# Project Report 6.5610
# Generalizing Yao's XOR Lemma from Multicalibration

Rohan Goyal* rohan_g@mit.edu
Jaehyun Koo* koosaga@mit.edu
John Kuszmaul* john.kuszmaul@gmail.com
Alex Luchianov* lknv@mit.edu

Spring 2025

## 1 Introduction

### 1.1 Yao's XOR Lemma

Yao's XOR Lemma [Yao82, GNW11] is an important result in cryptography, showing that the simple operation of XOR can be used as a powerful tool for hardness amplication. This lemma lays the foundation for many cryptographic constructions, which we will visit later. We first begin by presenting the statement of Yao's XOR Lemma:

**Definition 1.1** ($\delta$-hard functions)**.** *A function $f : \{0,1\}^n \to \{0,1\}$ is called $\delta$-hard, if for any circuit $C$ of size at most $\mathsf{poly}(n)$,*

$$Pr_{x\in\{0,1\}^n}[C(x) \neq f(x)] \geq \delta$$

Note that every function is at most 1/2-hard. A trivial circuit that ignores the input $x$ and evaluates a majority of $f(x)$ will guess $f$ with at least 1/2

*MIT CSAIL

probability. In that sense, one can consider $\delta$-hard functions as the one where an adversary cannot have any *advantage* over $\frac{1}{2} - \delta$, where the *advantage* is the probability it can correctly guess a single bit minus 1/2.

**Definition 1.2** ($\delta$-hardcore functions)**.** *A function $f : \{0,1\}^n \to \{0,1\}$ is called $\delta$-hardcore, if for any circuit $C$ of size at most $\mathsf{poly}(n)$,*

$$Pr_{x \in \{0,1\}^n}[C(x) \neq f(x)] \geq 1 - \delta$$

**Lemma 1.3** (Yao's XOR Lemma, informal)**.** *Let $f$ be a $\delta$-hard function. Let $g : (\{0,1\}^n)^k \to \{0,1\}$ be the following:*

$$g(x_1, x_2, \ldots, x_k) = \bigoplus_{i=1}^{k} f(x_i)$$

*Then, for all $\epsilon > 0$, $g$ is $\frac{1}{2} - \frac{1}{2}(1-2\delta)^k - \epsilon$ hard.*

It's worth noticing that the bound of the lemma is essentially tight. Consider a $\delta$-hard function $f$ with an adversary circuit $C$ that guesses $f$ with probability almost $1 - \delta$. Using this circuit, we guess $g$ by taking the XOR of all guesses $f(x_i)$. The probability that this circuit guesses $g$ equals the probability where it fails to guess $f(x_i)$ by an even number of times, which is:

$$\begin{aligned} &\sum_{i=0}^{k/2} \binom{k}{2i} \delta^{2i}(1-\delta)^{k-2i} \\ &= \frac{1}{2}(((1-\delta)+\delta)^k + ((1-\delta)-\delta)^k) \\ &= \frac{1}{2}(1 + (1-2\delta)^k) \end{aligned}$$

Hence, $g$ can be at most $\frac{1}{2} - \frac{1}{2}(1-2\delta)^k$ hard.

## 1.2 Applications in Cryptography

Yao's XOR Lemma is a fundamental result of complexity theory and has several important cryptographic applications. Below, we list some of its important applications to show the wide implications of Yao's XOR Lemma.

**Hardness Amplification.** The most general implication of Yao's XOR Lemma is that one can take a slightly hard function and construct a function arbitrarily close to 1/2-hard by simply taking an XOR of $f(x_i)$. As seen earlier, no function can be harder than 1/2, so this lemma is a way to construct a very hard random function with relatively simple ingredients. In general, this concept is known as *hardness amplification* and is very useful in cryptography, where the notion of security is based upon hardness. In that sense, hardness amplification is a security amplification, and Yao's XOR Lemma implies one can achieve near-perfect security with mildly secure functions and simple primitives.

**Hardcore Predicates.** Let $f : \{0,1\}^n \to \{0,1\}^n$ be a permutation. A hard-core predicate for $f$ is a function $B : \{0,1\}^n \to \{0,1\}$ such that for any PPT adversary $\mathcal{A}$ can predict $B(x)$ with at most $1/2 + \mathsf{negl}(n)$ probability, given the value $f(x)$.

