# Finding Pessiland

Hoeteck Wee *

Computer Science Division
University of California, Berkeley
hoeteck@cs.berkeley.edu

**Abstract.** We explore the minimal assumptions that are necessary for non-trivial argument systems, such as Kilian's argument system for NP with poly-logarithmic communication complexity [K92]. We exhibit an oracle relative to which there is a 2-round argument system with poly-logarithmic communication complexity for some language in NP, but no one-way functions. The language lies outside $\mathsf{BPTime}(2^{o(n)})$, so the relaxation to computational soundness is essential for achieving sublinear communication complexity. We obtain as a corollary that under black-box reductions, non-trivial argument systems do not imply one-way functions.

## 1 Introduction

Pessiland, coined by Impagliazzo [I95], is a world in which there are hard-on-average languages in NP but no one-way functions. In Pessiland, generating hard instances of NP-languages is easy, but we do not know of a way of exploiting these hard-on-average problems in cryptography. In fact, Impagliazzo and Luby [IL89] proved that most cryptographic applications, including bit commitment, private-key encryption and digital signatures, require one-way functions (which allow us to generate hard instances of NP-languages along with a witness) and are therefore impossible to realize in Pessiland.

Recently, Barak's construction of (non-black-box) zero-knowledge arguments [B01] renewed interest in the round complexity and the minimal assumptions necessary for the existence of non-trivial argument systems for NP and NEXP [K92,M00,BG02,W05]. We consider an argument system for NP or NEXP to be non-trivial if the communication complexity is subpolynomial in the length of the witness. Currently, the best construction for NEXP is a 4-round protocol based on the existence of (standard) collision-resistant hash functions [BG02]. If we could relax the assumption to one-way functions, then Barak's construction would yield a constant-round zero-knowledge

* Work supported by US-Israel BSF Grant 2002246. Presently visiting Tsinghua University, Beijing, China.

argument for NP under the same assumption. On the other hand, we do not even know if one-way functions are necessary for non-trivial argument systems. For 2-round argument systems, it is known that a relaxation of hard-on-average languages in NP is necessary [W05] (also, Appendix A.2).

### 1.1 Main results

In this work, we establish a connection between the two problems: we provide a relativized construction of Pessiland which contains a non-trivial 2-round argument system for a language in NP.

**Theorem 1.** *There exists an oracle relative to which there exists a strongly hard-on-average language in* $\mathsf{NP} \cap \mathsf{coNP}$*, but no one-way functions. Furthermore, there is a 2-round public-coin argument system with poly-logarithmic communication complexity for a language that lies within* NP *but outside* $\mathsf{BPTime}(2^{o(n)})$.

It is important that our argument system is for a language outside $\mathsf{BPTime}(2^{o(n)})$, as it means that the relaxation to computational soundness is essential for achieving sublinear communication complexity. This rules out trivial 2-round argument systems with poly-logarithmic communication complexity for languages in BPP or $\mathsf{NTime}(\log^2 n)$. In particular, a relativizing argument in [GH98] implies that languages outside $\mathsf{BPTime}(2^{o(n)})$ do not have interactive proof systems with sublinear (total) communication complexity, regardless of the number of rounds, and even if the verifier is allowed a polynomial amount of private randomness.

As a corollary, we deduce that there does not exist a black-box construction (such as those used in [V04,W05]) of one-way functions or collision-resistant hash functions from non-trivial 2-round argument systems. This partially explains why we have not been able to prove a statement of the form "if there exists a non-trivial 2-round argument system, then there exists one-way functions". In particular, a proof of this statement must use a non-relativizing argument or make some stronger assumptions on the underlying language. On the other hand, we do not expect to disprove this statement. Suppose non-trivial 2-round argument systems do not exist (which is quite plausible); then, the statement is vacuously true.

The black-box construction of primitives from interactive protocols in [V04,W05] only yields auxiliary-input primitives, as the input instance for the protocol is hard-wired into the algorithm computing the primitive. As such, one would ideally like to rule out auxiliary-input one-way functions (that is, we only require that the function be computable by a nonuniform polynomial-time algorithm) while exhibiting a non-trivial argument system. At this point, we are only able to achieve a much weaker result:

**Theorem 2.** *There exists an oracle relative to which there exists a strongly hard-on-average language in* NP*, but no auxiliary-input one-way functions.*

The analysis of our first construction is fairly straight-forward apart from some subtle details, and uses several techniques from previous work (such as [IR89,GT00]); the insight lies in the construction and in establishing a connection between Pessiland and non-trivial argument systems. Our second construction, on the other hand, requires a more intricate and novel analysis.

### 1.2 Perspective and related works

*Round-efficient argument systems.* All previous constructions of non-trivial argument systems (in the standard model) [K92,BG02] require 4 rounds and the existence of collision-resistant hash functions. Micali [M00] gave the first relativized construction of a non-trivial 2-round argument system, by using a random oracle to instantiate collision-resistant hash functions and the Fiat-Shamir paradigm in Kilian's 4-round protocol [K92]. While these previous constructions were for either $\mathsf{NP}$-complete or $\mathsf{NEXP}$-complete languages, our relativized construction (which does not require one-way functions or collision-resistant hash functions) is for a language in $\mathsf{NP}$ but possibly not $\mathsf{NP}$-complete. We stress that previous work [W05] deducing hard-on-average problems in $\mathsf{NP}$ from non-trivial argument systems for $\mathsf{NP}$ (and $\mathsf{NEXP}$) does not exploit the structure of $\mathsf{NP}$ in any way; it merely uses the fact $\mathsf{NP}$ does not have a proof system with the same communication complexity as the underlying argument system under standard complexity assumptions.

*Relationships between cryptographic primitives.* Starting with the work of Impagliazzo and Rudich [IR89], the study of relationships between cryptographic primitives has focused on the impossibility of basing complex primitives on simpler ones, particularly one-way functions and one-way permutations. Our main result goes in the reverse direction: it shows the impossibility of constructing simpler primitives from a specific cryptographic application (in a black-box manner). It also provides an example of a cryptographic application (for a contrived language, unfortunately) which may be based on weaker assumptions than the existence of one-way functions. In an unpublished work, Impagliazzo and Rudich gave the first[1] relativized construction of Pessiland, which yields a black-box separation between hard-on-average languages in $\mathsf{NP}$ and one-way functions.

