*

$$\exists \mathbf{q} \in \mathbb{R}^k \text{ such that } \begin{cases} M\mathbf{q} = \mathbf{1}_\ell \\ \overline{M}\mathbf{q} \neq \mathbf{0}_\ell \end{cases} .$$

Similarly, f is left semi-balanced if

$$\exists \mathbf{p} \in \mathbb{R}^\ell \text{ such that } \begin{cases} M^T\mathbf{p} = \mathbf{1}_k \\ \overline{M}^T\mathbf{p} \neq \mathbf{0}_k \end{cases} .$$

Theorem 6.3. *If f is left and right semi-balanced, then f is not computable with complete fairness.*

We dedicate the rest of the section to the proof of Theorem 6.3. We show that any secure protocol computing f with respect to the ideal model with complete fairness, implies the existence of a secure protocol computing $\mathcal{F}_{SS}$ with respect to the augmented model where the adversary learns the honest party's output. Assuming f is left and right semi-balanced, fix $\mathbf{q} \in \mathbb{R}^k$, $\mathbf{p} \in \mathbb{R}^\ell$ such that

$$\begin{cases} \sum_i |p_i| = 1 \\ \mathbf{p}^T M = (\delta_1, \ldots, \delta_1), \quad \delta_1 > 0 \\ \mathbf{p}^T \overline{M} \neq (0, \ldots, 0) \end{cases}, \qquad \begin{cases} \sum_i |q_i| = 1 \\ M\mathbf{q} = (\delta_2, \ldots, \delta_2)^T, \quad \delta_2 > 0 \\ \overline{M}\mathbf{q} \neq (0, \ldots, 0)^T \end{cases} .$$

Suppose that parties have access to a trusted party $\mathcal{T}$ computing f. Consider protocol π in the *f-hybrid* model:

- **Inputs:** On empty inputs, P_1 chooses x_i with probability $|p_i|$, P_2 chooses column y_j with probability $|q_j|$.
- **Invoke Trusted Party:** P_1, P_2 invoke the trusted party on inputs x_i and y_j respectively. As per the ideal model with complete fairness (Figure 1), $\mathcal{T}$ hands both parties a bit b.
- **Outputs:** If $p_i < 0$, then P_1 outputs $1 - b$. Similarly, if $q_j < 0$, then P_2 outputs $1 - b$. Otherwise, parties are instructed to output b.

Define p^+, q^+, p^-, q^- such that $p^- = 1 - p^+ = \sum_{p_i<0} |p_i|$, and $q^- = 1 - q^+ = \sum_{q_j<0} |q_j|$.

Lemma 6.4. $(p^+ - p^-)\delta_2 = (q^+ - q^-)\delta_1$.

Proof.

$$\begin{aligned} (p^+ - p^-)\delta_2 &= \mathbf{p}^T(\delta_2, \ldots, \delta_2)^T = \mathbf{p}^T M \mathbf{q} \\ &= (\delta_1, \ldots, \delta_1)\mathbf{q} = (q^+ - q^-)\delta_1 \ . \end{aligned}$$

□

Lemma 6.5. *An honest execution of π yields the following outputs:*

- $\Pr[\mathrm{Out}_1 = 1] = \delta_1 + p^-$,
- $\Pr[\mathrm{Out}_2 = 1] = \delta_2 + q^-$,
- Out_1 *and* Out_2 *are not independent random variables.*

Proof. Write $\mathrm{Out}_2^{(1)}(x_i)$ for $\Pr[\mathrm{Out}_2 = 1 \mid x = x_i]$, then

$$\begin{aligned} \left(\mathrm{Out}_2^{(1)}(x_1), \ldots, \mathrm{Out}_2^{(1)}(x_\ell)\right)^T &= \sum_{q_i<0} |q_i|(\mathbf{1}_\ell - \mathsf{col}_i) + \sum_{q_i \geq 0} |q_i| \mathsf{col}_i \\ &= \sum_{q_i<0} |q_i| \mathbf{1}_\ell + \sum_{i=1}^{k} q_i \mathsf{col}_i \\ &= q^- \mathbf{1}_\ell + M\mathbf{q} = (q^- + \delta_2, \ldots, q^- + \delta_2)^T. \end{aligned}$$