Given any one-way permutation, Yao's lemma provides a way to construct a hardcore predicate from a permutation. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation. In the function $f$, there is a bit $i \in [n]$ such that any PPT adversary can guess with at most $1 - \frac{1}{n} + \mathsf{negl}(n)$ probability - otherwise, using union bound, we can invert a permutation. This bit doesn't have a significant security per se, but Yao's XOR Lemma can be used to remove the advantage.

Let $g(x_1, x_2, \ldots, x_m) = f(x_1)||f(x_2)||\ldots||f(x_m)$, where $a||b$ denotes the concatenation of $a$ and $b$. $g$ is a permutation, and each of the $i$-th bit of $f$ is $\frac{1}{n} - \mathsf{negl}(n)$ hard. If $m \geq \Omega(n \log n)$, Yao's lemma implies that the XOR of all $i$-th bits in $\{f(x_j)\}_{j=1}^m$ will be $\frac{1}{2} - \mathsf{negl}(n)$ hard. From this, we can construct a hardcore predicate by simply taking the XOR of all $i$-th bits in each of the $f(x_j)$s.

**Pseudo Random Number Generators.** In the previous paragraph, we showed how to generate a permutation that contains a hardcore predicate using Yao's lemma. Given a hardcore predicate $B$ and its associated permutation $f$, Blum and Micali [BM82] proposed a very simple scheme to generate a pseudo-random number generator, as follows:

- In initialization stage, sample a random seed $S$ from $\{0,1\}^n$.
- For each query, output $B(S)$ as a random bit, and set $S \leftarrow f(S)$.

This simple scheme, commonly known as the Blum-Micali Generator, is one concrete application of Yao's lemma in practical cryptographic primitives.

## 1.3 IHCL

As the previous section shows, Yao's XOR Lemma is fundamental in cryptography. However, the proof of this lemma is quite technical. The most popular, although not original, proof of the XOR lemma is done by Impagliazzo's Hard-Core Lemma, which we abbreviate as IHCL.

**Theorem 1.4** (IHCL, informal, [Imp95,Hol05])**.** *Let $\mathcal{F}$ be a family of boolean functions on $\mathcal{X}$, $\epsilon > 0$, and $g : \mathcal{X} \to \{0,1\}$ a function that is $(\mathcal{F}', \delta)$-hard, meaning that $\Pr[f(x) \neq g(x)] \geq \delta$ for all functions $f$ that have "low complexity" relative to $\mathcal{F}$. Then there exists a set $H \subseteq \mathcal{X}$ of size at least $2\delta|\mathcal{X}|$ such that $g$ is $(\mathcal{F}, 1/2 - \epsilon)$-hard on $H$.*

Note that Definition 1.1 is replaced with the notion of $(\mathcal{F}', \delta)$-hard, which more explicitly states the computation bound as a circuit with a certain size restriction that can evaluate any function on $\mathcal{F}$. The formal definition of the hardness will be presented later in Section 2.

Intuitively speaking, IHCL implies that every function with difficulty $\delta$ has a *hardcore* of size $2\delta|\mathcal{X}|$ which is very hard to compute for an adversary. Note that this definition of hardcore is not the same as that of hardcore predicates, although they share a similarity in that the adversary does not have an advantage.

Theorem 1.4 was first proved in [Imp95] with a looser lower bound of $|H| \geq \delta|\mathcal{X}|$, and it was improved to the optimal lower bound of $|H| \geq 2\delta|\mathcal{X}|$ in [Hol05]. This optimal lower bound is required in proving Yao's XOR Lemma, as in the form of Lemma 1.3.

Given this optimal lower bound of [Hol05], we can prove Yao's lemma. By Theorem 1.4, for a $\delta$-hard function there is a *hardcore* of size at least $2\delta \cdot 2^n$. This means, for a random $x_i$, the probability that it is inside the hardcore is at least $2\delta$, and since each $x_i$ is independent, the probability that an input does not lie in any of the hardcore is at most $(1-2\delta)^k$. For all other $1-(1-2\delta)^k$ portion of inputs, they have at least one input $i \in [k]$ such that $x_i$ belongs to the hard-core of $i$. Since $f(x_i)$ is $\frac{1}{2}-\epsilon$ hard, whatever advantage it gathered on $\oplus_{j\neq i} f(x_j)$ is nullified by this unpredictable bit from $f(x_i)$. This establishes $\frac{1}{2} - \epsilon'$ hardness over $1-(1-2\delta)^k$ portion of inputs, which gives us a desired bound.