[1] We only learnt about the work of Impagliazzo and Rudich after independently arriving at the same construction. We also clarify that *finding* in the title alludes to the search for constructions of Pessiland with stronger cryptographic implications (and a positive result for exploiting average-case hardness) than a mere separation between hard-on-average languages and one-way functions.

## 2 Preliminaries

We use $\Pi_\ell$ to denote the set of all permutations on $\{0,1\}^\ell$, $\mathcal{F}_{n,\ell}$ to denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^\ell$, and $U_n$ to denote the uniform distribution over $\{0,1\}^n$. A negligible function is a function of the form $n^{-\omega(1)}$. In the context of describing probability distributions, we write $x \sim U_n$ to denote choosing $x$ according to the distribution $U_n$; we also use $x \in S$ to denote choosing an element $x$ from the set $S$ uniformly at random. We use $\cdot$ to denote the standard dot product of binary strings, and $\mathrm{H}(\cdot)$ to denote the Shannon entropy function, namely, $\mathrm{H}(p) = -p\log p - (1-p)\log(1-p)$, for $p \in [0,1]$.

### 2.1 Models of computation

A circuit has AND and OR gates where each gate has in-degree 2 and out-degree 1, and is labeled with a bit that indicates whether its value should be negated. The size of a circuit is the number of gates. A nonuniform polynomial-time algorithm refers to a family of polynomial-size circuits; specifically, we may consider the polynomial-time algorithm as being circuit evaluation and the nonuniformity being the corresponding circuit. An oracle circuit has 3 types of gates: AND, OR and oracle gates. The in/out-degree of the oracle gate matches the input/output length of the oracle. It is easy to see that an oracle circuit of size $s$ having input/output length $n$ and oracle access to a function $f : \{0,1\}^n \to \{0,1\}$ can be encoded using $O(sn\log(sn))$ bits. A nonuniform oracle polynomial-time algorithm refers to a family of polynomial-size oracle circuits.

### 2.2 Average-case hardness and one-way functions

**Definition 1.** *For any $\alpha \in [0,1/2]$, a function $f : \{0,1\}^n \to \{0,1\}$ is* $\alpha$-hard for size $s$ *if every circuit of size $s$ fails to compute $f$ on an $\alpha$ fraction of inputs.*

**Definition 2.** *For any function $\alpha : \mathbb{N} \to [0,1/2]$, a function $f : \{0,1\}^* \to \{0,1\}$ is* $\alpha$-hard *if for every nonuniform polynomial-time algorithm $A$, for all sufficiently large $n$'s,*

$$\Pr_{x\sim U_n}[A(x) \neq f(x)] > \alpha(n)$$

*A function $f$ is* weakly hard-on-average *(resp.* strongly hard-on-average*) if $f$ is $\alpha$-hard for some $\alpha(n) = n^{-c}$ where $c > 0$ is a constant (resp. some $\alpha(n) = 1/2 - n^{-\omega(1)}$). A language $L$ is* $\alpha$-hard *if the characteristic function for $L$ is $\alpha$-hard. We also extend the notions of weakly and strongly hard-on-average to languages.*

**Definition 3.** *For any function $\alpha : \mathbb{N} \to [0,1]$, a function $f : \{0,1\}^* \to \{0,1\}^*$ is* $\alpha$-one-way *(resp.* auxiliary-input $\alpha$-one-way*) if $f$ is computable in polynomial time (resp. by* *a nonuniform polynomial-time algorithm) and if for every nonuniform polynomial-time algorithm A, and all sufficiently large n's,*

$$\Pr_{x \sim U_n}[A(f(x)) \notin f^{-1}(f(x))] > \alpha(n)$$

*A function $f$ is* weakly one-way *(resp.* strongly one-way*) if $f$ is $\alpha$-one-way for some $\alpha(n) = n^{-c}$ where $c > 0$ is a constant (resp. some $\alpha(n) = 1 - n^{-\omega(1)}$).*

All of these notions extend naturally to the setting of oracle nonuniform polynomial-time algorithms (and oracle circuits). We will often appeal to the following technical lemma from [GT00] stating that random permutations are strongly one-way. We will also use the fact that the proof relativizes.

**Lemma 1 ([GT00]).** *For all sufficiently large $\ell$, with probability $1 - 2^{-2^{\ell/2}}$ over $\pi \in \Pi_\ell$, for all oracle circuits A of size $2^{\ell/5}$,*

$$\Pr_{x \sim U_\ell}[A^\pi(\pi(x)) \neq x] > 1 - 2^{-\ell/5}$$

### 2.3 Interactive proofs and argument systems

For a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$, the *language associated with $R$* is $L_R = \{x : \exists y\ (x,y) \in R\}$.

**Definition 4.** *An interactive protocol $(P,V)$ is an* interactive proof system *for a language $L$ if there is a relation $R$ such that $L = L_R$, and functions $c, s : \mathbb{N} \to [0,1]$ such that $1 - c(n) > s(n) + 1/poly(n)$ and the following holds:*

- *(efficiency): the length of all the messages are bounded by a polynomial in the length of the common input x, and V is computable in probabilistic polynomial time.*
- *(completeness): for all $(x,w) \in R$, then $V$ accepts in $(P(w),V)(x)$ with probability at least $1 - c(|x|)$,*
- *(soundness): for all $x \notin L$, then for every $P^*$, $V$ accepts in $(P^*,V)(x)$ with probability at most $s(|x|)$.*

We call $c(\cdot)$ the *completeness error* and $s(\cdot)$ the *soundness error*. We say that $(P,V)$ has *negligible error* if both $c$ and $s$ are negligible. We say that it has *perfect completeness* if $c = 0$. $P$ is an *efficient prover* if $P(w)$ is computable by a probabilistic polynomial-time algorithm when $(x,w) \in R$. The *communication complexity* of the proof system is the total length of all the messages exchanged by both parties, and the *round complexity* is the total number of messages exchanged by both parties (in both directions).

