HOW N PEOPLE CAN SIMULATE A FAIR COIN FLIP

PHIL WERTHEIMER

August 18, 2011

Abstract. This paper provides an introduction and overview to mystery values, which are analogues to eigenvalues that arise in the [KWG10] model of secure multiparty computation. Essentially, most of the paper serves to answer, and elaborate on, the question "when can n people simulate a fair coin flip?"

Contents

1. Introduction

Secure multiparty computation considers situations where parties with private variables would like to compute some public function of these variables while maintaining individual secrecy. The prototypical example of this situation is Andrew Yao's Millionaire Problem, where two millionaires would like to know who has more money, but neither wants to reveal how much he is worth. This is in fact possible, as Yao proved in his 1982 paper.

This paper is concerned with a secure multiparty computation problem related to the idea of simulating a probability space. Simulating a probability space can be thought of as using one or more spaces in conjunction to model another space e.g. simulating a coin flip with a die roll by labeling 1-3 "heads" and 4-6 "tails." The problem is thus: can n people, each equipped with a coin, simulate a fair coin flip? Can they simulate a coin flip that generates heads with an arbitrary probability p and tails with probability (1 - p)?

In order to answer this question, this paper gives a category-theoretic introduction to the Kronecker product: the primary tool for constructing what [KWG10] dubbed "mystery values." These analogues to eigenvalues provide insight into the simulation of probability spaces, and this paper ultimately hints at their application to number theory.

2. Category Theory

Definition 2.1. A **category** $\mathcal{C}$ consists of a collection of objects, Ob($\mathcal{C}$), along with maps that connect these objects. We call these maps **morphisms**; for A,B $\in$ C, let $\mathrm{Hom}_{\mathcal{C}}$(A,B) denote the set of morphisms from A to B. Categories must obey two axioms:

CAT 1: The composition of morphisms is associative. That is, for all objects A, B, C and D, all morphisms f: A $\to$ B, g: B $\to$ C, and h: C $\to$ D satisfy (h $\circ$ g) $\circ$ f = h $\circ$ (g $\circ$ f).
CAT 2: For each object C $\in$ Ob($\mathcal{C}$) there is a morphism id_C, called the identity morphism, such that for every morphism f: A $\to$ B, $\mathrm{id}_B \circ \mathrm{f} = \mathrm{f} = \mathrm{f} \circ \mathrm{id}_A$.

For example, we can consider the category of all vector spaces, whose morphisms are linear transformations; or the category of all groups, whose morphisms are group homomorphisms. We would also like to consider morphisms between categories.

Definition 2.2. Let $\mathcal{C}$, $\mathcal{D}$ be categories. A **functor** F: $\mathcal{C} \to \mathcal{D}$ associates to each object C $\in$ Ob($\mathcal{C}$) an object F(C) $\in$ Ob($\mathcal{D}$). Since categories are fundamentally made up of both objects and morphisms, functors must also act on morphisms, associating with each morphism $f \in \mathrm{Hom}_{\mathcal{C}}$(A,B) a morphism F($f$) $\in \mathrm{Hom}_{\mathcal{D}}$(F(A),F(B)). Functors must obey two axioms to ensure preservation of composition and identity morphisms.

FUN 1: For all morphisms f: A $\to$ B and g: B $\to$ C, F(g $\circ$ f)= F(g) $\circ$ F(f).
FUN 2: $\mathrm{F}(\mathrm{id}_C) = \mathrm{id}_{F(C)}$ $\forall$ C $\in$ Ob(C).

So far, we have explored maps between objects in a category (morphisms) and maps between categories themselves (functors). We can also consider maps between functors.

Definition 2.3. Given two functors F and G between the categories $\mathcal{C}$ and $\mathcal{D}$, a **natural transformation** η: F $\to$ G is a collection of morphisms in $\mathcal{D}$ indexed by the objects in $\mathcal{C}$. Specifically, each object C $\in \mathcal{C}$ is associated with a morphism η_C: F(C) $\to$ F(D) in $\mathcal{D}$, called the **component** of η at C, such that for every morphism f: X $\to$ Y in $\mathcal{C}$ we have: $\eta_Y \circ \mathrm{F(f)} = \mathrm{G(f)} \circ \eta_X$. This equation is depicted in the following commutative diagram:

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ {\scriptstyle \eta_X}\downarrow & & \downarrow{\scriptstyle \eta_Y} \\ G(X) & \xrightarrow[G(f)]{} & G(Y) \end{array}$$

Definition 2.4. Since every group is also a set, there exists a functor from the category of all groups to the category of all sets. This functor is appropriately named the **forgetful functor**. It maps both objects and morphisms to themselves, treating groups as sets and group homomorphisms as set functions. Note that the forgetful functor can be applied to any two categories, where one of them "is" the other e.g. a vector space is a set, a commutative ring is a ring. In this paper, U will denote the forgetful functor.