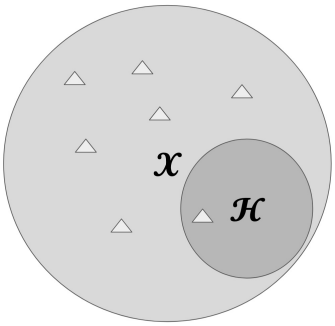


Figure 1: *If a given function has a hardcore (denoted as $\mathcal{H}$), then one of the input $x_i$ will hit a hardcore with probability at least $1-(1-2\delta)^k$. Since $f(x_i)$ in a hardcore is hard to guess, the XOR-ed function is hard as well.*

## 1.4 Our contributions

In the recent work of Casacuberta, Dwork, and Vadhan [CDV24], they introduced IHCL++, which is a stronger and more general version of IHCL. As we've seen that Theorem 1.4 implies Lemma 1.3, a natural question is whether the improvement made in IHCL++ could be applied to Yao's lemma. We answer this affirmatively, which we call as Yao++.

The key improvement made in Yao++ is to replace the notion of *hard functions* with a *partition* instead. In IHCL, we needed an assumption that the given function should have a certain degree of hardness - this is a concept that is somewhat unwieldy to deal with, given that it is notoriously hard to prove the incomputability of functions. This was improved in IHCL++ where we don't need any assumptions about the hardness of the function. Instead, for any given function $g : \mathcal{X} \to \{0,1\}$, IHCL++ shows that a partition $\mathcal{P}$ exists where individual partitions are either negligibly small or hard to guess - the *hardness* guarantee in IHCL++ comes from a *balance parameter* $b_P = \min(E_{x\in P}[g(x)], 1 - E_{x\in P}[g(x)])$, where $P \in \mathcal{P}$. In other words, if the expected value of the given function inside $P$ is close to $1/2$, the hardness guarantee follows.

In our work, we define Yao++ in a way that it captures the notion of balanced parameter in IHCL++. Then, we prove Yao++ in a self-contained

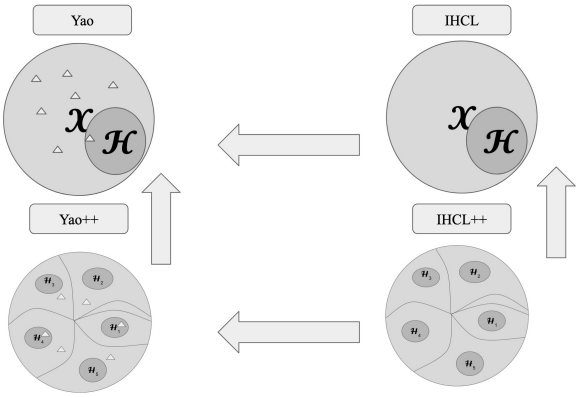


Figure 2: *A diagram that shows the relation between Yao, IHCL, Yao++, IHCL++. Arrows denote implication.*

way, and show that Yao++ implies Yao's lemma. Our proof is self-contained and considerably shorter than the other proofs, specifically those that rely on IHCL [Imp95, Hol05] and are thus heavily involved. As a result, our Yao++ has the additional advantage of being a concise proof of the well-known cryptographic theorem.

# 2 Recent Progress (Multicalibration + Complexity)

Recent progress on multicalibration has allowed for generalizations of several theorems. In particular, Casacuberta, Dwork, and Vadhan [CDV24] showed in 2024 stronger and more general version of Impagliazzo's Hardcore Lemma (IHCL), the Dense Model Theorem, and characterizations of pseudoentropy. The original characterizations of these results all can be proved using the Regularity lemma [TTV09]. However, instead of using the Regularity lemma as the central starting point, the work of [CDV24] instead explores how we can approach these results using the Multicalibration theorem [HJKRR18] instead.

In order to understand these recent results, we begin with the Multicalibration theorem. The theorem, as stated below, is verbatim from [CDV24]. In particular, it is presented in the language of partitions.