**Definition 5.** *An* argument system $(P,V)$ *is defined in the same way as an interactive proof system, with the following modification:*

– *The soundness condition is replaced with* computational soundness*: For every nonuniform polynomial-time machine $P^*$ and for all sufficiently long $x \notin L$, the verifier $V$ accepts in $(P^*,V)(x)$ with probability at most $s(|x|)$.*

In this paper, we focus on public-coin argument systems with perfect completeness, negligible soundness error, and an efficient prover.

### 2.4 Relativization and black-box reductions

In each of our relativized constructions, we consider a family of oracles $\mathcal{O} = \{\mathcal{O}_n\}_{n\geq 1}$, with an oracle for each input length. For simplicity, we will only present our results for the model where an oracle Turing machine (respectively an oracle circuit) on an input of length $m$ only queries $\mathcal{O}_n$ for a single value of $n$, where $n = n(m)$ is polynomially related to $m$. This is already sufficient to capture most black-box reductions and transformations used in cryptography.

For black-box constructions of cryptographic primitives from interactive protocols, we require that the construction uses oracle access to the efficiently computable entities in the protocol, such as the verifier, the efficient prover (if one exists), and the simulator (in the case of zero-knowledge). An example is the construction of one-way functions from zero-knowledge proof systems in [V04], where the function is computed using black-box access to the simulator and the verifier for the underlying proof system. Such constructions usually only yield auxiliary-input cryptographic primitives because we need to hardwire the instance used in the protocol into the algorithm for computing the primitive. We omit a formal definition of black-box constructions used in this work (as a sufficiently general framework will be fairly involved without yielding any additional insight); instead, we refer the reader to [RTV04] for a formal treatment of black-box constructions and reductions.

## 3 The Impagliazzo-Rudich construction

We begin by reviewing the relativized construction of Pessiland due to Impagliazzo and Rudich (unpublished). We use some of the ideas and proofs in our main constructions.

**Theorem 3 (Impagliazzo-Rudich).** *There exists an oracle relative to which there exists a strongly hard-on-average language in* $\mathsf{NP} \cap \mathsf{coNP}$*, but no one-way functions.*

For any $f \in \mathcal{F}_{n,n}$ (namely, a function from $\{0,1\}^n$ to $\{0,1\}^n$), we define a verification oracle for $f$:

$$V_f(x,y) = \begin{cases} 1 & \text{if } f(x) = y \\ 0 & \text{otherwise} \end{cases}$$

The construction used in the proof of Theorem 3 is as follows:

**Construction 1.** *For each $n \in \mathbb{N}$, we have an oracle $V_\pi$, for some permutation $\pi \in \Pi_n$ (specifically, one that satisfies the condition in Lemma 1 and that in Lemma 2 below). In addition, we provide access to a* $\mathsf{PSPACE}$ *oracle.*

We choose $\pi$ by sampling a random permutation on $\{0,1\}^n$. If $\pi$ is strongly one-way, then the $\mathsf{NP}$-relation $\{(x,w) \mid \pi(w) = x\}$ yields a hard-on-average search problem (with a unique witness), and upon applying the Goldreich-Levin transformation [GL89], we obtain a strongly hard-on-average language in $\mathsf{NP} \cap \mathsf{coNP}$. Furthermore, a polynomial-time oracle Turing machine $M$ makes a query to $V_\pi$ of the form $(x, \pi(x))$ with negligible probability, so $M^Z$ agrees with $M^{V_\pi}$ on almost all inputs. Here, $Z : \{0,1\}^* \to \{0,1\}$ denotes the function that evaluates to 0 everywhere. Using the $\mathsf{PSPACE}$ oracle, we may then invert $M^Z$ everywhere and thus $M^{V_\pi}$ almost everywhere.

**Lemma 2.** *Fix $T(n) = n^{\log n}$ and an encoding of oracle Turing machines. For all sufficiently large $n$, with probability at least $1/2n^2$ over $\pi \in \Pi_n$, for all oracle Turing machines $M$ that can be described using at most $\log n$ bits and makes at most $T(n)$ oracle queries,*

$$\Pr_{x \sim U_n}\left[M^{V_\pi}(x) = M^Z(x)\right] \geq 1 - \frac{1}{2T(n)}$$

*Proof.* Fix an oracle Turing machine $M$. By linearity of expectations, we have

$$\mathrm{E}_{\pi \in \Pi_n}\left[\left|\{x \in \{0,1\}^n : M^{V_\pi}(x) \neq M^Z(x)\}\right|\right] \leq 2^n \cdot \frac{T(n)}{2^n - T(n)}$$

By Markov's inequality,

$$\Pr_{\pi \in \Pi_n}\left[\left|\{x \in \{0,1\}^n : M^{V_\pi}(x) \neq M^Z(x)\}\right| \geq \frac{2^n}{2T(n)}\right] \leq \frac{2T(n)^2}{2^n - T(n)} < \frac{1}{4n^3}$$

This allows us to take a union bound over all oracle Turing machines $M$ with description at most $\log n$ bits (there are at most $2n$ of them). □

*Remark 1.* As stated, the above lemma only allows us to rule out one-way functions computed by oracle Turing machines $M$ that on an input of length $n$, only queries $V_\pi$ corresponding to a permutation on $\{0,1\}^n$. To handle the case where $M$ queries oracles corresponding to permutations on different input lengths, we choose $\pi \in \Pi_n$ to allow for a union bound over all oracle Turing machines $M$ that can be described using at most $\log n$ bits and makes at most $T(n)$ queries to $V_\pi$ on some input of length $m(n)$ where $m(n)$ is polynomially related to $n$ (instead of only considering $m(n) = n$).