Definition 2.5. Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Given functors F: $\mathcal{C} \to \mathcal{D}$ and G: $\mathcal{D} \to \mathcal{C}$, we say that F is a **left adjoint functor** to G and that G is a **right adjoint functor** to F if for every pair of objects C $\in$ Ob($\mathcal{C}$), D $\in$ Ob($\mathcal{D}$), there is a bijection $\phi_{C,D}$: Hom$_{\mathcal{D}}$(F(C),D) $\to$ Hom$_{\mathcal{C}}(C, G(D))$ which is natural in C and D. The existence of this family of bijections is denoted by Hom$_{\mathcal{C}}$(C,G(D)) $\cong$ Hom$_{\mathcal{D}}$(F(C),D).

Definition 2.6. Whereas forgetful functors "forget" some structure, there are functors that add structure. The left adjoint to a forgetful functor is called a **free functor**.

Example 2.7. Let *Grp* denote the category of all groups, and let *Ab* denote the category of all abelian groups. The free functor from *Grp* into *Ab* is known as the **abelianization** of *Grp*. The abelianization functor F can be realized as follows: Let G be a group. A commutator is an element of the form $aba^{-1}b^{-1}$, where a,b $\in$ G; such a commutator is denoted [a,b]. Let [G,G] denote the commutator subgroup of G; that is, the subgroup of G generated by its commutators. It is not hard to prove that [G,G]$\lhd$ G. Therefore, G/[G,G] has a group structure; in fact, modding out by [G,G] turns out to be precisely the abelianization functor. Specifically, the abelianization functor sends objects (groups) to their quotient groups by their commutator subgroups, and sends morphisms to the unique morphism so that the quotient map is a natural transformation. For any group G, let η_G: G $\to$ U(F(G)) = G/[G,G] be the quotient map and for any abelian group A, let ϵ_A: A $\to$ U(F(A)) = A be the identity map. Then for any α: F(G) $\to$ A, define $\phi_{G,A}(\alpha)$ to be $\alpha \circ \eta_G$: G $\to$ U(A). Then $\phi_{G,A}$ is an isomorphism for every G and A which satisfies the properties of a natural transformation. Furthermore, $\epsilon_A = \phi^{-1}(\mathrm{id}_{U(A)})$ for every A, as required. Therefore, we have the desired adjunction between *Grp* and *Ab*. We must also check that the functor is well-defined and does, in fact, "abelianize" *Gr*.

Let g,h $\in$ G. g[G,G] = h[G,G] $\Rightarrow$ h^{-1}g[G,G] = [G,G] $\Rightarrow$ h^{-1}g $\in$ [G,G] $\Rightarrow$ F(h^{-1}g) $\in$ F([G,G]) = {1}, as [G,G] is the kernel of FG. Therefore h^{-1}g = 1, and h=g as desired.

Since all commutators are mapped to the identity in G/[G,G], we have $ghg^{-1}h^{-1}$[G,G]=1 $\forall$ g,h $\in$ G. Thus, gh[G,G]=hg[G,G] $\forall$ g,h $\in$ G i.e. F(G) = G/[G,G] is abelian.

Definition 2.8. Let $\mathcal{C}$ be a category, and let A,B,X,Y $\in$ Ob($\mathcal{C}$). We define the **Hom functor** from $\mathcal{C}$ to *Set*, the category of all sets, as follows. Hom$_{\mathcal{C}}$(A, -): $\mathcal{C} \to$ *Set* maps each object X in $\mathcal{C}$ to the set of morphisms Hom$_{\mathcal{C}}$(A,X), and maps each morphism f: X $\to$ Y to the function Hom$_{\mathcal{C}}$(A,f): Hom$_{\mathcal{C}}$(A,X) $\to$ Hom$_{\mathcal{C}}$(A,Y) given by g $\mapsto$ f $\circ$ g $\forall$ g $\in$ Hom$_{\mathcal{C}}$(A,X). Similarly, Hom$_{\mathcal{C}}$(-, B): $\mathcal{C} \to$ *Set* maps each object X $\in$ Ob($\mathcal{C}$) to the set of morphisms Hom$_{\mathcal{C}}$(X,B), and sends each morphism h: X $\to$ Y to the function

$\mathrm{Hom}_{\mathcal{C}}$(h,B): $\mathrm{Hom}_{\mathcal{C}}$(Y,B) $\to$ $\mathrm{Hom}_{\mathcal{C}}$(X,B) given by g $\mapsto$ g $\circ$ h $\forall$g $\in$ $\mathrm{Hom}_{\mathcal{C}}$(Y,B). The Hom functor induces the following diagram:

$$\begin{array}{ccc} X & & Hom_{\mathcal{C}}(A,X) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Hom_{\mathcal{C}}(A,f)} \\ Y & & Hom_{\mathcal{C}}(A,Y) \end{array}$$

