On Basing One-Way Functions on NP-Hardness

Adi Akavia[*] Oded Goldreich[†] Shafi Goldwasser[‡] Dana Moshkovitz[§]

November 22, 2005

Abstract

We consider the question of whether it is possible to base the existence of one-way functions on $\mathcal{NP}$-hardness. That is we study the the possibility of reductions from a worst-case $\mathcal{NP}$-hard decision problem to the task of inverting a polynomial time computable function. We prove two negative results:

1. For any polynomial time computable function f: the existence of a randomized *non-adaptive* reduction of worst case $\mathcal{NP}$ problems to the task of average-case inverting f implies that co$\mathcal{NP} \subseteq \mathcal{AM}$. It is widely believed that co$\mathcal{NP}$ is not contained in $\mathcal{AM}$. Thus, this result may be regarded as showing that such reductions cannot exist (unless co$\mathcal{NP} \subseteq \mathcal{AM}$).

   This result improves previous negative results that placed co$\mathcal{NP}$ in *non-uniform* $\mathcal{AM}$.

2. For any polynomial time computable function f for which it is possible to efficiently compute pre-image sizes (*i.e.*, $|f^{-1}(y)|$ for a given y): the existence of a randomized reduction of worst case $\mathcal{NP}$ problems to the task of inverting f implies that co$\mathcal{NP} \subseteq \mathcal{AM}$. Moreover, this is also true for functions for which it is possible to verify (via and AM protocol) the approximate size of pre-image sizes (*i.e.*, $|f^{-1}(y)|$ for a given y). These results holds for *any reduction*, including *adaptive* ones.

   The previously known negative results regarding worst-case to average-case reductions were confined to *non-adaptive* reductions.

In the course of proving the above results, two new $\mathcal{AM}$ protocols emerge for proving *upper bounds* on the sizes of $\mathcal{NP}$ sets. Whereas the known *lower* bound protocol on set sizes by [Goldwasser-Sipser] works for any $\mathcal{NP}$ set, the known *upper* bound protocol on set sizes by [Aiello-Hastad] works in a setting where the verifier knows a random secret element (unknown to the prover) in the $\mathcal{NP}$ set. The new protocols we develop here, each work under different requirements than that of [Aiello-Hastad], enlarging the settings in which it is possible to prove upper bounds on $\mathcal{NP}$ set size.

Area: Cryptography and Complexity.

Keywords: One-Way Functions, Worst-Case to Average-Case reductions, Adaptive versus Non-adaptive reductions, Interactive Proof Systems.

[*] akavia@mit.edu

[†] oded.goldreich@weizmann.ac.il

[‡] shafi@theory.csail.mit.edu

[§] dana.moshkovitz@weizmann.ac.il

1 Introduction

One-way functions are functions that are easy to compute but hard to invert, where the hardness condition refers to the average-case complexity of the inverting task. The existence of one-way functions is the cornerstone of modern cryptography: almost all cryptographic primitives imply the existence of one-way functions, and many of them can be constructed based either on the existence of one-way functions or on related (but seemingly stronger) versions of this assumption.

The hardness condition of one-way functions is an average-case complexity condition. Clearly, this average-case hardness condition implies a worst-case hardness condition; that is, the existence of one-way functions implies that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$. A puzzling question of fundamental nature is whether or not the necessary worst-case condition is a sufficient one; that is, can one base the existence of one-way functions on the assumption that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$.

More than two decades ago, Brassard [Br] observed that the inverting task associated with a one-way *permutation* cannot be $\mathcal{NP}$-hard under *deterministic* reductions, unless $\mathcal{NP} = \text{co}\mathcal{NP}$. Namely, it is impossible to have a deterministic reduction from a worst-case $\mathcal{NP}$-hard decision problem to the task of inverting a polynomial time computable permutation unless $\mathcal{NP} = \text{co}\mathcal{NP}$.

Feigenbaum and Fortnow [FeFo] followed by Bogdanov and Trevisan [BoTr], shifted attention to the question whether it is possible to have worst-case to average-case randomized reductions among $\mathcal{NP}$-hard *decision problems*. It was shown [BoTr] that any *non-adaptive* reduction of $\mathcal{NP}$-hard worst case problem to the average-case complexity of $\mathcal{NP}$ (with respect to any sampleable distribution) implies that $\text{co}\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$. The non-uniform advice given to the verifier on input of length n is the number of n-bit strings in the $\mathcal{NP}$ target language (i.e the language that the reduction maps to). Recall that $\mathcal{AM}$ is the class of sets having two-round interactive proof systems, and it is widely believed that $\text{co}\mathcal{NP}$ is not contained in $\mathcal{AM}$ (equiv., $\mathcal{NP}$ is not contained in $\text{co}\mathcal{AM}$). Thus, we regard the above result as a negative result showing such reductions cannot exist (unless $\text{co}\mathcal{NP} \subseteq \mathcal{AM}$).

Using known reductions between search and decision problems [BCGL, ImLe], [BoTr] also derive conclusions on the possibility of basing the existence of one-way *functions* on $\mathcal{NP}$-hardness as follows: If there exists an $\mathcal{NP}$-complete set for which deciding any worst case instance is *non-adaptively* reducible to *inverting* any polynomial time computable function (or more generally, solving a search problem with respect to a sampleable distribution) then $\text{co}\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$. However, the *techniques* of [BoTr] refer explicitly only to decision problems, and do not exploit the extra structure of the task of inverting polynomial-time computable functions (nor even the underlying extra structure of working with a sampleable search problems).

In this paper, we return to fully focus on the question of whether it possible to base the existence of one-way functions on the presumed worst case hardness assumption of $\mathcal{NP}$-hard problems. Indeed, we believe that the study of the possibility of basing one-way functions on worst-case $\mathcal{NP}$-hardness is the most important motivation for the study of worst-case to average-case reductions for $\mathcal{NP}$. By explicitly capitalizing on the additional "computational structure" of the search problem associated with the inverting task of polynomial time computable problem, we are able to improve previous results as follows.

1.1 New Results on OWF Complexity

- For any polynomial-time computable function f, the existence of a randomized *non-adaptive* reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\text{co}\mathcal{NP} \subseteq \mathcal{AM}$. This result improves over the previous negative results of [BoTr] that placed $\text{co}\mathcal{NP}$ in *non-uniform* $\mathcal{AM}$ (instead of in *uniform* $\mathcal{AM}$).
- For any polynomial-time computable function f, such that $|f^{-1}(y)|$ is efficiently computable given y, the existence of a (randomized) reduction of $\mathcal{NP}$ to the task of inverting f implies that $\text{co}\mathcal{NP} \subseteq \mathcal{AM}$. More generally, this extends to functions for which given y the pre-image size $|f^{-1}(y)|$ is

efficiently verifiable via an $\mathcal{AM}$ protocol. These include, for example, regular functions (for which all pre-image sizes are the same) with efficiently recognizable range.