**Lemma 3** **([LTW05]).** *Let $f, g : \{0,1\}^n \to \{0,1\}^n$ be functions that agree on an $\varepsilon$ fraction of inputs. Let $A()$ be the probabilistic procedure that, for every $y \in \{0,1\}^n$, $A(y)$ outputs $\perp$ if $f^{(-1)}(y) = \emptyset$, and a uniformly random element of $f^{(-1)}(y)$ otherwise. Then, the probability that $A(g(x)) \in g^{(-1)}(g(x))$ is at least $\varepsilon^2$, when taken over the uniform choice of $x \in \{0,1\}^n$ and over the internal coin tosses of $A$.*

*Remark 2.* Since we also provide access to a $\mathsf{PSPACE}$ oracle, we should say that with overwhelming probability over $\pi$, $M^{\mathsf{Z},\mathsf{PSPACE}}$ agrees with $M^{V_\pi,\mathsf{PSPACE}}$ almost everywhere. This is true since the proof of Lemma 2 relativizes. With a $\mathsf{PSPACE}$ oracle, we may uniformly sample pre-images for $M^{\mathsf{Z},\mathsf{PSPACE}}$ in probabilistic polynomial time, which together with Lemma 3, is sufficient to rule out one-way functions.

**Lemma 4** **([GT00,GL89]).** *For all sufficiently large n, with probability $1 - o(1/n^2)$ over $\pi \in \Pi_n$, the function $f : \{0,1\}^{2n} \to \{0,1\}$ given by $f(y,r) = \pi^{-1}(y) \cdot r$ is $(1/2 - n^{-\log n})$-hard against oracle circuits of size $n^{\log n}$ with oracle access to $\pi$.*

## 4 Our first Pessiland

We present our construction that establishes Theorem 1. Fix $n$ and $\ell = 100 \log^2 n$. For each $f \in \mathcal{F}_{n,3n}$ and a collection of permutations $\{\pi_y \in \Pi_\ell \mid y \in \{0,1\}^{3n}\}$, we define a 3-tuple $(V_\pi, V_f, T)$ where $V_\pi$ and $V_f$ are verification oracles for checking the relations induced by $\{\pi_y\}$ and $f$, and $T$ is a trapdoor permutation oracle for computing $\pi_y$ and $\pi_y^{-1}$ if given $(w, y)$ such that $f(w) = y$.

Our 2-round protocol for the language $L_f = \{y \mid \exists w : f(w) = y\}$ is shown in Fig 1. On input $y \in \{0,1\}^{3n}$, the prover is asked to invert $\pi_y$ on a random input, and the verifier checks the answer using the verification oracle $V_\pi$. The trapdoor permutation oracle yields an efficient prover for the YES instances. For the NO instances, generating an accepting response is as hard as inverting a random permutation.

$$V_\pi(y, \alpha, \beta) = \begin{cases} 1 & \text{if } \pi_y(\alpha) = \beta \\ 0 & \text{otherwise} \end{cases}$$

$$V_f(w, y) = \begin{cases} 1 & \text{if } f(w) = y \\ 0 & \text{otherwise} \end{cases}$$

$$T(w, y, b, z) = \begin{cases} \pi_y(z) & \text{if } f(w) = y \text{ and } b = 0 \\ \pi_y^{-1}(z) & \text{if } f(w) = y \text{ and } b = 1 \\ \perp & \text{otherwise} \end{cases}$$

**Common input**: An instance $y \in \{0,1\}^{3n}$.

**Prover's private input**: A witness $w \in \{0,1\}^n$.

$V \rightarrow P$ : Send $\beta \xleftarrow{\mathrm{R}} \{0,1\}^{O(\log^2 n)}$.

$P \rightarrow V$ : Send $\alpha = T(w,y,\beta)$.

**Verification**: $V$ accepts if $V_\pi(y,\alpha,\beta) = 1$ (that is, $\pi_y(\alpha) = \beta$).

**Fig. 1.** 2-round public-coin protocol $\mathsf{prot}$ for the language $L_f = \{y \mid \exists w : f(w) = y\}$

**Construction 2.** *For each $n \in \mathbb{N}$, we have an oracle $(V_\pi, V_f, T)$, for some appropriate choices of $f \in \mathcal{F}_{n,3n}$ and $\{\pi_y \in \Pi_{O(\log^2 n)} \mid y \in \{0,1\}^{3n}\}$. In addition, we provide access to a $\mathsf{PSPACE}$ oracle.*

We begin with an overview of the analysis for our construction.

*Computational soundness.* A successful cheating prover is one that inverts $\pi_y$ on a noticeable fraction of inputs, for some $y \notin L_f$. However, for each $y \notin L_f$, the random permutation $\pi_y$ is one-way against oracle circuits of size $n^{\log n}$ with probability $1 - 2^{-n^{\log n}}$ (Lemma 1). This holds even if the circuit is given oracle access to $V_f, \pi_y$ and $(\pi_{y'}, \pi_{y'}^{-1})$ for all $y' \neq y$ (which are sufficient to simulate the oracles $(V_\pi, V_f, T)$), because $\pi_{y'}$ and $f$ are chosen independently of $\pi_y$. We can then take a union bound to ensure that every permutation in the collection $\{\pi_y\}$ is strongly one-way, as shown in Lemma 5.

*Ruling out low-communication proof systems.* A 2-round argument system for $L_f$ with communication complexity $\ell(n)$ is only interesting if we could rule out 2-round interactive proof systems for the language $L_f$ with the same communication complexity. We prove in Lemma 6 that there is no subexponential-size oracle circuits for deciding $L_f$, given oracle access to $V_f$ and to $\{(\pi_y, \pi_y^{-1})\}_{y \in \{0,1\}^{3n}}$, which is sufficient to simulate oracle access to $(V_\pi, V_f, T)$. This implies $L_f \notin \mathsf{BPTime}(2^{o(n)})$. Note that an algorithm running in time $\mathsf{BPTime}(2^{O(\ell(n))})$ can compute and invert the permutations $\pi_y$ everywhere given oracle access to $V_\pi$. It is therefore essential to our proof that the collection of permutations $\{\pi_y\}$ is defined independently of $f$.