Now let us consider the hom functor over $\mathcal{V}$ (the category of all vector spaces) $\mathrm{Hom}_{\mathcal{V}}$(V, -): $\mathcal{V} \to \mathcal{V}$. This set of morphisms is actually a vector space itself. Elements of $\mathrm{Hom}_{\mathcal{V}}$(V, -) are linear transformations that we can represent with matrices. For linear transformations L and M, L+M and cL are also linear transformations $\forall$ c $\in \mathbb{R}$.

Definition 2.9. Let V, W, and X be vector spaces. A function f: V$\times$W $\to$ X is called **bilinear** if it is linear in each variable separately. That is, $\forall$ v_1,v_2 $\in$ V, w_1,w_2 $\in$ W and all scalars a and b, f(av_1 + bv_2, w_1) = af(v_1,w_1) + bf(v_2,w_1) and f(v_1, aw_1 + bw_2) = af(v_1,w_1) + bf(v_1,w_2).

Definition 2.10. Let V, W, X, and Y be vector spaces. A bilinear map $\otimes$: V$\times$W $\to$ X is said to have the **universal property** if for every bilinear map ϕ: V$\times$W $\to$ Y, there exists a unique linear map ψ: X $\to$ Y such that the following diagram commutes:

$$\begin{array}{ccc} V \times W & \xrightarrow{\phi} & Y \\ {\scriptstyle \otimes}\downarrow & \nearrow_{\psi} & \\ X & & \end{array}$$

Definition 2.11. The **tensor product** of two vector spaces V and W, $\otimes$: V$\times$W $\to$ X, is a bilinear map with the universal property. The space X is uniquely determined by V and W up to isomorphism— that is, tensor products are unique— and is denoted V$\otimes$W. Hence:

$$\begin{array}{ccc} V \times W & \xrightarrow{\phi} & Y \\ {\scriptstyle \otimes}\downarrow & \nearrow_{\psi} & \\ V \otimes W & & \end{array}$$

Theorem 2.12. *Tensor products exist.*
Proof. I will now give an explicit construction of the tensor product over two vector spaces U and V.

Lemma 2.12: Let B: U×V → W be a bilinear map. Then B is uniquely determined by its values on the basis vectors $(\overrightarrow{u_i},\overrightarrow{v_i})$. Furthermore, any choice of values on these vectors determines a bilinear function.
Proof. Let $\overrightarrow{u} = \sum \mathrm{a}_i\overrightarrow{u_i}$ and let $\overrightarrow{v} = \sum \mathrm{b}_j\overrightarrow{v_j}$. Then by linearity in the first variable, $\mathrm{B}(\overrightarrow{u},\overrightarrow{v}) = \sum_i \mathrm{a}_i\mathrm{B}(\overrightarrow{u_i},\overrightarrow{v})$. By linearity in the second variable, we have $\sum_i \mathrm{a}_i \sum_j \mathrm{b}_j\mathrm{B}(\overrightarrow{u_i},\overrightarrow{v_j})$ $= \sum_{i,j} \mathrm{a}_i b_j\mathrm{B}(\overrightarrow{u_i},\overrightarrow{v_j})$. Thus, specifying the values on $(\overrightarrow{u_i},\overrightarrow{v_i})$ defines B, and since these vectors form a basis, any collection of values yields a bilinear function. □

Now define U⊗V = $\mathbb{F}\{(\overrightarrow{u_i},\overrightarrow{v_i})\}$. From the Lemma, the bilinear functions U×V are in 1-1 correspondence with linear maps $\mathbb{F}\{(\overrightarrow{u_i},\overrightarrow{v_i})\}$, so to verify the universal property we need only produce a bilinear map U×V → U⊗V. Using our 1-1 correspondence, we simply need a linear transformation U⊗V → U⊗V; the identity works just fine. □

Corollary 2.13. dim(V⊗W)=dim(V)·dim(W).