We stress that the result holds for any reduction, including *adaptive* ones. Previously known negative results regarding worst-case to average-case reductions were essentially confined to *non-adaptive* reductions.[1]

We point out that, our proof in the case of general reductions and size-verifiable functions holds not only for reductions from worst case NP problems to average case inverting of f, but also from worst case NP problems to inverting f in the worst case. Since it is easy to construct polynomial time computable functions f (not necessarily size verifiable ones) for which solving a worst case NP problem is reducible to inverting f, our results yield a separation between inverting size-verifiable polynomial time computable functions and inverting general polynmial time computable function (assuming as usual co$\mathcal{NP} \not\subseteq \mathcal{AM}$).

For a discussion of possible interpretations of these negative results see appendix A.

1.2 Upper Bound Protocols in New Settings

To prove our negative results $\mathcal{AM}$ protocols for verifying the size of various sets are necessary. For example, in the context of non-adaptive reductions for general functions (*i.e.*, functions that are not necessarily size-verifiable), we need to design an $\mathcal{AM}$ protocol for verifying $|f^{-1}(y)|$ when y is sampled by the randomized reduction. This requires both proving lower and upper bounds on the size of $\mathcal{NP}$ sets. General $\mathcal{AM}$ protocols for lower bounds on the size of $\mathcal{NP}$ and $\mathcal{AM}$ sets were shown by Goldwasser-Sipser[GoSi] in the context of proving that $IP = \mathcal{AM}$. However, the only known $\mathcal{AM}$ protocol to prove upper bounds on set sizes due to Aiello and Hastad [AiHa], works only when the verifier knows a random element $x \in f^{-1}(y)$ where x is unknown to the prover. This condition does not usually hold in our context.

We thus develop two new $\mathcal{AM}$ protocols for upper bounding the sizes of $\mathcal{NP}$ sets which work in settings which arise within our work. These protocols can be utilized elsewhere and may be of independent interest.

The intuition underlying the first protocol is inspired by the main idea of Feigenbaum and Fortnow [FeFo]. The common thread is identifying a situation in which the prover may cheat (without being detected) only in one direction, and deducing that frequent cheating by the prover leads to a (significant) deviation from some expected statistic. The second protocol is inspired by the idea of [BoTr] of "hiding" (from the prover) queries of interest among queries drawn from a close distribution for which you know some statistics (given by a non-uniform advice).

The first protocol is titled *confidence by comparison protocol* (CBC) and is applicable when many $\mathcal{NP}$ sets are drawn out of a distribution D for which some statistics on the sizes of the sets is known. The statistics we use are (an approximation for) the expectation $\mathbb{E}_{S\sim D}[|S|]$, when the sets S drawn from D are of polynomial size, and (an approximation for) expectations of the form $\mathbb{E}_{S\sim D}[\lfloor(\log_{1+\rho}(|S|))\rfloor]$ when the sets S are potentially of super polynomial size. See appendix B for details.

The second protocoll titled *the Hiding protocol* for proving upper bound on on the size of NP sets can be used whenever the $\mathcal{NP}$ set is drawn from a distribution D and the verifier can also sample sets from another distribution $\widetilde{D}$ that has the following two properties: (a) There exists an $\mathcal{AM}$ protocol for proving upper bound on sets drawn from $\widetilde{D}$ (this protocol could be, for example, the Aiello-Hastad protocol, or the above CBC protocol), and (b) $\widetilde{D}$ is statistically close to D in the sense that $\forall S \subseteq \{0,1\}^n$, $\Pr_{S\sim D}[S] \leq \lambda \cdot \Pr_{S\sim\widetilde{D}}[S]$ for $1 \leq \lambda \leq poly(n)$. See appendix B for details.

We remark that both protocols can be used to upper bound $\mathcal{AM}$ sets as well an $\mathcal{NP}$ sets. $\mathcal{AM}$ sets are sets for which membership can be verified via an $\mathcal{AM}$ protocol.

[1] [FeFo] also handles restricted levels of adaptivity, which are not extended in [BoTr].

1.3 Relation to Feigenbaum-Fortnow and Bogdanov-Trevisan

We briefly overview the work of Feigenbaum and Fortnow [FeFo] and the work of Bogdanov and Trevisan [BoTr] and highlight the differences in our approach.

In [FeFo] the question of whether or not $\mathcal{NP}$-complete problems can be *random self-reducible* is addressed. That is, can (worst case) instances of $\mathcal{NP}$-complete problems be reduced to one or more *random instances, where the latter instances are drawn according to a predetermined distribution.* The main result of [FeFo] is that if such (*non-adaptive*) reductions exist, then co$\mathcal{NP}$ is in a non-uniform version of $\mathcal{AM}$, denoted $\mathcal{AM}_{\text{poly}}$. Non-uniformity was used in their work to encode statistics about the target distribution of the reduction.

The work of [BoTr] points out that [FeFo] can be viewed as a result regarding the impossibility of worst-case to average-case reductions (of a restricted type) for $\mathcal{NP}$-complete problems. They note that even if one cares about the average-case complexity of a problem with respect to a specific distribution (e.g., the uniform one) then it needs not be the case that a worst-case to average-case reduction must make queries according to this distribution, and that the distribution of queries may depend on the input to the reduction, and so statistics regarding it cannot be given as advice. Nonetheless, combining the ideas of [FeFo] with additional ideas (some borrowed from the study of locally-decodable codes [KaTr]), Bogdanov and Trevisan showed that any *non-adaptive* reduction of (worst-case) $\mathcal{NP}$ to the average-case complexity of $\mathcal{NP}$ (with respect to any sampleable distribution) implies that co$\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$ (using the non-uniformity to encode similar statistics as in [FeFo]).

Although [BoTr] state that a main motivation of their work is the question of basing one-way functions on worst-case $\mathcal{NP}$-hardness, its focus in technique (like that of [FeFo]) is on *decision problems.* Using known reductions between search and decision problems in the context of distributional problems [BCGL, ImLe], Bogdanov and Trevisan [BoTr] also derive implications on the (im)possibility of basing one-way functions on $\mathcal{NP}$-hardness. In particular, they conclude that if there exists an $\mathcal{NP}$-complete set for which deciding any instance is *non-adaptively* reducible to *inverting a one-way function* (or, more generally, to a search problem with respect to a sampleable distribution), then co$\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$.

The works [BoTr, FeFo] fall short of a general impossibility result in two ways. First, they only consider *non-adaptive* reductions, whereas the celebrated worst-case to average-case reductions of lattice problems (cf. [Aj, MiRe]) are adaptive[2], which seems to illustrate the power of adaptive versus non-adaptive reductions.[3] Second, [BoTr, FeFo] reach conclusions involving a *non-uniform* complexity class (i.e., $\mathcal{AM}_{\text{poly}}$). Non-uniformity seems an artifact of their techniques, and one may hope to conclude that co$\mathcal{NP} \subseteq \mathcal{AM}$ rather than co$\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$. (One consequence of the uniform conclusion is that it implies that the polynomial time hierarchy collapses to the second level, whereas the non-uniform conclusion only implies a collapse to the third level.)