*Ruling out one-way functions.* The analysis is virtually identical to that for the Impagliazzo-Rudich Pessiland, since a polynomial-time oracle Turing machine is unlikely to query $(V_\pi, V_f, T)$ at any input where the answer is neither 0 nor $\perp$. Note that in order to satisfy the efficient prover condition (for YES instances), it suffices to provide oracle access to $\pi_{f(w)}^{-1}$ in $T$. By incorporating oracle access to $\pi_{f(w)}$ into $T$, we also rule out the trivial auxiliary-input one-way permutation given by $\pi_{f(w)}^{-1}$.

However, we do not know how to rule out every auxiliary-input one-way function for this construction.

*A strongly hard-on-average language.* We can construct the language from the strongly hard-on-average function given by $g : \{0,1\}^{3n+2\ell} \to \{0,1\}$ where $g(y,\beta,r) = \pi_y^{-1}(\beta) \cdot r$.

**Lemma 5.** *For all sufficiently large $n$, for every $f \in \mathfrak{F}_{n,3n}$, with probability $1 - 2^{-\Omega(n^{\log n})}$ over $\{\pi_y\}_{y\in\{0,1\}^{3n}} \in \Pi_\ell^{2^{3n}}$, for all $y \in \{0,1\}^{3n}$ and for all oracle circuits $A$ of size $n^{\log n}$,*

$$\Pr_{x\sim U_\ell}[A^{V_f,\pi_y,\{(\pi_{y'},\pi_{y'}^{-1})|y'\neq y\}}(\pi_y(x)) = x] < 2^{-n^{\log n}}$$

*Proof.* By Lemma 1 (and the fact that it relativizes), if we fix a sufficiently large $n$, along with any $f \in \mathfrak{F}_{n,3n}$, any $y \in \{0,1\}^{3n}$, and any $\pi_{y'} \in \Pi_\ell$ for all $y' \neq y$, we know that with probability $2^{-\Omega(n^{\log n})}$ over $\pi_y \in \Pi_\ell$, for all oracle circuits $A$ of size $n^{\log n}$,

$$\Pr_{x\sim U_\ell}[A^{V_f,\pi_y,\{(\pi_{y'},\pi_{y'}^{-1})|y'\neq y\}}(\pi_y(x)) = x] < 2^{-n^{\log n}}$$

The lemma follows from taking a union bound over all $y \in \{0,1\}^{3n}$. □

**Lemma 6.** *For all sufficiently large $n$, for every collection of permutations $\{\pi_y\}_{y\in\{0,1\}^{3n}}$, with probability $1 - 2^{-\Omega(2^n)}$ over $f \in \mathfrak{F}_{n,3n}$, there is no oracle circuit of size $2^{n/5}$ that given oracle access to $V_f$ and to $\{(\pi_y, \pi_y^{-1})\}_{y\in\{0,1\}^{3n}}$ decides $L_f$.*

*Proof.* We establish this result following the counting argument in [GT00]. We may neglect oracle access to $\{(\pi_y, \pi_y^{-1})\}_{y\in\{0,1\}^{3n}}$ since the argument relativizes. The idea is to show that any function $f$ for which there is an oracle circuit $A$ that given oracle access to $V_f$ decides $L_f$ has a "short" description (given $A$). There are very few such functions, so a random $f$ satisfies the hardness property with overwhelming probability.

Formally, fix an oracle circuit $A : \{0,1\}^{3n} \to \{0,1\}$ of size $2^{n/5}$ and suppose $A$ on oracle access to $V_f$ decides $L_f$ for some $f \in \mathfrak{F}_{n,3n}$. We simulate $A$ on every input in $\{0,1\}^{3n}$ in lexicographic order and observe the queries that $A$ makes to $V_f$. WLOG, assume $A$ never makes the same query twice on a given input. Define $X \subseteq \{0,1\}^n$ to be all $x$ such that $A$ queries $V_f$ on $(x, f(x))$.

CASE 1: $|X| \leq \frac{3}{4} \cdot 2^n$. Given the set $X$ and $f|_X$, we may simulate $A$ on all inputs without oracle access to $V_f$, thereby recovering the set $f(\{0,1\}^n)$. We may then specify $f$ on each input outside $X$ using just $n$ bits (instead of $3n$ bits) since we only need $n$ bits to specify an element in the set $f(\{0,1\}^n)$.

CASE 2: $|X| > \frac{3}{4} \cdot 2^n$. Over all possible inputs, $A$ makes at most $2^{3n} \cdot 2^{n/5}$ queries to $V_f$. Therefore, there are at most $\frac{1}{4} \cdot 2^n$ values of $x$ for which $A$ makes more than

$4 \cdot 2^{2n} \cdot 2^{n/5}$ queries to $V_f$ of the form $(x, \cdot)$. In particular, there is a subset $X'$ of $X$ with $\frac{1}{2} \cdot 2^n$ elements, and for each $x \in X'$, $A$ makes at most $4 \cdot 2^{2n} \cdot 2^{n/5}$ queries to $V_f$ of the form $(x, \cdot)$. Given the circuit $A$, the set $X'$ and $f|_{\{0,1\}^n \setminus X'}$, we may specify $f$ on each input in $X'$ using $11n/5 + 2$ bits (instead of $3n$ bits) since we only need to specify $i$ such that the $i$'th query $A$ makes of the form $(x, \cdot)$ returns 1.

In both cases, given $A$, we may specify $f$ with $2^n(2n/5 - 2)$ less bits (relative to the $2^n \cdot 3n$ bits required to specify a function in $\mathcal{F}_{n,3n}$). It takes an additional $O(2^{n/5}n^2)$ bits to specify $A$. $\square$

## 5 A second Pessiland

We present our next construction that establishes Theorem 2. It is similar to the Impagliazzo-Rudich Pessiland except we provide a verification oracle for a random function instead of a random permutation.