Theorem 2.14. *Tensor products are unique. That is, if* ⊗*: V*×*W* → *(V*⊗*W)* are bilinear maps with the universal property, then there exists an isomorphism f: (V*⊗*W)** → *(V*⊗*W)* such that* ⊗* = *f* ∘ ⊗.
Proof: From the definition of tensor product, substitute (V⊗W)* for Y and ⊗* for ϕ to obtain a unique map ψ: (V⊗W) → (V⊗W)* such that ⊗* = ψ ∘ ⊗. Reversing the roles of ((V⊗W),⊗) and ((V⊗W)*,⊗*), we also have a map ψ^*: (V⊗W)* → (V⊗W) such that ⊗ = ψ^* ∘ ⊗*. I claim that $\psi \circ \psi^* = \mathrm{id}_{(V\otimes W)^*}$ and $\psi^* \circ \psi = \mathrm{id}_{(V\otimes W)}$, and hence ψ is an isomorphism. Indeed, $(\psi^* \circ \psi)\circ \otimes = \psi^* \circ (\psi \circ \otimes) = \psi^* \circ \otimes^* = \otimes$. Now apply the definition of tensor products, substitituing V⊗W for Y and ⊗ for ϕ. Then both $\psi^* \circ \psi$ and $\mathrm{id}_{(V\otimes W)}$ can play the role of ⊗. So by uniqueness of ⊗ in the definition, we see that $\psi^* \circ \psi = \mathrm{id}_{(V\otimes W)}$. Finally, to see that the map ⊗ that appears in the statement of theorem is unique, note that from the first statement of this proof, there is only one map ⊗ such that ⊗* = ψ ∘ ⊗. □

Remark. Given vector spaces V and W, the term "tensor product" applies both to the bilinear map ⊗ and the product space V⊗W. The intended meaning should be clear from context.

Definition 2.15. The **Kronecker product** of matrices corresponds to the tensor product of linear maps. Let V,W,X and Y be vector spaces and let A: V → W and B: X → Y be matrices of linear transformations. Then the Kronecker product A ⊗ B represents the tensor product of the two maps V ⊗ X → W ⊗ Y as an mn × rs block matrix:

$$\begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

3. Secure Multiparty Computation

3.1. **Introduction and a Few Cool Problems.** Distributed computing considers scenarios where a number of distinct, connected computing devices (or parties) wish to carry out a joint computation of some function. Secure multiparty computation seeks to enable parties to carry out distributed tasks securely. Although it is difficult to come up with an exact definition for security, [L02] offers some central properties that a secure computation should exhibit.

Definition 3.1. [L02] Properties of a secure computation.

(1) Privacy: no party should learn anything more than its prescribed output; specifically, the information gained about other parties' inputs is only what can be derived from the output itself.

(2) Guaranteed output delivery: Corrupted parties should not be able to prevent honest parties from receiving their output i.e. resistance to a "denial of service" attack.

(3) Fairness: Corrupted parties should receive their outputs iff honest parties receive theirs. If we are dealing with electronic signatures, it would be very bad if a corrupt party obtains a signed contract and the honest party does not.

(4) Correctness

Example 3.2: Consider a teacher who wants to determine how many hours per week the average student in her class spends on homework. A simple survey would provide inaccurate results; those who spend many hours per week on homework will be embarrassed to admit it, while those who spend very little time on their work will lie to protect their reputation as good students. Even an anonymous survey may not correct this problem. However, a very simple, yet clever, protocol exists for this multiparty computation to be secure. Namely, student A takes the number of hours per week that he studies, and adds to it a random $n \in \mathbb{N}$. He then whispers this sum to student B, who adds her hours to the sum, and whispers this new sum to student C. This process continues until it comes back to student A, who then subtracts n from the number he receives. He then passes the number to the teacher, who can divide by the number of students to obtain the average she desired. Note that each student learns absolutely nothing about the individuals prior (assuming, of course, that n is sufficiently random).

Example 3.3: **Fuzzy Russell Problem**: N friends are seated at a circular table in a restaurant. After playing a cooperative game for some time, they decide to spice it up and play a non-cooperative game: a game in which one of the players is working against everyone else; this adversary will be called Fuzzy Russell. Given that each person is capable of random number generation, but may only whisper to the people seated directly to his left and right, determine a protocol whereby the N friends can choose the Fuzzy Russell, such that only the Fuzzy Russell knows that he has been chosen, and no one else gains any information about who it could be. That is, when the protocol is finished, if you are not

the Fuzzy Russell, you know that you are not the Fuzzy Russell, but each other person could be the Fuzzy Russell with equal probability. Assume each participant is honest i.e. will obey the protocol.

Lemma 3.3: It is possible for one person to privately communicate with another at the table.