We emphasize that the *techniques* of [BoTr] refer explicitly only to decision problems, and do not relate to the underlying search problems (e.g., inverting a supposedly one-way function). In doing so, they potentially lose twice: they lose the extra structure of search problems and they lose the additional structure of the task of inverting polynomial-time computable functions.

We believe that the study of the possibility of basing one-way functions on worst-case $\mathcal{NP}$-hardness is the most important motivation for the study of worst-case to average-case reductions for $\mathcal{NP}$. In such a case, one should consider the possible gain from studying the former question directly, rather than as a special case of a more general study. Let us begin by pointing out the differences between

[2] Interestingly, we note that the work of [Mi] separate Ajtai's [Aj] reduction into two parts: *non adaptive* worst-case to average-case reduction, and *adaptive* worst-case to worst-case reduction.

[3] We comment that the power of adaptive versus non-adaptive reductions has been studied in various works (e.g., [FFLS, HNOS, BaLa]). It is known that if $\mathcal{NP} \not\subseteq BPE$, then there exists a set in $\mathcal{NP} \setminus \mathcal{BPP}$ that is adaptively random self-reducible but not non-adaptively random self-reducible.

$\mathcal{NP}$-search problems and one-way functions.

1.4 $\mathcal{NP}$ Search Problems vs. One-Way Functions

To illustrate the difference between $\mathcal{NP}$ search problems and the finding an inverse of a polynomial time computable function, we re-formulate the problem of inverting a polynomial-time computable function as follows (or rather spell out what it means in terms of search problems). The problem of (average-case) inverting f on the distribution $f(U_n)$, where U_n denotes the uniform distribution over $\{0,1\}^n$, has the following features:

1. The problem is in $\mathcal{NP}$; that is, the solution is relatively short and given an instance of the problem (i.e., y) and a (candidate) solution (i.e., x), it is easy to verify that the solution is correct (i.e., $y = f(x)$).
2. There exists an efficient algorithm that generates random instance-solution pairs (i.e., pairs (y, x) such that $y = f(x)$, for uniformly distributed $x \in \{0,1\}^n$).
3. We care about the average-case complexity of the problem; that is, the probability that an efficient algorithm given a random (efficiently sampled) instance y (i.e., $y \leftarrow f(U_n)$) finds $x \in f^{-1}(y)$.

Indeed, the first and third items are common to all average-case $\mathcal{NP}$-search problems (with respect to sampleable distributions), but the second item is specific to the context of one-way functions (cf. [Go, Sec. 2.1]). A sampleable distribution of random instance-solution pairs is not necessarily implied by a sampleable distribution of instances. Capitalizing on the second item is the source of our success to obtain stronger (negative) results.

1.5 The Benefits of Direct Study of One-Way Functions

The results presented in this paper indicate the gains of studying the question of basing one-way functions on $\mathcal{NP}$-hardness *directly*, rather than as a special case of a more general study. The gains being, getting rid of the non-uniformity altogether (and replacing non-uniform advice that provide needed statistics with $\mathcal{AM}$ protocols designed to provide these statistics), and obtaining a meaningful negative result for the case of general (adaptive) reductions.

Moreover, working directly with one-way functions allows us to consider *natural* special cases of potential one-way functions and to establish stronger results regarding general (i.e non-adaptive) reductions for them.

In particular, our results isolate the ability to compute (and more generally the ability to verify via an $\mathcal{AM}$ protocol) the number of inverses $|f^{-1}(y)|$ given y, as an important parameter in classifying the complexity of inverting f. We call such functions size-verifiable.

The simplest case of size-verifiable functions is obviously a permutation. Another interesting special case of *regular* functions. Loosely speaking, in such a function f, each image of f has a number of preimages that is (easily) determined by the length of the image. We prove that any reduction (which may be *fully adaptive*) of $\mathcal{NP}$ to inverting a regular polynomial-time computable function that has an efficiently recognizable range (possibly via an $\mathcal{AM}$-protocol) implies co$\mathcal{NP} \subseteq \mathcal{AM}$.

We remark that it was already known in the context of cryptographic constructions (e.g., [GKL, GIL+, DiIm, HHK+]), that it is easier to work with regular functions than with general one way functions. For example, the original construction of cryptographically strong pseudo random generators required one-way permutation[BlMi], followed by a construction which was able to use regular functions by [GKL], and finally culminated in the [HILL] complex construction which could use any one-way function. Our work shows that regularity of a function (or, more generally, size-verifiability) is important also for classifying the complexity of inverting f, and not only the ease of using it within cryptographic constructions.

Finally, we point out that the result we prove for size-verifiable functions holds even if we restrict the reduction to be a *worst-case reduction.* Namely, unless co$\mathcal{NP} \subseteq \mathcal{AM}$, there exist no reductions from worst case NP problems to inverting a size-verifiable polynomial time computable function (This is easily seen as the proof of Theorem 3 never utilizes the fact that the oracle accessed by the reduction is allowed to err on some of the queries). In contrast, it is known that reductions do exist from worst case $\mathcal{NP}$ problems to inverting some (general) polynomial time computable function (see [Go]). This, on one hand, indicates that further ideas will be required to extend our results concerning adaptive reductions to general one-way functions, and on the other hand yields a separation between inverting size-verifiable polynomial time computable functions and inverting general polynmial time computable function (assuming as usual co$\mathcal{NP} \not\subseteq \mathcal{AM}$).

In summary, we hope that this framework of working directly in the context of one-way functions, will lead to resolving the general question of the possibility of basing *any* one-way function on worst-case $\mathcal{NP}$-hardness via *any* reduction. In light of the results of this paper, we are tempted to conjecture an impossibility result (pending, as usual, on co$\mathcal{NP} \not\subseteq \mathcal{AM}$).

Organization of the rest of this work. In Section 2, we provide an overview of our proofs as well as a formal statement of our main results. Appendix A we discuss possible interpretations of our negative results (as well as those of [FeFo, BoTr]). Details of our upper bounds protocols are found in Appendix B.

2 Overview of Results and Proofs

Having observed the potential benefit of working explicitly with the inverting task of a function f, materializing this benefit is the bulk of the technical challenge and the technical novelty of this work.