The output of P_1 is obtained in a similar fashion. Moving on, if Out_1 and Out_2 are independent, then they are equal with probability

$$(\delta_1 + p^-)(\delta_2 + q^-) + (-\delta_1 + p^+)(-\delta_2 + q^+) \ .$$

On the other hand, when the players execute π, they will agree on the output if and only if both players flip their bits, or both players do not. Thus, $\Pr[\mathrm{Out}_1 = \mathrm{Out}_2] = p^-q^- + p^+q^+$. Consequently, if Out_1 and Out_2 are independent random variables, we deduce that

$$(\delta_1 + p^-)(\delta_2 + q^-) + (-\delta_1 + p^+)(-\delta_2 + q^+) = p^-q^- + p^+q^+ \ ,$$

which boils down to $2\delta_1\delta_2 = \delta_1(q^+ - q^-) + \delta_2(p^+ - p^-)$. Now, by Lemma 6.4 and knowing that $\delta_1, \delta_2 \neq 0$, we deduce that that $\delta_2 = \sum_j q_j$ and $\delta_1 = \sum_{i=1}^{\ell} p_i$. These in turn are equivalent to $\overline{M}\mathbf{q} = \mathbf{0}_\ell$ and $\mathbf{p}^T\overline{M} = \mathbf{0}_k^T$, which we have ruled out by assumption. Hence, we conclude that

$$\begin{aligned}&\Pr[\mathrm{Out}_1 = \mathrm{Out}_2] = \\ &\qquad \Pr[\mathrm{Out}_1 = 0]\Pr[\mathrm{Out}_2 = 0] + \Pr[\mathrm{Out}_1 = 1]\Pr[\mathrm{Out}_2 = 1] + \chi \ ,\end{aligned}$$

for some $\chi \neq 0$. □

It remains to show that every adversary in the f-hybrid model can be simulated in the ideal model. We briefly describe the simulation strategy for a corrupted P_1, the other case being identical. First, construct $\ell \times k$ matrix $\widetilde{M}$ such that the j-th column of $\widetilde{M}$ is equal to col_j if $q_j > 0$, and $1 - \mathsf{col}_j$ otherwise. Now, simulator $\mathcal{S}$ invokes $\mathcal{A}$ on the security parameter and auxiliary input z. The adversary hands x_i to $\mathcal{S}$ for the computation of f, and the simulator invokes the trusted party for computing $\mathcal{F}_{SS}$. As per the augmented model with complete fairness, the trusted party leaks the honest party's output, say Out_2, to $\mathcal{S}$. The simulator chooses y_j according to the prescribed distribution conditioned on $\widetilde{M}_{i,j} = \mathrm{Out}_2$. If $q_j < 0$, the simulator hands $1 - \mathrm{Out}_2$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ hands Out_2 to $\mathcal{A}$. In any case, the simulator outputs whatever $\mathcal{A}$ outputs, and halts. It is easy to see that the simulation strategy yields identical output distributions in the hybrid and ideal world.

Conclusion

To conclude, we make an observation regarding the number of functions that are either GHKL-fair or semi-balanced. A weaker version appears in [3] and the argument is based on [14]. Take a random function $f : X \times Y \to \{0, 1\}$ such that $|X| > |Y|$. Then, with probability greater than $1 - \nu(|Y|)$, where $\nu(*)$ is some negligible function, f is GHKL-fair. Intuitively, this occurs because k random $0/1$-vectors of size ℓ (with $\ell > k$) will almost surely form a linearly independent set and yet their linear span will not contain the all-1 vector.

Similarly, take a random function $f : X \times Y \to \{0, 1\}$ such that $|X| = |Y|$. Then, with probability greater than $1 - \nu(|Y|)$, where $\nu(*)$ is the same negligible function, f is (left *and* right) semi-balanced. The intuition now relies on the fact that for a random square matrix M, both M and $\overline{M}$ are non-singular with overwhelming probability. Putting everything together, we come to the following conclusion: Almost all functions for which $|X| \neq |Y|$ are computable with fairness, whereas almost all functions for which $|X| = |Y|$ are not.

References

[1] Agrawal, S., Prabhakaran, M.: On fair exchange, fair coins and fair sampling. In: Canetti, R., Garay, J. (eds.) CRYPTO '13, LNCS, vol. 8042, pp. 259–276. Springer, Heidelberg (2013)

[2] Asharov, G.: Towards characterizing complete fairness in secure two-party computation. In: Lindell, Y. (ed.) TCC '13, LNCS, vol. 8349, pp. 291–316. Springer, Heidelberg (2014)

[3] Asharov, G.: Towards characterizing complete fairness in secure two-party computation (extended version). Cryptology ePrint Archive, Report 2014/098 (2014), http://eprint.iacr.org/2014/098

[4] Asharov, G., Lindell, Y., Rabin, T.: A full characterization of functions that imply fair coin tossing and ramifications to fairness. In: Sahai, A. (ed.) TCC '13, LNCS, vol. 7785, pp. 243–262. Springer, Heidelberg (2013)

[5] Blum, M.: Coin flipping by telephone a protocol for solving impossible problems. SIGACT News 15(1), 23–27 (1983)

[6] Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptology 13(1), 143–202 (2000)

[7] Cleve, R.: Limits on the security of coin flips when half the processors are faulty. pp. 364–369. STOC '86, ACM (1986)