Proof. Any message can be converted to a binary string; for example, 0110100101110100 represents the word "it". Furthermore, any binary string can be converted to an integer. Thus, let us view any message as an integer, say $\mathcal{M}$. Randomly generate $\mathcal{L} \in \mathbb{Z}$, with $\mathcal{L} < \mathcal{M}$. Now send the integer $(\mathcal{M} - \mathcal{L})$ left, with the instruction "pass this along until it reaches person X, and tell him to add it to the other integer he receives" and send the integer $\mathcal{L}$ to the right, with the same instruction. Person X adds both integers he receives to reveal $\mathcal{M}$, the message. Note that knowledge of only $(\mathcal{M} - \mathcal{L})$ or $\mathcal{L}$ reveals nothing about $\mathcal{M}$; in fact, these integers may not even correspond to actual strings of letters. And, assuming that the participants are honest, the message Person X receives will always be the intended one; clearly $(\mathcal{M} - \mathcal{L}) + \mathcal{L} = \mathcal{M}$. □

The Protocol:

(1) Person A assigns an integer ∈ {1,2,...,N} to each of the people at the table by sending each person a private message—in the sense of the lemma— with his/her number (of course, these are randomly chosen). So, everyone knows his/her number and no one else's, except for Person A who knows everyone's.

(2) Person B randomly generates an integer ∈ {1,2,...,N}, representing the number of the person who will be Fuzzy Russell. Person B privately distributes this integer to everyone except for Person A, using the protocol from the lemma.

(3) Person B randomly chooses a bit (0 or 1) and sends it both ways along the circle, with the instructions "If you are not the Fuzzy Russell, pass this on. If you are the Fuzzy Russell, switch the bit (i.e. add 1 mod 2) and pass that on. Continue until this reaches Person A."

(4) Person A receives two bits. If they are the same, he knows he is the Fuzzy Russell; if they are different, he knows he isn't the Fuzzy Russell, but gains no information about who it could be. If he knew the original bit that was sent, then he could narrow down the possible Fuzzy Russells to the people on one side of him. But this, of course, cannot happen if the bit was chosen randomly.

3.2. **Oracles and Modeling Probability Spaces.** As the Fuzzy Russell Problem demonstrates, it is often necessary for each party involved in a secure multiparty computation to be able to generate random numbers. In order to discuss random number generation, it is essential to define a probability space.

Definition 3.4. The **sample space** Ω of an experiment or random trial is the set of all possible outcomes (also called "events").

So for a coin flip, Ω = {Heads,Tails}; for a dice roll, Ω = {1,2,3,4,5,6}. We would like to have a function which, given an event as an input, produces the probability of that event occurring as its output.

Definition 3.5. A **probability measure** μ is a function on a sample space such that
i. μ takes values in [0,1], returning 0 for the empty set and 1 for the entire space.
ii. μ must satisfy the *countable additivity* property: for all countable collections $\{E_i\}$ of pairwise disjoint sets in the sample space, $\mu(\bigcup_{i\in I} E_i) = \sum_{i\in I} \mu(E_i)$.

Definition 3.6. A **probability space** is a triple (Ω, $\mathcal{E}$, μ) where:

(1) Ω is a sample space.

(2) $\mathcal{E}$ is a set of subsets of Ω, called events, such that
I. $\mathcal{E}$ contains $\emptyset$.
II. $\mathcal{E}$ is closed under complements: if $E \in \mathcal{E}$, then $(\mathcal{E} - E) \in \mathcal{E}$.
III. $\mathcal{E}$ is closed under countable unions: if $E_i \in \mathcal{E}$ for countably many i, then $\bigcup_{i=1,2,...} E_i \in \mathcal{E}$.

(3) μ is a probability measure i.e. μ: $\mathcal{E} \to [0,1]$ such that:
I. μ is countably additive: if $\{E_i\} \subset \mathcal{E}$ is a countable collection of pairwise disjoint sets, then $\mu(\bigcup E_i) = \sum \mu(E_i)$, where $\bigcup$ denotes the disjoint union.
II. The measure of the entire space is 1. That is, $\mu(\Omega)=1$.

This may seem like a lot of information, but for the purposes of this paper, our sample space will be the possible outcomes of some number of coin flips (events): {Heads, Tails}, and the probability measure just tells us how likely it is, for example, to get three heads from four coin flips.

The notion of modeling probability spaces is best explained with an example: For instance, one can model a coin flip by picking a card out of a fair deck: label the red cards "heads" and the black cards "tails." The quotients of a probability space $\mathcal{P}$ are precisely the spaces a person can model with $\mathcal{P}$.

Definition 3.7. A **finite random oracle** is a collection $\mathcal{O}$ of finite probability spaces that is closed under quotients. That is, if $\mathcal{P} \in \mathcal{O}$ and there is a measure-preserving map $\mathcal{P} \to \mathcal{Q}$, then $\mathcal{Q} \in \mathcal{O}$. Random oracles are partially ordered by inclusion: we say that $\mathcal{O}$ is **stronger than** $\mathcal{U}$ iff $\mathcal{O} \supset \mathcal{U}$. For any set $\mathbb{S}$, an $\mathbb{S}$-random oracle takes probabilities in $\mathbb{S}$.

Note, being closed under quotients is not really a restriction: it demonstrates the notion that certain probability spaces can model other probability spaces.