Let us first clarify what we mean by saying that a decision problem L is (efficiently and randomly) reducible to the problem of inverting a one-way function f. We take the straightforward interpretation (while using several arbitrary choices, like in the threshold determining an inverting oracle):

Definition 1 (inverting oracles and reductions). *A function $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$ is called a* (average-case) f-inverting oracle *if, for every n, it holds that* $\Pr[\mathcal{O}(f(x)) \in f^{-1}(f(x))] \geq 1/2$*, where the probability is taken uniformly over $x \in \{0,1\}^n$. For a probabilistic oracle machine R, we denote by $R^{\mathcal{O}}(w)$ a random variable representing the output of R on input w and access to oracle $\mathcal{O}$, where the probability space is taken uniformly over the probabilistic choices of machine R* (i.e., its randomness)*. A probabilistic polynomial-time oracle machine R is called a* reduction of L to (average-case) inverting f *if, for every $w \in \{0,1\}^*$ and any f-inverting oracle $\mathcal{O}$, it holds that* $\Pr[R^{\mathcal{O}}(w) = \chi_L(w)] \geq 2/3$*, where $\chi_L(w) = 1$ if $w \in L$ and $\chi_L(w) = 0$ otherwise. A reduction R, on input w, may ask polynomially many queries to the inverting oracle. In* adaptive *reductions, later queries may depend on the oracle answers to earlier queries. In* non-adaptive *reductions all queries are computed in advance (based solely on the input w and the random coins of the reduction). For simplicity of presentation, we assume all queries are of length $|w|$.*

A reduction as in Definition 1 may only establish that f is a weak one-way function (i.e., that f cannot be inverted with probability exceeding 1/2 on every input length), which makes our impossibility results even stronger.[4] Throughout this work, the function f will always be polynomial-time computable, and for simplicity we will also assume that it is length preserving (i.e., $|f(x)| = |x|$ for all x).

High-level structure of our proofs and their challenges. Our proofs all work via the contra-positive. Suppose, that there exists a reduction R from deciding an (NP-complete language) L to inverting the function f. We aim to use this reduction to give an AM-protocol for $\overline{L}$. (A similar AM-protocol can be given for L itself, but there is no point in doing so because $L \in \mathcal{NP}$ by hypothesis.)

The main backbone of our AM-protocol for $\overline{L}$ is for the verifier to emulate the reduction R on input w and decide whether or not $w \in \overline{L}$ according to R's output. Of course, the verifier cannot run the reduction fully on its own, because the reduction requires access to an f-inverting oracle. Instead, the

[4] In contrast, the standard definition of one-way function requires that any efficient inverting algorithm succeeds with negligible probability (i.e., probability that is smaller than $1/\text{poly}(n)$ on all but finitely many n's). Here we relax the security requirement in two ways (by requiring more of a successful inverting algorithm): first, we require that the inverting algorithm be successful on any input length, and second that the success probability exceeds 1/2 rather than $1/\text{poly}(n)$.

prover will play the role of the inverting oracle, thus enabling the emulation of the reduction. Needless to say, the verifier will check that all answers are actually f-preimages of the corresponding queries. Since we aim at a constant-round protocol, we send all queries to the prover in one round, which in the case of an adaptive reduction *requires* to send the randomness r of the reduction to the prover. Note that also in the non-adaptive case, we may as well just send r to the prover, because the prover may anyhow be able to determine r from the queries.

The fact that r is given (explicitly or implicitly) to the prover is the source of all difficulties that follow. It means that the prover need not answer the queries obliviously of other queries (or of r), but may answer the queries depending on r. In such a case, the prover behavior is not consistent with any single oracle, and thus the verifier should not trust the prover to emulate the reduction. More concretely, the problem is that whereas an inversion **oracle** for f on query y provides an inverse in $f^{-1}(y)$ independently from other queries, a **prover** may choose which inverse to provide depending on other queries and more generally on r. This is no problem when f is 1-1, as the only a single inverse exists and prover's answer is determined by the query. Indeed, in the special case that f is 1-1 and length preserving, inverting f cannot be $\mathcal{NP}$-hard for rather obvious reasons (as has been well-known for a couple of decades; cf. [Br]).[5] The difficulties arise only in case f is not 1-1.

To illustrate what may happen in case f is not 1-to-1, consider a 2-to-1 function f. Given an arbitrary reduction of L to inverting f, consider a modified reduction that tosses n additional coins $\rho_1, ..., \rho_n$, issues n additional queries, and halts without output if and only if for $i = 1, ..., n$ the i-th additional query is answered with the $(\rho_i + 1)$-st corresponding preimage (in lexicographic order). This reduction works with probability that is very close to the original one, but a cheating prover can always cause its emulation to halt without output.

Forcing the Prover to Act as an Inversion Oracle. Our idea is to force the prover to give an inverse of y which is independent of the randomness r. In particular, we will require that the prover answers each query y with the *smallest inverse* in $f^{-1}(y)$. The difficulty naturally will be in verifying that the inverse provided is indeed the smallest one.

For the rest of this extended abstract, we provide an outline of how this is achieved. First, when the function is size-verifiable, and second when the reduction is non-adaptive. In the following, we denote by q the number of queries made by R, by $R(w, r, a_1, ..., a_{i-1})$ the i-th query made by R on input w and randomness r after receiving the oracle answers $a_1, ..., a_{i-1}$, and by $R(w, r, a_1, ..., a_q)$ the corresponding final decision. Recall that for simplicity, we assume that all queries are of length $n \stackrel{\text{def}}{=} |w|$.

2.1 Size-Verifiable Functions (Adaptive Reductions)

Our aim is to present an AM-protocol for $\overline{L}$, when we are given a general (adaptive) reduction R of the worst-case decision problem of L to average-case inverting f.

Definition 2 (Size Verifiable)**.** *We say that a function* $f\colon \{0,1\}^* \to \{0,1\}^*$ *is* size verifiable *if* $\exists\mathcal{AM}$ *proof system for the set*[6] $\{(y, |f^{-1}(y)|) : y \in \{0,1\}^*\}$.

A natural example of a function that is size verifiable (and for which the relevant set is not known to be in $\mathcal{BPP}$) is the integer multiplication function. That is, we consider the function that maps pairs of integers (which are not necessarily prime or of the same length) to their product. In this case the set $\{(y, |f^{-1}(y)|) : y \in \{0,1\}^*\}$ is in $\mathcal{NP}$ (where, the NP-witness is the prime factorization) but is widely

[5] Intuitively, inverting such an f (which is a search problem in which each instance has a unique solution) corresponds to a decision problem in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ (i.e., given (y, i) determine the i-th bit of $f^{-1}(y)$). Thus, the fact that inverting f cannot be $\mathcal{NP}$-hard (unless $\mathcal{NP} = \text{co}\mathcal{NP}$) is analogous to the fact that sets in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ cannot be $\mathcal{NP}$-hard (again, unless $\mathcal{NP} = \text{co}\mathcal{NP}$). In contrast, in case f is not 1-1, the corresponding decision problems are either not known to be in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ or are *promise problems* (cf. [ESY]) in the "promise problem class" analogue of $\mathcal{NP} \cap \text{co}\mathcal{NP}$. Recall that promise problems in the latter class *may be* $\mathcal{NP}$-hard even if $\mathcal{NP} \neq \text{co}\mathcal{NP}$ (see [ESY]).

[6] Or, more generally, f is size verifiable if $\exists\mathcal{AM}$ proof system for the set $\{(y, s) : y \in \{0,1\}^*, s \in (1 \pm \Theta(1))\,|f^{-1}(y)|\}$.

believed not to be in $\mathcal{BPP}$ (e.g., it is believed to be infeasible to distinguish product of two $(n/2)$-bit random primes from the product of three $(n/3)$-bit long random primes).