**Construction 3.** *For each $n \in \mathbb{N}$, we have an oracle $V_f$, for some appropriate choice of $f \in \mathcal{F}_{n,n}$. In addition, we provide access to a* PSPACE *oracle.*

First, we show that for most $f \in \mathcal{F}_{n,n}$, the language $L_f = \{y \mid \exists x : f(x) = y\}$ is weakly hard-on-average (Lemma 7); the proof is an extension of that for Lemma 6, except more involved because we are establishing average-case hardness instead of worst-case hardness. Since the main technical result from [HVV04] on hardness amplification within NP relativizes, we may deduce that there is a strongly hard-on-average language $L'_f$ in NP/poly, obtained by applying some monotone transformation to some padded variant of $L_f$. We provide an additional oracle that on input $1^n$, outputs the nonuniformity needed to compute $L'_f$ in NP. To rule out auxiliary-input one-way functions, it suffices to show that the function computed by any small oracle circuit may be approximated by the function computed by a standard circuit with a polynomial blow-up in size (Lemma 8).

**Lemma 7.** *For all sufficiently large $n$, with probability $1 - 2^{-\Omega(n^2)}$ over $f \in \mathcal{F}_{n,n}$, the language $L_f = \{y \mid \exists x : f(x) = y\}$ is 0.01-hard against oracle circuits of size $2^{o(n)}$ with oracle access to $V_f$.*

*Proof (sketch).* A standard "balls in bins" analysis (e.g. [MR95, Theorem 4.18]) tells us that with probability $1 - 2^{-\Omega(2^n)}$ over $f \in \mathcal{F}_{n,n}$, $|f(\{0,1\}^n)|$ is bounded from above by $\frac{2}{3} \cdot 2^n$ (we may replace $\frac{2}{3}$ by any constant larger than $1 - \frac{1}{e}$). We may then simply focus on $f$ such that $|f(\{0,1\}^n)| < \frac{2}{3} \cdot 2^n$, and proceed as in the proof of Lemma 6. Again, we consider an oracle circuit $A : \{0,1\}^n \to \{0,1\}$ that solves $L_f$ on at least a 0.99 fraction of inputs and we define $X$ to be all $x$ such that $A$ queries $V_f$ on $(x, f(x))$.

CASE 1: $|X| \leq 0.02 \cdot 2^n$. Let $Y = \{y \mid A(y) \neq L_f(y)\}$, that is, the subset of inputs on which $A$ is wrong. Given $f|_X$ and the sets $X, Y$ (which may be specified using $(0.02n + \mathrm{H}(0.02) + H(0.01) + o(1))2^n$ bits), we may simulate $A$ on all inputs without oracle access to $V_f$, thereby recovering the set $f(\{0,1\}^n)$. We may then specific $f$ on inputs outside $X$ using $\log(\frac{2}{3} \cdot 2^n)$ bits. Therefore, given the circuit $A$, we may specify $f$ using $2^n n - (0.98 \log \frac{3}{2} - \mathrm{H}(0.01) - \mathrm{H}(0.02) - o(1))2^n < 2^n(n - 0.35)$ bits.

CASE 2: $|X| > 0.02 \cdot 2^n$. We argue that there is a subset $X'$ of $X$ with $0.01 \cdot 2^n$ elements, and for each $x \in X'$, $A$ makes at most $100 \cdot 2^{o(n)}$ queries to $V_f$ of the form $(x, \cdot)$. Given the circuit $A$, we may then specify $f$ using $(0.99 + o(1))2^n n$ bits. $\square$

To facilitate the proof of the next lemma, we introduction an additional notation: for any $f \in \mathcal{F}_{n,n}$ and any subset $Q$ of $\{0,1\}^n$, we define:

$$V_{f,Q}(x,y) = \begin{cases} 1 & \text{if } f(x) = y \text{ and } x \in Q \\ 0 & \text{otherwise} \end{cases}$$

**Lemma 8.** *For all sufficiently large n, with probability $1 - 2^{-\Omega(n^2)}$ over $f \in \mathcal{F}_{n,n}$, for all oracle circuits C of size s where $n \leq s \leq 2^{n/10}$ and for all $\varepsilon \geq 2^{-n/10}$, there exists a circuit $C'$ of size $O(s^4 n^3/\varepsilon^2)$ such that $C^{V_f}$ and $C'$ agree on a $1 - \varepsilon/2$ fraction of inputs.*

To see why the naive approach of setting $C' = C^Z$ (as in Lemma 2) fails, consider an oracle circuit $C$ that independent of its input, outputs $V_f(0^n, 1^n)$. Then, with probability $1 - 2^{-n}$, $C'$ and $C$ agree on all inputs, and with probability $2^{-n}$, disagree on all inputs. This is not sufficient for a union bound over all polynomial-size circuits. To work around this, we hardwire into $C'$ information about $f$. Specifically, we show that with overwhelming probability over $f \in \mathcal{F}_{n,n}$, for all $C$ of size $s$, there exists a set $Q \subseteq \{0,1\}^n$ of size $O(s^4 n^2/\varepsilon^2)$ such that the circuit $C^{V_{f,Q}}$ agrees with $C^{V_f}$ on a $1 - \varepsilon/2$ fraction of inputs. Note that we allow $Q$ to depend on $f$. We may specify $f|_Q$ using $|Q|n$ bits of nonuniformity, so $C^{V_{f,Q}}$ may be computed by a circuit $C'$ of size $O(s^4 n^3/\varepsilon^2)$ (without oracle access to $V_f$).

Here is an outline of the analysis. Let us examine the first oracle query made by the circuit $C$ on different inputs, and we define $Q_1$ to be all $x$ such that the first query $C$ makes to $V_f$ matches $(x, \cdot)$ on more than a $\varepsilon^3/s^3 n^2$ fraction of inputs. Therefore, $|Q_1| = \mathrm{poly}(s, n, 1/\varepsilon)$. Now, consider the oracle circuit $C_1$ that behaves like $C$, except the first oracle query is made to $V_{f,Q_1}$ instead of $V_f$. Suppose $C$ and $C_1$ differs on a $\varepsilon/2s$ fraction of inputs. This must be because for a $\varepsilon/2s$ fraction of inputs, the first query $C$ makes to $V_f$ matches $(x, f(x))$, for some $x \notin Q_1$. For a random $f$ and a fixed $x$, this happens with probability $2^{-n}$. Moreover, this must happen for at least $s^2 n^2/\varepsilon^2$ different values of $x$ not in $Q_1$ (since each $x \notin Q_1$ accounts for at most a $\varepsilon^3/s^3 n^2$ fraction of inputs). For a random $f$, the evaluation of $f$ on each of these $x$ values are independent.