In secure multiparty computation, each party is sometimes assumed to have a private random oracle. If one party does not have access to a random oracle, we would like to know if it is possible for the other parties to simulate one.

Example 3.8: Suppose we have a jar of n coins, one of which is fair, the rest of which are unfair with unknown biases (i.e. do not produce heads and tails with an equal probability of .5). Given this jar, it is possible to simulate a fair coin flip without knowledge of which coin is the fair one. The protocol is as follows: flip every coin once and record the results. If there are an even number of heads, call the result "heads;" if there are an odd number of heads, call the result "tails." To see why this works, imagine that the (n-1) unfair coins are flipped first. Let h be the number of heads recorded from these (n-1) flips. Since flipping the fair coin produces heads and tails with equal likelihood, after all n flips, it is equally likely that the number of heads is h or (h+1). That is, the parity of h is fairly determined, so even though all but one of the coins are unfairly biased, the result still hinges upon the result of the fair coin flip.

4. Simulating Finite Random Oracles

Definition 4.1. Robustness is immunity to the malfunctioning of one or more random oracles. For instance, if Alice has several finite random oracles and wants to generate a random bit with bias α, her algorithm is said to be robust if it works correctly even if one or more oracles is miscalibrated (and she doesn't know which one(s)).

Definition 4.2. If n people with private full-strength finite random oracles can robustly simulate a coin flip with bias α, we say α is **n-cooperative**. The set of all n-cooperative numbers is denoted $\mathfrak{C}$(n).

The notion of robustness is actually equivalent to that of privacy in random oracle simulation. If several parties want to simulate a private coin flip for another party, the result of the flip cannot depend on any one of the sample spaces; rather, it can only depend on them all taken together. Similarly, Alice's bit cannot depend on any one of the oracles because she doesn't know which ones are unreliable, so it must depend on all of them taken together.

We would like to be able to concoct a {heads,tails}- valued function capable of producing an α-biased coin flip. [KWG10] models each probability space as a probability vector (each coordinate is between 0 and 1 and the sum of all the coordinates is 1). Under this construction, we can view the product probability space as the Kronecker product of these vectors. For example, suppose we have a fair coin and a fair 4-sided object like a dreidel[1]. We can assign "heads" or "tails" to each result obtained from one flip of the coin and one spin of the dreidel.

[1]An obscure reference for my Jewish readers.

$$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \otimes \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/8 & 1/8 \\ 1/8 & 1/8 \\ 1/8 & 1/8 \\ 1/8 & 1/8 \end{bmatrix} \to \begin{bmatrix} H & T \\ T & H \\ T & H \\ H & T \end{bmatrix}$$

The H/T matrix represents assignment of outcomes to either "heads" or "tails" in order to model a coin flip. So, if we want to calculate the probability of heads, we can substitute H=1, T=0 into our matrix to get:

$$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \frac{1}{2}.^2$$

This construction provides a quick check for robustness. Let's suppose one of the oracles is miscalibrated e.g. the dreidel is weighted, and calculate the new probability of heads:

$$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/12 \\ 1/4 \\ 1/6 \\ 1/2 \end{bmatrix} = \frac{1}{2}.$$

Interestingly, the result hasn't changed. In fact, let

$$A(x^{(1)}, x^{(2)}) = x^{(1)} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} x^{(2)}$$

and let $\beta^{(1)} = [\frac{1}{2}\ \frac{1}{2}]$, $\beta^{(2)} = [\frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]^T$. Then

$$\begin{aligned} \mathrm{A}(\mathrm{x}^{(1)}, \beta^{(2)}) &= \tfrac{1}{2} \\ \mathrm{A}(\beta^{(1)}, \mathrm{x}^{(2)}) &= \tfrac{1}{2} \end{aligned}$$

for all probability vectors $\mathrm{x}^{(1)}$ and $\mathrm{x}^{(2)}$. This explains the coin flipping of example 3.8 when the jar only contains n=2 coins: with one fair coin, it does not matter how unfair the other coin (or in this case, dreidel) is; we can still simulate a fair coin flip. That is to say, as long as only one of the oracles is miscalibrated, the simulated coin flip is still robust.

[2]Note the transpose of our matrix was used here. The original would have been equally usable, but would have required me to change the order of the matrix multiplactions.