Another example of a size-verifiable function are regular functions with efficiently recognizable range (possibly via an AM-protocol). Recall that, loosely speaking, for regular functions the number of pre-images is efficiently determined by the input length. More generally, f is size-verifiable, when the number of pre-image is determined by the input length, even if the number of pre-images cannot be *efficiently computed*[7] Likewisse, the "approximable preimage-size" function of [HHK+]) are size verifiable.

We show it is unlikely that size-verifiable one-way functions can be based on NP-hardness.

Theorem 3 (Adaptive Reductions)**.** *Unless* co$\mathcal{NP} \subseteq \mathcal{AM}$*, there exists no reduction* (even not an adaptive one) *from deciding an NP-complete language to inverting a* size-verifiable *polynomial-time computable function.*

Let us proceed to give an outline of the proof. We will make the simplifying assumption throughout this outline that the verifier can even compute size of the set of f-preimages for any string y on its own. The analysis can be easily extended to the case that the verifier can only check the correctness of the size claimed and proved by the prover.

Protocol for a very simple case: As a warm-up we first assume that $|f^{-1}(y)| \leq \text{poly}(|y|)$, for every y. In this case, on common input w, the parties proceed as follows.

1. The verifier selects uniformly coins r for the reduction, and sends r to the prover.
2. Using r, the prover emulates the reduction as follows. When encountering a query y, the prover uses the lexicographically first element of $f^{-1}(y)$ as the oracle answer (and uses $\perp$ if $f^{-1}(y) = \phi$). Thus, it obtains the corresponding list of queries $y_1, ..., y_q$, which it sends to the verifier along with the corresponding sets $f^{-1}(y_1), ..., f^{-1}(y_q)$.
3. Upon receiving $y_1, ..., y_q$ and $A_1, ..., A_q$, the verifier checks, for every i, that $|A_i| = |f^{-1}(y_i)|$ and that $f(x) = y_i$ for every $x \in A_i$. Letting a_i denote the lexicographically first element of A_i, the verifier checks that $R(w, r, a_1, ..., a_{i-1}) = y_i$ for every i. The verifier accepts w (as a member of $\overline{L}$) if and only if all checks are satisfied and $R(w, r, a_1, ..., a_q) = 0$.

Note that the checks performed by the verifier "force" the prover to emulate a uniquely determined (perfect) inverting oracle (i.e., one that answers each query y with the lexicographically first element of $f^{-1}(y)$). Thus, the correctness of the reduction implies the completeness and soundness of the above AM-protocol.

The idea when the size of $f^{-1}(y)$ is not bounded by a polynomial: In general, however, the size of $f^{-1}(y)$, for y in the range of f may not be bounded by a polynomial in n (where $n = |y| = |w|$). In this case, the prover cannot send the entire set $f^{-1}(y)$ to the verifier. The natural idea is to have the verifier send an adequate random hash function $h : \{0,1\}^n \to \{0,1\}^\ell$ and let the prover answer with $h^{-1}(0^\ell) \cap f^{-1}(y)$ (rather than with $f^{-1}(y)$), where $\ell = \lfloor (\log_2 |f^{-1}(y)|/\text{poly}(n)) \rfloor$. The problem is that in this case the verifier cannot check the "completeness" of the list of preimages (because it cannot compute $|h^{-1}(0^\ell) \cap f^{-1}(y)|$), which allows the prover to omit a few members of $h^{-1}(0^\ell) \cap f^{-1}(y)$ at its choice. This freedom of choice (of the prover) may obliterate the soundness of the protocol.

To overcome this difficulty, we will require the prover to prove sizes $\left|h^{-1}(0^\ell) \cap f^{-1}(y)\right|$; that is, prove both lower and upper bounds on this size. For proving lower bounds we use the Goldwasser-Sipser protocol, whereas for proving upper bounds we use the confidence-by-comparison (aka, CBC) protocol as follows. Although we have no way of determining the size of $h^{-1}(0^\ell) \cap f^{-1}(y)$, we do know that its *e*xpected size is exactly $|f^{-1}(y)|/2^\ell$, where the expectation is taken over the choice of h (assuming indeed that a random h maps each point in $\{0,1\}^n$ uniformly on $\{0,1\}^\ell$). Furthermore, the

[7] Let us sketch a size verification protocol for such functions. Since the number of pre-images is equal over all y's, it suffices to estimate $\left|f^{-1}(y)\right|$ on any arbitrary y. So, the verifier may choose a random x and send $y = f(x)$, for which the prover claims and proves the size of $\left|f^{-1}(y)\right|$ by utilizing the lower and upper bound protocols of [GoSi, AiHa] respectively.

prover cannot add elements to $h^{-1}(0^\ell)\cap f^{-1}(y)$ (because the verifier can verify membership in this set), it can only omit elements. Repeating the process many times, if the prover cheats (that is, omits at least one element) in many of the sets, then it ends-up sending sets that are –on average– noticeably[8] smaller than their expected size; whereas, for honest prover, the average should roughly match the expectation. The verifier rejects if the averages deviate from the expectation by too much. This guarantees that the on most repetitions, the prover does not cheat (with high probability).

We remark that the intuition underlying the CBC protocol is inspired by the main idea of Feigenbaum and Fortnow [FeFo]. The common thread is identifying a situation in which the prover may cheat (without being detected) only in one direction[9], and deducing that frequent cheating by the prover leads to a (significant) deviation from some expected statistic. In [FeFo, BoTr] the necessary statistics were gives as *advice*, thus yielding an $\mathcal{AM}_{\rm poly}$ protocol; while, in our work, no statistics are given, and instead, we acquire them via an $\mathcal{AM}$ protocol. The required statistics for our settings are, naturally, completely different than those used in [FeFo, BoTr].

Protocol for the general case: In the following protocol we use families of hash functions of very high quality (e.g., poly(n)-wise independent ones). Specifically, in addition to requiring that a random $h:\{0,1\}^n\to\{0,1\}^\ell$ maps each point uniformly, we require that, for a *suitable polynomial* p and for any $S\subseteq\{0,1\}^n$ of size at least $p(n)\cdot 2^\ell$, with overwhelmingly high probability over the choice of h it is the case that $|h^{-1}(0^\ell)\cap S|<2|S|/2^\ell$. In particular, the probability that this event does not occur is so small that, when conditioning on this event, the expected size of $|h^{-1}(0^\ell)\cap S|$ is $(1\pm 2^{-n})\cdot|S|/2^\ell$. (Thus, under this conditioning and for S as above, the variance of $2^\ell|h^{-1}(0^\ell)\cap S|/|S|$ is less than 2.)

1. The verifier selects uniformly $m=n\cdot q2p(n)2=\text{poly}(n)$ sequences of coins, $r^{(1)},...,r^{(m)}$ for the reduction, and sends them to the prover. In addition, for each $k=1,...,m$, $i=1,...,q$ and $\ell=1,...,n$, it selects and sends a random hash function $h_{k,i,\ell}:\{0,1\}^n\to\{0,1\}^\ell$.