Thus, the probability (over $f$) that $C$ and $C_1$ differs on a $\varepsilon/2s$ fraction of inputs is roughly $2^{-\Omega(ns^2)}$.

*Proof.* Formally, fix $f \in \mathcal{F}_{n,n}$. We define oracle circuits $C_0, C_1, \ldots, C_s$ and subsets $Q_0, Q_1, \ldots, Q_s$ of $\{0,1\}^n$ inductively as follows:

- $Q_0 = \emptyset$ and $C_0 = C$.
- $Q_i$ is union of $Q_{i-1}$ and the set

  $$\left\{x \in \{0,1\}^n \;\middle|\; \Pr_z\left[i\text{'th oracle query for computing } C_{i-1}^{V_f}(z) \text{ matches } (x,\cdot)\right] \geq \varepsilon^2/s^3n^2\right\}$$

- $C_i$ on input $z$ and oracle access to $V_f$ simulates the computation of $C^{V_f}(z)$ except for $j = 1,2,\ldots,i$, the $j$'th oracle query is answered using $V_{f,Q_j}$ instead of $V_f$. We will hardwire the description of the sets $Q_1,\ldots,Q_i$ into $C_i$, so upon oracle access to $V_f$, $C_i$ may simulate the oracles $V_{f,Q_j}, j = 1,\ldots,i$.

*Claim.* For all $i = 1,2,\ldots,s$, $\quad \Pr_{f\in\mathcal{F}_{n,n}}\left[\Pr_z\left[C_{i-1}^{V_f}(z) \neq C_i^{V_f}(z)\right] < \varepsilon/2s\right] \geq 1 - 2^{-\Omega(sn^2)}$

It follows readily from the claim that

$$\Pr_{f\in\mathcal{F}_{n,n}}\left[\Pr_z\left[C^{V_f}(z) \neq C_s^{V_f}(z)\right] < \varepsilon/2\right] \geq 1 - s\cdot 2^{-\Omega(sn^2)}$$

This implies that with overwhelming probability over $f$, $C^{V_f}$ and $C^{V_{f,Q_s}}$ agree on a $1-\varepsilon/2$ fraction of inputs. We may bound $|Q_s|$ by $s^4n^2/\varepsilon^2$ since $|Q_i| \leq |Q_{i-1}| + s^3n^2/\varepsilon^2$. Hence, $C^{V_{f,Q_s}}$ may be computed by a circuit $C'$ of size $O(s^4n^2/\varepsilon^2)$. The lemma then follows from taking a union bound over all circuits of size $s$, all $s$ between $n$ and $2^{n/10}$, and all $1/\varepsilon$ between 2 and $2^{n/10}$. □

Now, we provide the proof of the above claim.

*Proof (of claim).* We start with the case $i = 1$. Note that the definition of $Q_1$ does not depend on $f$. Consider any input $z$ to $C^{V_f}$. If the first oracle query made by $C^{V_f}$ corresponds to an element in $Q_1$, then $\Pr_f[C_1^{V_f}(z) = C^{V_f}(z)] = 1$. Otherwise, $\Pr_f[C_1^{V_f}(z) = C^{V_f}(z)] = 1 - 2^{-n}$. For each $x \in \{0,1\}^n$, we define

$$\alpha_x = \begin{cases} \Pr_z\left[\text{first oracle query for } C^{V_f}(z) \text{ matches } (x,\cdot)\right] & \text{if } x \notin Q_1 \\ 0 & \text{otherwise} \end{cases}$$

(note that $\alpha_x$ is independent of $f$) and $Y_x$ to be the random variable (where the randomness is over $f \in \mathcal{F}_{n,n}$) for the probability

$$\Pr_z\left[\text{first oracle query for } C^{V_f}(z) \text{ matches } (x,\cdot) \text{ and } C^{V_f}(z) \neq C_1^{V_f}(z)\;\right]$$

Hence, we have $\sum_x \alpha_x \leq 1$ and for all $x \in \{0,1\}^n$:

$$0 \leq Y_x \leq \alpha_x \leq \varepsilon^2/s^3n^2 \text{ and } \mathrm{E}_f[Y_x] = \alpha_x 2^{-n}$$

In addition,

$$\Pr_{f,z}\left[C_1^{V_f}(z) \neq C^{V_f}(z)\right] = \mathrm{E}_f\left[\sum_x Y_x\right]$$

By convexity, we have $\sum_x \alpha_x^2 \leq \varepsilon^2/s^3n^2$. Applying the Hoeffding bound [H63] yields:

$$\Pr_f\left[\sum_x Y_x - 2^{-n} \geq \varepsilon/4s\right] \leq e^{-2(\varepsilon/4s)^2/\sum_x \alpha_x^2} \leq e^{-sn^2/8}$$

In the general case, we fix an assignment to $f|_{Q_{i-1}}$, so the set $Q_i$ is also fixed. As before, we define

$$\alpha_x = \begin{cases} \Pr_z\left[i\text{'th oracle query for } C_{i-1}^{V_f}(z) \text{ matches } (x,\cdot)\right] & \text{if } x \notin Q_i \\ 0 & \text{otherwise} \end{cases}$$

(here, $\alpha_x$ is independent of $f|_{\{0,1\}^n \setminus Q_{i-1}}$) and $Y_x$ to be the random variable (where the randomness is over $f|_{\{0,1\}^n \setminus Q_{i-1}}$) for the probability

$$\Pr_z\left[i\text{'th oracle query for } C_{i-1}^{V_f}(z) \text{ matches } (x,\cdot) \text{ and } C_{i-1}^{V_f}(z) \neq C_i^{V_f}(z)\ \right]$$