**Theorem 2.1** (Theorem 2.1 (Multicalibration Theorem) from [CDV24]. Originally from [HJKRR18]). *Let $\mathcal{X}$ be a finite domain, $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \{0, 1\}$, $g : \mathcal{X} \to [0, 1]$ an arbitrary function, $\mathcal{D}$ a probability distribution over $\mathcal{X}$, and $\varepsilon, \gamma > 0$. There exists a partition $\mathcal{P}$ of $\mathcal{X}$ such that:*

1. *$\mathcal{P}$ has $O(1/\varepsilon)$ parts.*
2. *$\mathcal{P}$ has "low complexity" relative to $\mathcal{F}$. Specifically, there is a boolean circuit $C : \mathcal{X} \to [k]$ (i.e., with gates of fan-in at most 2 and $\lceil \log |\mathcal{X}| \rceil$ input gates and $\lceil [k] \rceil$ output gates) of size poly$(1/\varepsilon, 1/\gamma, \log |\mathcal{X}|)$ with $O(1/\varepsilon^2)$ oracle gates instantiated with functions from $\mathcal{F}$ such that the $\mathcal{P} = \{C^{-1}(1), \ldots, C^{-1}(k)\}$.*
3. *$\mathcal{P}$ is $(\mathcal{F}, \varepsilon, \gamma)$ multicalibrated for $g$ on $\mathcal{D}$: that is, for all $f \in \mathcal{F}$ and all $P \in \mathcal{P}$ such that $\Pr_{x\sim\mathcal{D}}[x \in P] \geq \gamma$, we have*

$$\left|\mathbb{E}_{x\sim\mathcal{D}|_P}[f(x) \cdot (g(x) - v_P)]\right| \leq \varepsilon. \tag{1}$$

*where $v_P := \mathbb{E}_{x\sim D|_P}[g(x)]$ and $\mathcal{D}|_P$ denotes the conditional distribution $\mathcal{D}|_{h(x)\in P}$.*

We highlight that unlike IHCL, this new IHCL++ is assumptionless (i.e., there is no $\delta$-hard assumption in the theorem statement). As we will see below, it turns out that IHCL++ is also strictly more general; IHCL++ implies IHCL. One final advantace of IHCL++ over IHCL, is that using the techniques of multicalibration, the proof of IHCL++ is relatively simple (though still very far from trivial) compared to past proofs of IHCL [CDV24].

## 2.1 New Method to get IHCL

The Multicalibration theorem was used by [CDV24] to prove a new, more general variation of Impagliazzo's Hardcore Lemma (IHCL). [CDV24] refer to this new theorem as IHCL++. We include the theorem below in its verbatim form.

In order to understand the significance of IHCL++, let's unpack some of the terminology used in the theorem statement. First, $\eta_P := \Pr_{x\sim D}[x \in P]$ dentotes the ***size parameter*** of a piece $P$ of the partition.

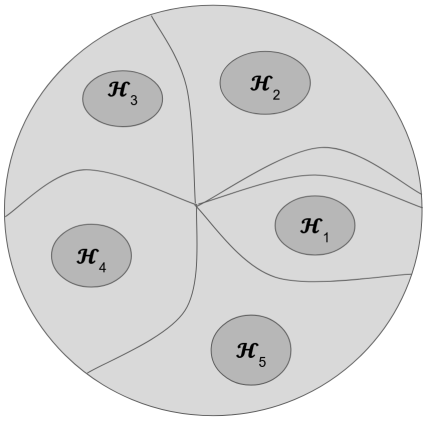


Figure 3: *A diagram depicting the hardcore distribution within each of the partitions in IHCL++.*

**Definition 2.2** (verbatim from [CDV24])**.** *Given a set of functions $\mathcal{F} = f$ on a finite domain $\mathcal{X}$, $\boldsymbol{\mathcal{F}_{t,q,k}}$ denotes the class of partitions $\mathcal{P}$ of $\mathcal{X}$ such that there exists $\hat{f} \in \mathcal{F}_{t,q}$, $\hat{f} : \mathcal{X} \to [k]$, satisfying $\mathcal{P} = \{\hat{f}^{-1}(1), \ldots, \hat{f}^{-1}(k)\}$.*

In turn, recall that $\mathcal{F}_{t,q}$ is the set of functions that can be computed by a circuit of size at most $t$ with at most $q$ oracle calls to the family of functions $\mathcal{F}$.

For completeness, we also include a formal definition of $(\mathcal{F}, \delta)$-hardness, which is the verbatim definition used by [CDV24].[1]

**Definition 2.3** (formal definition of $(\mathcal{F}, \delta)$-hardness, verbatim from [CDV24])**.** *Given a class $\mathcal{F}$ of randomized functions $f : \mathcal{X} \to \{0,1\}^\ell$, a distribution $\mathcal{D}$ on $\mathcal{X}$, an arbitrary randomized function $g : \mathcal{X} \to \{0,1\}^\ell$, and $\delta > 0$, we say that $g$ is $(\mathcal{F}, \delta)$-hard on $\mathcal{D}$, if for all $f \in \mathcal{F}$,*

$$\Pr_{x \sim D}[f(x) = g(x)] \leq 1 - \delta.$$

Note that the randomness in the probability is also drawn over the coins used by the randomized functions $f$ and $g$. Further, note that for $\ell = 1$, the hardness cannot be greater than $1/2 - \varepsilon$ (as a coin toss can achieve $\delta = 1/2$).