   To streamline the following description, for $j\leq 0$, we artificially define $h_{k,i,j}$ such that $h_{k,i,j}^{-1}(0^j)\stackrel{\rm def}{=}\{0,1\}^n$. In such a case, $S\cap h_{k,i,j}^{-1}(0^j)=S$, and so an instruction to do something with the former set merely means using the latter set.

2. For every $k=1,...,m$, the prover uses $r^{(k)}$ to emulate the reduction as follows. When encountering the i-th query, $y_i^{(k)}$, it determines $\ell_i^{(k)}=\lfloor(\log_2|f^{-1}(y_i^{(k)})|/p(n))\rfloor$, and uses the lexicographically first element of $f^{-1}(y_i^{(k)})\cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})$ as the oracle answer (and uses $\perp$ if the latter set is empty).

   Thus, it obtains the corresponding list of queries $y_1^{(k)},...,y_q^{(k)}$, which it sends to the verifier along with the corresponding sets $f^{-1}(y_1^{(k)})\cap h^{-1}_{k,1,\ell_1^{(k)}}(0^{\ell_1^{(k)}}),...,f^{-1}(y_q^{(k)})\cap h^{-1}_{k,q,\ell_q^{(k)}}(0^{\ell_q^{(k)}})$.

   We assume that none of the latter sets has size greater than $4p(n)$. Note that the bad event occurs with negligible probability, and in such a case the prover halts and the verifier rejects. (Otherwise, all mq sets are sent in one message.)

3. Upon receiving $y_1^{(1)},...,y_q^{(1)},...,y_1^{(m)},...,y_q^{(m)}$ and $A_1^{(1)},...,A_q^{(1)},...,A_1^{(m)},...,A_q^{(m)}$, the verifier conducts the following checks:

   (a) For every $k=1,...,m$ and $i=1,...,q$, the verifier checks that for every $x\in A_i^{(k)}$ it holds that $f(x)=y_i^{(k)}$ and $h_{k,i,\ell_i^{(k)}}(x)=0^{\ell_i^{(k)}}$, where $\ell_i^{(k)}=\lfloor(\log_2|f^{-1}(y_i^{(k)})|/p(n))\rfloor$. Letting $a_i^{(k)}$ be the lexicographically first element of $A_i^{(k)}$, it checks that $R(w,r^{(k)},a_1^{(k)},...,a_{i-1}^{(k)})=y_i^{(k)}$.

   (b) For every $i=1,...,q$, it checks that

[8] We remark that additive deviations are interchangeable with multiplicative ones, for $|h^{-1}(0^\ell)\cap f^{-1}(y)|=\text{poly}(n)$.

[9] More concretely, in this work as well as in [FeFo], there are some NP-sets (*e.g.*, $h^{-1}(0^\ell)\cap f^{-1}(y)$, in the above) s.t. the prover cannot claim a non-member of the set is a member (because it is required to provide a witness).

$$\frac{1}{m} \cdot \sum_{k=1}^{m} \frac{2^{\ell_i^{(k)}} \cdot |A_i^{(k)}|}{|f^{-1}(y_i^{(k)})|} > 1 - \frac{1}{100q \cdot p(n)} \tag{1}$$

where 0/0 is defined as 1.

The verifier accepts w if and only if all the foregoing checks are satisfied and it holds that $R(w, r^{(k)}, a_1^{(k)}, ..., a_q^{(k)}) = 0$ for a uniformly selected $k \in \{1, ..., m\}$.

For the analysis of the protocol we refer the reader to Appendix C

2.2 Non-Adaptive Reductions, General Functions

We now turn to outline the proof of our second main result. Throughout this section R is always *non-adaptive.*

Theorem 4 (General Functions). *Unless* co$\mathcal{NP} \subseteq \mathcal{AM}$*, there exists no* non-adaptive *reduction from deciding an* $\mathcal{NP}$*-complete language to inverting a polynomial-time computable function.*

Overall idea. In the previous section we showed a protocol which worked when f was size verifiable. Ideally, we would like to show that any polynomial time computable function f is size verifiable. Namely, prove both a lower and upper bound on the size of $f^{-1}(y)$ for a given y. Whereas, known lower-bound protocols (cf. [GoSi]) could be applied to these sets, known upper-bound protocols (cf. [AiHa]) cannot be applied because they require that the verifier has, or can obtain, a random (and secret) member of these sets. In our setting the verifier may not know any inverse let alone a random and secret one.

Still, we will be able to use the protocol of the previous section, when it is combined with an idea of [BoTr] of dividing queries into light and heavy ones. We say y is *light* if $\Pr_r[y = R(x, r)] \leq \lambda \cdot \Pr_z[y = f(z)]$ (for $\lambda \in [2, 3]$ to be chosen at random by the verifier), and y is *heavy* otherwise. Then: (1) we show how to prove upper bounds on $|f^{-1}(y)|$ when the y queried is light; (2) we show how to detect which queries are light and which are heavy via an AM protocol. Finally, (3) we adapt the protocol of the previous section so that the prover answers light queries y –as before– by the smallest inverse in $f^{-1}(y)$, and answers heavy queries by $\perp$. Size verification is then applied only to light queries. We claim that the adapted protocol still provides an $\mathcal{AM}$ proof for $x \in \overline{L}$, since there exists an oracle O which answers as the prover did above, which still has good success probability: $\Pr_{y \sim f(U_n)}[O(y) \in f^{-1}(y)] \geq 1 - \frac{1}{\lambda} \geq \frac{1}{2}$, and thus, the reduction correctly decides L (w.h.p), even when accessing such an oracle.

The idea of dividing queries into light and heavy ones is due to [BoTr]. What we mean by light is different than in [BoTr]: In our case y is light if the probability it is selected by the reduction $R(x, U_n)$ is smaller than (some constant times) the probability that y was selected according to $f(U_n)$. While, in [BoTr], y is light if the probability it is selected by the reduction $R(x, U_n)$ is greater than some constant threshold. This difference, makes detecting heavy (or, light) queries harder in our settings, because it simultaneously requires both lower and upper bound for the corresponding sets.

We will be able to preform (1) and (2) only for non-adaptive reductions, and thus the theorem is achieved for general functions and non-adaptive reductions.

A central difference between our case and the case of [BoTr], is that while they use *non-uniform advice*, we "gather" the statistics required for accomplishing tasks (1) and (2) above. For the sake of this extended abstract, we focus on showing how we gather the required statistics. For a description of the entire protocol we refer the reader to Appendix D.

Gathering Statistics. Denote by $coins(y) \overset{def}{=} \{r \mid R(w, r) = y\}$ the number of coins leading to a query y. The statistics we need are (approximations of) (a) expectations on $|coins(y)|$ over reduction queries y, (b) expectations on $|f^{-1}(y)|$ over *light* reduction queries y, and (c) the probability that a reduction query is light.