[8] Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press (2004)

[9] Gordon, D.S., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation. pp. 413–422. STOC '08, ACM (2008)

[10] Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation (extended version). Cryptology ePrint Archive, Report 2008/303 (2008), http://eprint.iacr.org/2008/303

[11] Moran, T., Naor, M., Segev, G.: An optimally fair coin toss. In: TCC '09. pp. 1–18. LNCS, Springer, Heidelberg (2009)

[12] Yao, A.C.: Protocols for secure computations pp. 160–164 (1982)

[13] Ziegler, G.M.: Lectures on 0/1-polytopes. In: Kalai, G., Ziegler, G.M. (eds.) Polytopes Combinatorics and Computation, DMV Seminar, vol. 29, pp. 1–41. Birkhauser Basel (2000)

A Considering Probability Vectors with Zero-Components

Let $f = \{x_1, \ldots, x_\ell\} \times \{y_1, \ldots, y_k\} \to \{0,1\}$ and consider its associated matrix M. Furthermore, fix a probability vector $\mathbf{p} \in \mathbb{R}^\ell$ and define $k \times k$ matrix $P = \mathsf{diag}(p_{y_1}, \ldots, p_{y_k})$ where

$$(p_{y_1}, \ldots, p_{y_k}) = \mathbf{p}^T M \ .$$

Recall the alternate GHKL-fairness conditions: for all $i \in \{1,\dots,\ell\}$, for all $a \in \{0,1\}$, there exists $(\mu_{i,1}^{(a)},\dots,\mu_{i,\ell}^{(a)})$ satisfying

1. $\sum_j \mu_{i,j}^{(a)} = 0$,
2. $(\mu_{i,1}^{(0)},\dots,\mu_{i,\ell}^{(0)})M = (\mathbf{1}_k^T - \mathsf{row}_i)P$ with $\mu_{i,j}^{(0)} \geq 0$ if $\mathbf{p}(j)=0$,
3. $(\mu_{i,1}^{(1)},\dots,\mu_{i,\ell}^{(1)})M = \mathsf{row}_i(P - \mathrm{Id}_k)$ with $\mu_{i,j}^{(1)} \geq 0$ if $\mathbf{p}(j)=0$.

Without loss of generality, suppose that $\mathbf{p}(1)=0$ and that the GHKL-fairness conditions are satisfied. Then

$$\begin{aligned}(\mu_{1,1}^{(0)},\dots,\mu_{1,\ell}^{(0)})M &= (\mathbf{1}_k^T - \mathsf{row}_1)P\\ (\mu_{1,1}^{(1)}+1,\dots,\mu_{1,\ell}^{(1)})M &= \mathsf{row}_1 P \ .\end{aligned}$$

Add the two expressions together:

$$(\mu_{1,1}^{(0)}+\mu_{1,1}^{(1)}+1,\dots,\mu_{1,\ell}^{(0)}+\mu_{1,\ell}^{(1)})M = \mathbf{1}_k^T P = \mathbf{p}^T M \ .$$

Thus,

$$\Big(\mu_{1,1}^{(0)}+\mu_{1,1}^{(1)}+1-\mathbf{p}(1),\ \dots,\ \mu_{1,\ell}^{(0)}+\mu_{1,\ell}^{(1)}-\mathbf{p}(\ell)\Big)M = 0 \ ,$$

and note that $\mu_{1,1}^{(0)}+\mu_{1,1}^{(1)}+1-\mathbf{p}(1) > 0$. Now, define $\mathcal{D} = \{\, j \mid \mathbf{p}(j)=0\}$ and let $d = |\mathcal{D}|$. Using the same trick as above for every row indexed by $\mathcal{D}$, deduce that $(\nu_1,\dots,\nu_\ell)M = 0$, where

- $\nu_i = 1 + \sum_{j\in\mathcal{D}}(\mu_{j,i}^{(0)}+\mu_{j,i}^{(1)}) > 0$ if $i \in \mathcal{D}$,
- $\nu_i = -d\cdot\mathbf{p}(i) + \sum_{j\in\mathcal{D}}(\mu_{j,i}^{(0)}+\mu_{j,i}^{(1)})$ if $i \notin \mathcal{D}$,
- $\sum_i \nu_i = 0$.

Next, choose $\gamma > 0$ such that

$$-\mathbf{p}(i) < \gamma\cdot\nu_i < 1-\mathbf{p}(i) \ ,$$

for every $i \in \{1,\dots,\ell\}$, and define probability vector $\widetilde{\mathbf{p}} = \mathbf{p} + \gamma\cdot\nu$. By noting that $\widetilde{\mathbf{p}}^T M = \mathbf{p}^T M = (p_{y_1},\dots,p_{y_k})$, we conclude that function f is GHKL-fair for a new probability vector without zero entries.