[1] Some other works in the area use the alternate notation of $\delta'$-hardness, where $\delta'$ denotes the adversary's advantage over a random coin toss. In our notation, this denotes a $1/2 - \delta$ hardness.

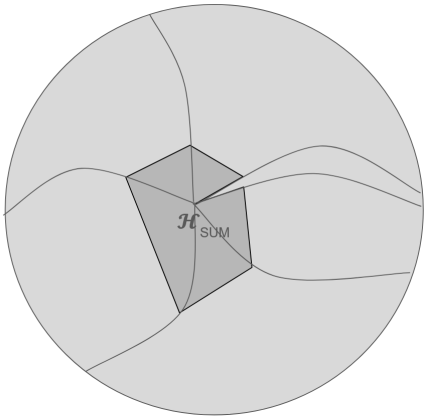


Figure 4: *A visual depiction of the argument that IHCL++ implies IHCL. We obtain one large hardcore set by combining the hardcore sets from each piece of the partition.*

**Theorem 2.4** (Theorem 3.2 (IHCL++) from [CDV24])**.** *Let $\mathcal{X}$ be a finite domain, let $\mathcal{F}$ be a family of functions $f : \mathcal{X} \to [0,1]$, let $g : \mathcal{X} \to [0,1]$ be an arbitrary function, $\mathcal{D}$ a probability distribution over $\mathcal{X}$, and let $\varepsilon, \gamma > 0$. There exists a partition $\mathcal{P} \in \mathcal{F}_{t,q,k}$ of $\mathcal{X}$ with $t = O(1/(\varepsilon^4\gamma) \cdot \log(|\mathcal{X}|/\varepsilon))$, $q = O(1/\varepsilon^2)$, $k = O(1/\varepsilon)$ which satisfies that for all $P \in \mathcal{P}$ such that $\eta_P \geq \gamma$, there exists a distribution $\mathcal{H}_P$ in $P$ of density $2b_P$ in $D|_P$ such that $g^{rand}$ is $(\mathcal{F}^{rand}, 1/2 - \frac{\varepsilon}{2b_P(1-b_P)})$-hard on $\mathcal{H}_P$.*

## 2.2 Proof that IHCL++ implies IHCL

Using IHCL++, we can actually recover the original IHCL theorem, thus making IHCL++ more general [CDV24]. As [CDV24] notes, "the key idea is to 'glue together' the hardcore distributions $\mathcal{H}_P$ within each $P \in \mathcal{P}$, where in this gluing together each $p \in \mathcal{P}$ is weighted acccording to its size parameter $\eta_P$ of the set $P$."

The first step of their analysis is to prove the following proposition.

**Proposition 1** (Proposition 3.3 from [CDV24])**.** *Let $\mathcal{X}, \mathcal{D}, \mathcal{F}, g, \varepsilon, \gamma, \mathcal{P}, t, q, k$ as in Theorem 2.4. Moreover, assume that $g$ is $(\mathcal{F}_{t+k,q}, \delta)$-hard, and suppose that $\eta_P \geq \gamma$ for all $P \in \mathcal{P}$. Then,*

$$\mathbb{E}_{P \sim \mathcal{P}(D)}[b_P] \geq \delta.$$

**Intuitive Understanding of Proposition 1.** Proposition 1 essentially seeks to understand IHCL++ while applying the assumption that $g$ is $\delta$-hard. Using this assumption, we would ultimately like to formally recover IHCL. For the full proof, see Section 3.3 of [CDV24], however, here we will sketch an intuitive understanding of the key ideas. We can understand Proposition 1 as claiming that the expectation of the balance parameter of a partition, sampled uniformly from the elements of $\mathcal{X}$, is at least $\delta$. This lower bound means that the partitions on (a weighted) average, cannot be too imbalanced (this means each piece of the partition is itself hard). The remainder of the proof follows via combining the hardcore distributions from each partition that is sufficiently imbalanced and with sufficiently large size parameter to recover the original IHCL theorem.

In the following section, we will see a similar argument to show how Yao++ implies Yao.