We accomplish (a), *i.e.*, obtain (approximation of) expectations on $|coins(y)|$ over reduction queries y as follows. The verifier samples many *independent* $y = R(w, r)$ (*i.e.*, distributed by $R(w, U_{n'})$). For each such y lower and upper bound are proved using the known protocols of [GoSi, AiHa] (note that r

is a random and secret element in $coins(y)$). The needed statistics are then estimated by averages on these proved sizes.

We now describe how to accomplish (b) and (c). To gather these statistics the verifier samples many *independent* $y = R(w, r)$ (*i.e.*, distributed by $R(w, U_{n'})$). We show (below) how to detect whether the sampled y is heavy or light, as well as to prove sizes of $f^{-1}(y)$ *for the light ones.* Once this is accomplished, the statistics are estimated as follows: The expectations on $|f^{-1}(y)|$ over light are estimated by averages over the light y. The probability that a reduction query is light is estimated by the fraction of light queries among those y's.

We next elaborate on how to detect whether y is heavy or light, and how to prove sizes of $f^{-1}(y)$ for the light y's. First, we address detection of heavy queries. Recall y is heavy if $\frac{coins(y)}{2^{n'}} > \lambda \cdot \frac{|f^{-1}(y)|}{2^n}$, so to prove y heavy, we need to *lower bound* $f^{-1}(y)$ and *upper bound* $coins(y)$. The lower bound is obtained using known protocols (cf. [GoSi]). The upper bound is obtained as follows. We are in settings where we want to upper bound many sets $coins(y)$, such that: (i) we have statistics on the expected size of these sets, and (ii) roughly speaking, these sizes can only be underestimated (this is guaranteed by applying [GoSi]). The latter property imlies that if the prover cheats (*i.e.*, underestimate the size) often, then the average over the claimed values should be noticeably smaller than the expectation. Thus the verifier compares the known statistics to average over values claimed by the prover, and rejects if there is a significant gap between them. This, essentially[10], guarantees that (most) heavy queries are detected.

Second, we address detection of light queries. Recall y is light if $\frac{coins(y)}{2^{n'}} \leq \lambda \cdot \frac{|f^{-1}(y)|}{2^n}$, so to prove y light, we need to *lower bound* $coins(y)$ and *upper bound* $f^{-1}(y)$. The lower bound is obtained using known protocols (cf. [GoSi]). The upper bound on $f^{-1}(y)$ we show how to obtained below. This, essentially, guarantees that (most) light queries are detected.

Finally, we show how to prove sizes of $f^{-1}(y)$ *for light* queries. (This, in particular, provides the upper bound we needed for the detection of light queries.) The idea, following [BoTr], is to "hide" the light queries among (many) queries drawn from $f(U_n)$. Namely, the verifier sends all these queries, together, in random order. Note that for queries $y = f(z)$ drawn from $f(U_n)$ we can upper bound $|f^{-1}(y)|$ using [AiHa]) (because z is a random secret element in the set $f^{-1}(y)$). For the light queries, the two distributions ($R(w, U_{n'})$ and $f(U_n)$) are statistically close, and thus –when y is light– the prover cannot tell what is the distribution from which it was drawn. So, if the prover cheats on a light query, it is likely to cheat on y drawn from $f(U_n)$ –in which case the verifier rejects. Therefore, the prover is forced to be truthful on all light queries, and in particulary, on the y's of interest (that is, the y's drawn from $R(w, U_{n'})$), w.h.p..

Acknowledgments

Dana Moshkovitz is grateful to Muli Safra for supporting her visit to MIT, where this research has been initiated.

[10] We note that in the above description we neglected to account for the inaccuracy in the lower/upper bound protocols. This inaccuracy may, for example, lead to size estimates c, s for $coins(y), f^{-1}(y)$, respectively, such that $\frac{c}{2^{n'}} \leq \lambda \frac{s}{2^n}$ while $\frac{coins(y)}{2^{n'}} > \lambda \frac{f^{-1}(y)}{2^n}$ (or vice versa). In this case, we fail to detect y as heavy (or light). Nonetheless, this issue is resolved (as in [BoTr]) by our *random choice of* λ: with high probability (over our choice of λ), there is a large enough gap between $\frac{coins(y)}{2^{n'}}$ and $\lambda \frac{f^{-1}(y)}{2^n}$, thus the direction of the above inequality cannot be reversed by the small inaccuracy incurred by the lower/upper bound protocols. This inaccuracy could also be problematic when comparing averages to gathered statistics. This, again, is resolved by randomizing the statistics we gather (for details, see Appendix B).

References

[AhRe] D. Aharonov and O. Regev. Lattice Problems in NP intersect coNP. In *45th IEEE Symposium on Foundations of Computer Science*, 2004.

[AiHa] W. Aiello and J. Hastad. Perfect Zero-Knowledge Languages can be Recognized in Two Rounds. In *28th IEEE Symposium on Foundations of Computer Science*, pages 439–448, 1987.

[Aj] M. Ajtai. Generating hard instances of lattice problems. In *28th ACM Symposium on the Theory of Computing*, pages 99–108, 1996.

[BaLa] L. Babai and S. Laplante. Stronger seperations for random-self-reducability, rounds, and advice. In *IEEE Conference on Computational Complexity 1999*, pages 98–104, 1999.

[Ba01] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 106–115, 2001.

[Ba02] B. Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In *43th IEEE Symposium on Foundations of Computer Science*, to appear, 2002.

[BCGL] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the Theory of Average Case Complexity. *Journal of Computer and System Science*, Vol. 44, No. 2, April 1992, pages 193–219.

[BlMi] M. Blum, and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits SIAM Journal on Computing, Vol. 13, 1984, paged 850–864.

[BoTr] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. In *Proc. 44th IEEE Symposium on Foundations of Computer Science*, pages 308–317, 2003.

[Br] G. Brassard. Relativized Cryptography. In *20th IEEE Symposium on Foundations of Computer Science*, pages 383–391, 1979.

[DiIm] G. Di-Crescenzo and R. Impagliazzo. Security-preserving hardness-amplification for any regular one-way function In *31st ACM Symposium on the Theory of Computing*, pages 169–178, 1999.

[ESY] S. Even, A.L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Inform. and Control*, Vol. 61, pages 159–173, 1984.

[FeFo] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993. Extended Abstract appeared in Proc. of IEEE Structures'91.

[FFLS] J. Feigenbaum, L. Fortnow, C. Lund, and D. Spielman. The power of adaptiveness and additional queries in random-self-reductions. *Computational Complexity*, 4:158–174, 1994. First appeared in Proceedings of the 7th Annual IEEE Conference on Structure in Complexity Theory, 1992, pp. 338-346.

[Fo] L. Fortnow. The complexity of perfect zero-knowledge. In S. Micali, editor, Randomness and Computation, volume 5 of Advances in Computing Research, pages 327–343. JAI Press, Greenwich, 1989.