Let J represent the all-ones tensor of appropriate dimensions. If α denotes the bias of the bit, the robustness condition becomes

$$\begin{gathered}
\mathrm{A}(\mathrm{x}^{(1)}, \beta^{(2)}) = \alpha \mathrm{J}(\mathrm{x}^{(1)}, \beta^{(2)}) \\
\mathrm{A}(\beta^{(1)}, \mathrm{x}^{(2)}) = \alpha \mathrm{J}(\beta^{(1)}, \mathrm{x}^{(2)}) \\
\Downarrow \\
(\alpha \mathrm{J}\text{-}\mathrm{A})(\mathrm{x}^{(1)}, \beta^{(2)}) = 0 \\
(\alpha \mathrm{J}\text{-}\mathrm{A})(\beta^{(1)}, \mathrm{x}^{(2)}) = 0 \\
\Downarrow \\
(\alpha \mathrm{J}\text{-}\mathrm{A}) \text{ is degenerate} \\
\Downarrow \\
\mathrm{Det}(\alpha \mathrm{J}\text{-}\mathrm{A}) = 0
\end{gathered}$$

This certainly looks familiar: changing J to I makes α an eigenvalue for the left and right eigenvectors $\beta^{(1)}$ and $\beta^{(2)}$.

Definition 4.3. [KWG10] We call α a **mystery value** of the n-linear form A if, for any $1 \leq \mathrm{j} \leq \mathrm{n}$,

$$(\alpha \mathrm{J}\text{-}\mathrm{A})(\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(j-1)}, \mathrm{x}^{(j)}, \beta^{(j+1)}, \ldots, \beta^{(n)}) = 0 \ \forall \mathrm{x}^{(j)}.$$

The vectors $\beta^{(i)}$ are α's corresponding **mystery vectors**.

So, an alternate definition could be: a function from the product of several finite probability spaces generates an α-biased bit robustly iff the corresponding multilinear form has mystery value α with the probability spaces as mystery vectors.

Theorem 4.4. [KWG10] *Mystery values are unique.*
Proof. Suppose α and α^* are both mystery values for the matrix A, with mystery vectors $\beta^{(i)}$ and $\beta^{(i^*)}$, respectively. By definition of mystery values and mystery vectors,

$$\begin{gathered}
\mathrm{A}(\mathrm{x}^{(1)}, \beta^{(2)}) = \alpha \\
\mathrm{A}(\beta^{(1^*)}, \mathrm{x}^{(2)}) = \alpha^*
\end{gathered}$$

Since the $\mathrm{x}^{(i)}$ can be replaced by any probability vectors, we obtain: $\alpha = \mathrm{A}(\beta^{(1^*)}, \beta^{(2)}) = \alpha^* \Rightarrow$ any 2 mystery values are equal. □

Corolloary 4.5. [KWG10] *If α is a mystery value of a bilinear form A, then $\alpha \notin (\mathbb{R} - \mathbb{Q})$. That is, two parties cannot simulate an irrationally-biased coin.*
Proof. Let A be a {0,1}-matrix with mystery value α. Any field automorphism $\sigma \in \mathrm{Gal}(\mathbb{C}/\mathbb{Q})$ fixes A, as its entries are all rational, so $\sigma(\mathrm{A}) = \mathrm{A}$. By uniqueness of mystery values, $\sigma(\alpha)$ must $= \alpha$. Thus, since α is in the fixed field of every automorphism over $\mathbb{Q}$, it cannot be irrational. □

We now know that mystery values cannot be irrational, but can we obtain every rational number as a mystery value? The answer is yes.

Theorem 4.6. [KWG10] $\mathfrak{C}(2) = \mathbb{Q} \cap [0,1]$. *That is, two parties with finite random oracles can robustly simulate only $\mathbb{Q}$-random oracles, but may obtain any rational mystery value.*
Proof. Corollary 4.5 proves $\mathfrak{C}(2) \subset (\mathbb{Q} \cap [0,1])$. Going the other way, I must provide a way for two parties to simulate a $\mathbb{Q}$-random oracle. See the second paragraph following Example 5.1 for this construction. □

Theorem 4.7. [KWG10] *Three or more people with finite random oracles can robustly simulate only $\bar{\mathbb{Q}}$-random oracles.* $\mathfrak{C}(\mathrm{p}) = \bar{\mathbb{Q}} \cap [0,1]\ \forall\ p \geq 3$.
Proof. Uses algebraic geometry and other techniques that are beyond the scope of this paper. See [KWG10].

5. Mystery Values and Matrix Bounding

We know that the mystery values that can be obtained from a bilinear form represented by a {0,1}-matrix are precisely the rational numbers between, and including, 0 and 1. However, given a rational number, say $\frac{3}{7}$, how can we come up with the appropriate matrix that produces this mystery value?