# 3 Version of Yao without assumptions

**Lemma 3.1** (Hardcoreness of function by sampling its components)**.** *The hardcoreness of a function $f$ such that $f$ can be partitioned into $P_1, P_2, \dots P_k$ with densities $\delta_1, \delta_2, \dots \delta_k$ and respective hardcorenesses $\epsilon_1, \epsilon_2, \dots \epsilon_k$ is $\epsilon_f$ where:*

$$\epsilon_f \leq \sum_{i=1}^{k} \delta_i \epsilon_i$$

**Lemma 3.2** (Hardcoreness of sampling $k$ functions and XOR-ing them)**.** *Let there be functions $f_1, f_2 \dots f_k$ and random variables $X_1, X_2 \dots X_k$ sampled according to some arbitrary distributions $D_1, D_2, \dots D_k$ such that $f_i(X_i)$ is $\epsilon_i$-hardcore. Then, the hardcoreness of a function $f$ such that $f(X_1, X_2, \dots X_k) = f_1(X_1) \oplus f_2(X_2) \oplus \cdots \oplus f_k(X_k)$ is $\epsilon_f$ where:*

$$\epsilon_f = \min(\epsilon_1, \epsilon_2, \dots \epsilon_k)$$

*Proof.* Assume otherwise. Then, there must be a circuit C such that:

$$\Pr_{X_i \sim D_i} [C(X_1, X_2, \dots, X_k) = f(X_1, X_2, \dots, X_k)] > 1 + \epsilon_f$$

Let us fix all other inputs, except for $X_{\epsilon-min}$ where $\epsilon - min$ is chosen so that $\min(e_1, \ldots \epsilon_k) = \epsilon_{\epsilon-min}$. Then, by averaging over all tuples of fixed inputs, we know that there must exist a tuple of fixed inputs such that:

$$\Pr_{\substack{X_{\epsilon\text{-min}} \sim D_{\epsilon\text{-min}} \\ X_i \text{ fixed for } i \neq \epsilon\text{-min}}} [C(X_1, X_2, \ldots, X_k) = f(X_1, X_2, \ldots, X_k)] > 1 + \epsilon_f$$

Since everything but $\epsilon - min$ is fixed this means that we have created a circuit capable of computing $f_{\epsilon-min}$ with higher than $\epsilon_{\epsilon-min}$ accuracy.

□

## 3.1 Yao++ Lemma

As the actual formulation of our theorem is quite verbose, we shall split it into a series of definitions that lead to the main body of the theorem.

We shall begin by naming the type of structure that can result from the application of the IHCL++ theorem:

**Definition 3.3** (Partitioned Hardcore Structure)**.** *Let $\mathcal{X}$ be a finite domain, and $\mathcal{F}$ a family of Boolean functions $f : \mathcal{X} \to \{0,1\}$. Let $g : \mathcal{X} \to \{0,1\}$ be a function, and let $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ be a partition of $\mathcal{X}$ such that for each part $P_i \in \mathcal{P}$, there exists a hardcore subset $\mathcal{H}_{P_i} \subseteq P_i$ of density $2B_{P_i}$ such that $g^{\text{rand}}$ is $(\mathcal{F}^{\text{rand}}, \epsilon_i)$-hardcore over $\mathcal{H}_{P_i}$.*

In order to better manipulate Partitioned Hardcore Structures, we wish to define a simplified structure that represents the same object, however is better tailored for our task:

**Definition 3.4** (Simplified Partitioned Hardcore Structure)**.** *We define $\delta_i$ a measure of the proportion of the total domain that lies in the hardcore set of the $i^{th}$ component. Formally, we let $\delta_i = 2B_{P_I} \cdot \frac{|P_i|}{|X|}$.*

*So that all $\delta_i$ add to 1, we create a virtual element that represents union of the* not *hardcore sections of each component in the partition. Formally, we let*

$$\epsilon_0 = \frac{1}{2} \quad \delta_0 = 1 - \sum_{i=1}^{N} \delta_i$$

Note that our Simplified Partitioned Hardcore Structure is merely a partition of the domain such that each component has a specified hardcoreness.

The difference between this Simplified Hardcore Structure and the Partitioned Hardcore Structure is that we aggregate all non-hardcore parts into a single component and that we explicitly define the densities of the components in relation to the total domain.