Again, the Hoeffding bound yields:

$$\Pr_{f|_{\{0,1\}^n \setminus Q_{i-1}}}\left[\sum_x Y_x - 2^{-n} \geq \varepsilon/4s\right] \leq e^{-sn^2/8}$$

This holds for all $f|_{Q_{i-1}}$. Averaging over all possible assignments of $f|_{Q_{i-1}}$, we have:

$$\Pr_f\left[\Pr_z\left[C_{i-1}^{V_f}(z) \neq C_i^{V_f}(z)\right] \geq \varepsilon/4s + 2^{-n}\right] \leq e^{-sn^2/8}$$

This completes the proof of the technical claim. $\square$

## Acknowledgements

I am grateful towards Salil Vadhan for sharing his insightful observations which led me towards the problems addressed in this work, and Luca Trevisan for his help with the proofs in Section 5. In addition, I thank Lance Fortnow and Russell Impagliazzo for pointing out previous constructions of Pessiland, and the anonymous referees for their helpful and constructive feedback.

## References


[B01] B. Barak. How to go beyond the black-box simulation barrier. In *Proc. 42nd FOCS*, 2001.

[BG02] B. Barak and O. Goldreich. Universal arguments and their applications. In *Proc. 17th CCC*, 2002.

[GH98] O. Goldreich and J. Håstad. On the complexity of interactive proofs with bounded communication. *IPL*, 67(4):205–214, 1998.

[GL89] O. Goldreich and L. Levin. Hard-core predicates for any one-way function. In *Proc. 21st STOC*, 1989.

[GT00] R. Gennaro and L. Trevisan. Lower bounds on efficiency of generic cryptographic constructions. In *Proc. 41st FOCS*, 2000.

[H63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[HVV04] A. Healy, S. Vadhan, and E. Viola. Using nondeterminism to amplify hardness. In *Proc. 36th STOC*, 2004.

[I95] R. Impagliazzo. A personal view of average-case complexity. In *Proc. 10th Structure in Complexity Theory Conference*, 1995.

[IL89] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proc. 30th FOCS*, 1989.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. 21st STOC*, 1989.

[K92] J. Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proc. 24th STOC*, 1992.

[LTW05] H. Lin, L. Trevisan, and H. Wee. On hardness amplification of one-way functions. In *Proc. 2nd TCC*, 2005.

[M00] S. Micali. Computationally sound proofs. *SICOMP*, 30(4):1253–1298, 2000.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[RTV04] O. Reingold, L. Trevisan, and S. Vadhan. Notions of reducibility between cryptographic primitives. In *Proc. 1st TCC*, 2004.

[V04] S. Vadhan. An unconditional study of computational zero knowledge. In *Proc. 45th FOCS*, 2004.

[W05] H. Wee. On round-efficient argument systems. In *Proc. 32nd ICALP (Track C)*, 2005.

# A Appendix

## A.1 The Hoeffding bound

We state the concentration result for sum of independent bounded random variables (with possibly arbitrary distributions) used in the proof of Lemma 8.

**Lemma 9 ([H63]).** *If $X_1, \ldots, X_n$ are independent random variables such that $a_i \leq X_i \leq b_i$, $i = 1, 2, \ldots, n$, then for all $t > 0$,*

$$\Pr[X - \mathrm{E}[X] \geq t] \leq e^{-2t^2 / \Sigma_i (b_i - a_i)^2}$$

*where $X = X_1 + \ldots X_n$.*

## A.2 Necessity of hardness assumptions

For ease of reference, we reproduce the proof from [W05] (with a minor improvement in the result) that a 2-round argument system for $\mathsf{NP}$ with subpolynomial communication complexity implies hard-on-average search problems in $\mathsf{NP}$. Under complexity assumptions, such a protocol cannot be a proof system [GH98]. Hence, there exists infinitely many NO instances that are merely "computationally sound", from which we may construct hard-on-average search problems in $\mathsf{NP}$. We stress that the construction of hard-on-average search problems uses the underlying verifier in a black-box manner.

**Lemma 10 ([W05]).** *Suppose a promise problem $\Pi = (\Pi_Y, \Pi_N)$ has a 2-round public-coin argument system $(P, V)$ with communication complexity $m(n)$, perfect completeness and negligible soundness error. Then, there exists a subset $I \subset \Pi_N$ such that:*

- *Ignoring inputs in $I$, $\Pi$ has a 2-round public-coin proof system with communication complexity $m(n)$, perfect completeness and soundness error less than 1. This implies $(\Pi_Y, \Pi_N \setminus I) \in \mathsf{DTime}(2^{O(m(n))})$.*
- *When $x \in I$, the predicate $V(x, \cdot, \cdot)$ induces a distribution over hard-on-average search instances in $\mathsf{NP}$. That is, for every $x \in I$:*

$$\Pr_r[\exists\, y : V(x, r, y) = 1] = 1,$$

*but for every $n$, every $x \in I \cap \{0,1\}^n$ and every nonuniform polynomial-time algorithm $A$, there exists a negligible function $\varepsilon(n)$ such that*

$$\Pr_r[V(x, r, A(r)) = 1] < \varepsilon(n)$$

**Theorem 4** **([w05]).** *Suppose* NP *has a 2-round public-coin argument system* $(P,V)$ *with communication complexity* $n^{o(1)}$*, perfect completeness and negligible soundness error. Then, (at least) one of the following is true:*

- $\mathsf{NP} \subseteq \mathsf{DTime}(2^{n^{o(1)}})$
- *There exists an infinite set* $I$ *such that for all* $x \in I$*, the predicate* $V(x,\cdot,\cdot)$ *induces a distribution over hard-on-average search instances in* NP *(as formalized in Lemma 10). This yields an auxiliary-input samplable distribution over satisfiable instances in* NP *where the search problem is infinitely-often strongly hard-on-average.*