Example 5.1: It is clear that I_n has mystery value $\frac{1}{n}$. Indeed:

$$\begin{bmatrix} 1/n & 1/n & \dots & 1/n \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1^* \\ \vdots \\ x_{(n-1)}^* \\ 1 - x_1^* - \dots - x_{(n-1)}^* \end{bmatrix} = \frac{1}{n} =$$

$$\begin{bmatrix} x_1 & \dots & x_{(n-1)} & 1 - x_1 - \dots - x_{(n-1)} \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix}$$

Since $\frac{1}{n}$ is the probability of "heads" in this construction, exchanging the 1's and 0's will give us the probability of tails. Thus, I_n with its 0's and 1's exchanged gives a mystery value of 1 - $\frac{1}{n} = \frac{n-1}{n}$. Also, since each of these bits is randomly generated, their outcomes are independent, so we can obtain products of these mystery values. For instance, we can generate a bit with bias $\frac{3}{16}$ by first generating one with bias $\frac{1}{4}$ and then generating another with bias $\frac{3}{4}$. But does this account for every rational probability?

The following is a construction for obtaining any rational probability. Suppose we would like to produce a bit with bias $\frac{a}{b}$. If each party can sample from a probability space with b elements i.e. a b-sided die, then assign to each outcome f_1 and f_2 an integer $\in \{0,1,...,\mathrm{b}\}$. Add the two outcomes mod b, say $\mathrm{s} = (f_1 + f_2)$ mod b. If $\mathrm{s} < \mathrm{a}$, call the result "heads" and if $\mathrm{a} \leq \mathrm{s} < \mathrm{b}$, call the result "tails." So, for example, an easy way to construct a matrix with a mystery value of $\frac{3}{5}$ would be to sample two 5-element probability spaces, take all

sums (a + b) mod 5, and assign 1's to sums that equal 0, 1, or 2, and 0's to sums that equal 3 or 4. The following matrix is obtained:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Notice there are three 1's in each row and column. In fact, any 5×5 matrix with this property will have $\frac{3}{5}$ as a mystery value.

Now we have a construction for a {0,1}-matrix which allows two people to generate a random bit with bias $\frac{a}{b}$. However, if for whatever reason, a bit of bias $\frac{82318}{1137925}$ is needed to be generated, this construction could be very inefficient, relying on constructing a 1137925×1137925 {0,1}-matrix with 82318 1's in each row and column. It is natural to ask the following question.

Mystery Value Bounding Problem: Is it possible for two people to generate a random bit with bias $\frac{a}{b}$ using a {0,1}-matrix whose dimensions are not each as large as b? In other words, is the denominator of the bias a lower bound on the dimensions of the {0,1}-matrix used to generate a random bit with that bias?

Recall that if α is a mystery value of the n×n bilinear form A, then Det(αJ-A) = 0. Det(αJ-A) is a polynomial (analogous to the characteristic polynomial) of degree n in α, where the coefficients of each term range from -n to n. Since our mystery value α is a solution to this equation, we have the following:

Mystery Value Bounding Problem Recasted: Let p be an n-degree polynomial with integer coefficients ∈ {-n,. . .,-1,0,1,. . .,n} and let $\rho_1, \rho_2, \ldots, \rho_m = \frac{r_1}{s_1}, \frac{r_2}{s_2}, \ldots, \frac{r_m}{s_m}$ be the rational roots of p. Is there some $\frac{r_i}{s_i}$ such that $s_i < n$?

Thus, a matrix bounding problem about finite random oracle simulation has been transformed into a number theoretic problem about the denominators of rational roots to a certain polynomial. If it could be proven that any n-degree polynomial with integer coefficients ∈ {-n,. . .,-1,0,1,. . .,n} can be expressed as the determinant of some n×n matrix with {0,1} entries, it would follow that any such polynomial can only contain one rational root between 0 and 1. This would-be analogue to the Perron-Frobenius Theorem is yet another sign that eigenvalues and mystery values may somehow be closely related, a possibility that could open the door to alternate proofs of theorems in linear algebra. And although this approach was not explored in this paper, viewing the {0,1} matrices as adjacency matrices may very well yield applications to directed graph theory.

6. Acknowledgements

It is my pleasure to thank my mentor, John Wiltshire-Gordon, for his endless patience and guidance throughout the REU. I'd also like to thank Benson Farb and Amie Wilkinson for their hospitality, advice, and help. Finally, a thanks to all of the University of Chicago faculty and students who participated in the REU this summer, without whom none of us (undergraduates) could have had this amazing opportunity.

References

[1] Gene S. Kopp and John D. Wiltshire-Gordon. When can several people simulate a private random oracle for someone else? http://arxiv.org/abs/1009.4188
[2] Yehuda Lindell. On the Composition of Secure Multi-Party Protocols. www.cs.biu.ac.il/ lindell/PAPERS/thesis.ps
[3] Joel Kamnitzer. Tensor Products. http://www.math.toronto.edu/jkamnitz/courses/mat247/tensorproducts2.pdf
[4] Mike Hill. Math 5651, Lecture 16: Tensor Products. http://people.virginia.edu/ mah7cd/Math5651/Lecture16.pdf