**Definition 3.5** (Prefix Aggregate)**.** *For Simplified Partitioned Hardcore Structure let $\epsilon_0, \ldots, \epsilon_N$ be ordered such that $\epsilon_{i-1} \geq \epsilon_i$. We define the prefix aggregate:*

$$\overline{\delta_i} = \sum_{j=0}^{i} \delta_j$$

*which represents the total mass of all components with hardcoreness at least $\epsilon_i$.*

**Theorem 3.6** (Yao++)**.** *For Simplified Partitioned Hardcore Structure let $\epsilon_0, \ldots, \epsilon_N$ be ordered such that $\epsilon_{i-1} \geq \epsilon_i$. Then the function $g^{\oplus k}$ is $\epsilon_{\oplus k}$-hard, with*

$$\epsilon_{\oplus k} \leq \sum_{i=0}^{N} \left( \overline{\delta_i}^k - \overline{\delta_{i-1}}^k \right) \cdot \epsilon_i$$

*Proof.* The above formula is obtained by sampling $k$ elements and taking the average of their maximum hardcoreness.

The Yao++ lemma can be easily derived by conditioning in what hardcore set each sample falls in, and adding all of those hardcorenesses up by the Lemma 3.1.

For a certain conditioning of samples we can compute the hardcoreness of the scenario using Lemma 3.2. □

## 3.2 Yao++ lemma cannot be improved by redistribution

For a given function $f$, there are several possible ways to partition it, each partition provides a way to estimate the hardcoreness of $f$. Given that different partitions can suggest different hardcoreness-es, we wish to prove that we cannot improve the estimated hardcoreness of a function by creating an *artificial* partition of the elements.

By *artificial* we mean a partition that is not constructed according to some direct knowledge of $f$, but rather knowledge of another partition. It

is quite intuitive why a sound hardcoreness estimator should not increase by *artificial* repartitioning, as the original partition has the highest degree of information and any repartitioning either retains the initial information or dilutes it.

**Definition 3.7** (Repartition)**.** *Given a partition of a function $f$ into components with densities $\delta_0, \delta_1, \ldots, \delta_n$ and hardcoreness values $\epsilon_0, \epsilon_1, \ldots, \epsilon_n$, a* repartition *is a new partition with component densities $\delta'_0, \delta'_1, \ldots, \delta'_m$ and hardcoreness values $\epsilon'_0, \epsilon'_1, \ldots, \epsilon'_m$, such that the new components are obtained as linear combinations of the original components.*

*Formally, there exists an $n \times m$ matrix $A$ such that:*

1. *Each row of $A$ sums to* 1*:*

$$\sum_{i=0}^{m} A_{j,i} = 1 \quad \textit{for all } j = 0, \ldots, n.$$

2. *The new densities are:*

$$\delta'_i = \sum_{j=0}^{n} \delta_j A_{j,i}.$$

3. *The new hardcoreness values are:*

$$\epsilon'_i = \sum_{j=0}^{n} \frac{\delta_j A_{j,i}}{\delta'_i} \epsilon_j.$$

It is quite obvious that more complex repartitioning procedures can be done by successive repartitioning steps. We claim that any repartitioning can be decomposed into *elementary* steps that merely combine elements some of the elements from two components into the elements of a new component. This is equivalent to saying that any repartitioning matrix can be written as the product of several repartitioning matrices where each matrix has most diagonal elements equal to 1.

Thus, in order to prove that any repartitioning results in a less tight bound, it is sufficient to prove that any *elementary* repartitioning results in a less tight bound.

Let us consider such an *elementary* step that combines a proportion $a$ of elements from a component $i$ with a proportion $b$ of elements from a component $j$. Without loss of generality, we assume that $i < j$:

$$\delta'_j \leftarrow \delta_j - b$$
$$\delta'_i \leftarrow \delta_i - a$$
$$\delta'_{new} = a + b$$
$$\epsilon'_{new} = \frac{a}{a+b} \cdot \epsilon_i + \frac{b}{a+b} \cdot \epsilon_j$$

Let us a build a new partition described by $\delta'_i$ and $\epsilon'_i$ according to the above operation. Let this new partition have hardcoreness $\epsilon'_{\oplus k}$, we claim that:

$$\epsilon_{\oplus k} \leq \epsilon'_{\oplus k}$$

In order to prove the claim, let us first assume the opposite. That is: $\epsilon_{\oplus k} > \epsilon'_{\oplus k}$.

Note that for $k$ given samples, the different partitions often have the same hardcoreness, only when the different partitions result in different components for the samples we actually get different hardcorenesses . Let us condition upon that fact.

We will be as pessimistic as possible and first condition on the fact that we sampled $k$ elements such that the element with minimum hardcoreness is one of the elements from the new component. Let us assume that $k_2$ of the sampled elements are from the new component.

We will be even more pessimistic and assume that the second minimum hardcoreness is worse than $\delta_a$.

Then it becomes trivial to prove mathematically:

$$\epsilon_{new} \geq \epsilon_i \cdot \left(\frac{a}{a+b}\right)^k + \epsilon_j \cdot \left(1 - \left(\frac{a}{a+b}\right)^k\right)$$

$$\epsilon_i \cdot \frac{a}{a+b} + \epsilon_j \cdot \frac{b}{a+b} \geq \epsilon_i \cdot \left(\frac{a}{a+b}\right)^{k_2} + \epsilon_j \cdot \left(1 - \left(\frac{a}{a+b}\right)^{k_2}\right)$$

Which is clearly true as: $\epsilon_i \geq \epsilon_j$. Note that for $k_2 = 1$ we have equality, but in that case it simply does not matter how we partition the elements

### 3.3 Why Stronger Yao Lemma implies the Weaker version

Note that in order to recover IHCL from IHCL++, we "glued" together the hardcore components. This operation was merely a repartitioning of the partition resulting from IHCL++. The previous section proves why any such repartitioning will be weaker in terms of hardcoreness.

## 4 Conclusion and Future Directions

Putting everything together, we consider [CDV24]'s improvement of Impagliazzo's hard core lemma to IHCL++ and following the proof of IHCL $\implies$ Yao [GNW11], we get a result of the form IHCL++ $\implies$ Yao++. This conclusively presents a new concrete application of the ++ approach outlined in the paper as suggested in the final section of [CDV24]. Along with the improvement in Yao's result, we give a simple exposition for applications of multicalibration in complexity.

Thus, our result presents a generalization of Yao's fundamental result through connections to the multicalibration theorem as done in [CDV24]. It would be interesting to see if this generalization translates into applications of Yao's lemma and lead to concrete theoretical improvements.

A few possible next applications of the ++ approach as detailed in [CDV24] are in leakage resilient cryptography [JP14, CCL18], Chang's inequality in Fourier analysis of boolean functions [IMR12], and weak notions of zero-knowledge [CLP15]. This approach was also shown to be helpful in characterizing the distinguishability of product distributions in [MPV25].

Furthermore, it would be interesting to explore if there are other unexplored connections in the areas of algorithmic fairness, complexity theory, and cryptography.

## 5 Acknowledgements

We would like to thank the our group's TA Katarina Cheng, as well as the instructors Henry Corrigan-Gibbs and Yael Kalai for their help and sup-

port throughout this project. We would also like to thank Aaron (Louie) Putterman for introducing us to the techniques used in the recent works on complexity theory and multicalibration.

## Team contributions

Since this work was theoretical, it is hard to divide the content of mathematical work carried by the team members so we believe that all members of the team contributed equally. Each section in the report and slides are primarily due by the following team members:

- section 1 by Jaehyun,
- section 2 by John,
- section 3 by Alex,
- conclusion by Rohan, and
- the presentation slides by Alex.

# References


[BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 112–117, 1982.

[CCL18] Yi-Hsiu Chen, Kai-Min Chung, and Jyun-Jie Liao. On the complexity of simulating auxiliary input. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 371–390, Cham, 2018. Springer International Publishing.

[CDV24] Sílvia Casacuberta, Cynthia Dwork, and Salil Vadhan. Complexity-theoretic implications of multicalibration. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1071–1082, 2024.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 66–92. Springer, 2015.

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 273–301, 2011.

[HJKRR18] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

[Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 664–673, 2005.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 538–545. IEEE, 1995.

[IMR12] Russell Impagliazzo, Cristopher Moore, and Alexander Russell. An entropic proof of chang's inequality. *CoRR*, abs/1205.0263, 2012.

[JP14] Dimitar Jetchev and Krzysztof Pietrzak. How to fake auxiliary input. In *Theory of Cryptography Conference*, pages 566–590. Springer, 2014.

[MPV25] Cassandra Marcussen, Aaron L. Putterman, and Salil Vadhan. Characterizing the distinguishability of product distributions through multicalibration. *arXiv preprint arXiv:2412.03562*, 2025. Version 2, February 25, 2025.

[TTV09] Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Regularity, boosting, and efficiently simulating every high-entropy distribution. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 126–136, 2009.

[Yao82] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.