[Go] O. Goldreich. *Foundation of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[GIL+] O. Goldreich, R. Impagliazzo, L.A. Levin, R. Venkatesan, and D. Zuckerman. Security Preserving Amplification of Hardness. In *31st IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[GKL] O. Goldreich, H. Krawczyk and M. Luby. On the Existence of Pseudorandom Generators. *SIAM Journal on Computing*, Vol. 22-6, pages 1163–1175, 1993.

[GVW] O. Goldreich, S. Vadhan and A. Wigderson. On interactive proofs with a laconic provers. *Computational Complexity*, Vol. 11, pages 1–53, 2003. Extended abstract in *28th ICALP*, Springer, LNCS 2076, pages 334–345, 2001.

[GoSi] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 73–90, 1989. Extended abstract in *18th STOC*, pages 59–68, 1986.

[HHK+] I. Haitner, O. Horvitz, J. Katz, C.Y. Koo, R. Morselli and R. Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. To appear in *Eurocrypt*, 2005.

[HILL] J. Hstad, R. Impagliazzo, L.A. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function SIAM J. Comput., Vol. 28 (4), pages 1364-1396, 1999.

[HNOS] E. Hemaspaandra, A.V. Naik, M. Ogiwara, and A.L. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. *Journal of Computer and System Science*, Vol. 53 (2), pages 194–209, 1996.

[ImLe] R. Impagliazzo and L.A. Levin. No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random. In *31st IEEE Symposium on Foundations of Computer Science*, 1990, pages 812–821.

[KaTr] J. Katz and L. Trevisan. On The Efficiency Of Local Decoding Procedures For Error-Correcting Codes. In *32nd ACM Symposium on the Theory of Computing*, pages 80–86, 2000.

[Mi] D. Micciancio Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. SIAM Journal on Computing, Vol. 34 (1), pages 118–169, 2004. [Preliminary versions in STOC 2002 and CCC 2002.]

[MiRe] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th IEEE Symposium on Foundations of Computer Science*, pages 372–381, 2004.

Appendix

A Discussion: interpretations of our negative results

Negative results of the type obtained in this work (as well as in [FeFo, BoTr]) can be interpreted in several ways: The straightforward view is that such results narrow down the means by which one can base one-way functions on $\mathcal{NP}$-hardness. Namely, under the assumption that co$\mathcal{NP}$ is not contained in $\mathcal{AM}$, our results show that (1) *non-adaptive* randomized reductions are not suitable for basing one-way functions on $\mathcal{NP}$-hardness, and (2) that one-way functions based on $\mathcal{NP}$-hardness can not be size verifiable (e.g., cannot be regular with an efficiently recognizable range).

Another interpretation is that these negative results are an indication that (worst-case) complexity assumptions regarding $\mathcal{NP}$ as a whole (i.e., $\mathcal{NP} \not\subseteq \mathcal{BPP}$) are not sufficient to base one-way functions on. But this does not rule out the possibility of basing one-way functions on the worst-case hardness of a subclass of $\mathcal{NP}$ (e.g., the conjecture that $\mathcal{NP} \cap \text{co}\mathcal{NP} \not\subseteq \mathcal{BPP}$). This is the case because our results (as previous ones) actually show that certain reductions of the (worst-case) decision problem of a set S to (average-case) inverting of f imply that $S \in \mathcal{AM} \cap \text{co}\mathcal{AM}$. But no contradiction is obtained if S belongs to $\mathcal{NP} \cap \text{co}\mathcal{NP}$ anyhow. Indeed, the decision problems related to lattices that are currently known to have worst-case to average-case reductions belong to $\mathcal{NP} \cap \text{co}\mathcal{NP}$ (cf. [Aj, MiRe] versus [AhRe]).

Yet another interpretation is that these negative results suggest that we should turn to a more relaxed notion of a reduction, which is uncommon in complexity theory and yet is applicable in the current context. We refer to "non black-box" reductions in which the reduction gets the code (of the program) of a potential probabilistic polynomial-time inverting algorithm (rather than black-box access to an arbitrary inverting oracle). The added power of such (security) reductions was demonstrated a few years ago by Barak [Ba01, Ba02].

B Size Verification $\mathcal{AM}$-Protocols

In this section we present $\mathcal{AM}$ protocols for proving lower and upper bounds on the size of NP sets. We are often satisfied with *approximate bounds*, namely, s such that $s \leq (1+\rho)\,|S|$ or $(1-\rho)\,|S| \leq s$ for small $\rho > 0$. For proving s is a lower bound we have the Goldwasser-Sipser [GoSi] protocol, which is applicable to any $\mathcal{NP}$ set S. In contrast, there is no known protocol for proving s is an upper bound on S. Nevertheless, when imposing extra conditions on S such protocols do exist. We present two new $\mathcal{AM}$ protocols for proving upper bound, each requiring a different extra condition on S, and is thus applicable in different settings. For completeness, we also include the Aiello-Hastad [Fo, AiHa] upper bound protocol (which is applicable in yet other settings). We shall make use of all these protocols in subsequent sections.

Notations: For any $\mathcal{NP}$ set S, we denote by R_S its corresponding $\mathcal{NP}$ relation, namely, $R_S(x,w) = 1$ iff w is an $\mathcal{NP}$-witness for $x \in S$, and $R_S(x,w) = 0$ otherwise.

B.1 Lower Bound Protocol

For completeness, we briefly describe the lower bound protocol of Goldwasser and Sipser [GoSi] proving the size of S is at least some number s (or more precisely, that $s \leq (1+\rho)\,|S|$). The idea behind it is the following: pick a random hash function h mapping $\{0,1\}^n$ to a range Γ of size slightly smaller than s. If $|S| \geq s$, then, with high probability, any member of Γ will have a pre-image by h. If $|S|$ is significantly smaller than s, an arbitrary member of Γ is not likely to have a pre-image by h in S. The exact parameters are detailed in the following theorem.

Theorem 5 (Goldwasser-Sipser [GoSi]). *$\forall n, \rho, \delta$ and $\mathcal{NP}$ set $S \subseteq \{0,1\}^n$, there is an $\mathcal{AM}$ protocol that, on common input $(1^n, (R_S, s))$ (where R_S is the $\mathcal{NP}$-relation corresponding to S), the following holds: For honest prover, if $s = |S|$, the verifier accepts, w.p. at least $1-\delta$; and converesely, for any prover strategy, if $s > (1+\rho)|S|$, the verifier rejects, w.p. at least $1-\delta$.*

B.2 Upper Bound Protocol

There is no known $\mathcal{AM}$ protocol for proving upper bound on the size of arbitrary $\mathcal{NP}$ sets S. Nevertheless, imposing extra conditions on S such protocols do exist. We present three different such protocols, each requiring a different extra condition on S. The first protocol is by Aiello and Hastad [AiHa], and can be applied to sets S for which the verifier has a random and secret (*i.e.*, not known to the prover) member $x \in S$. The next two protocols –*confidence by comparison (CBC)* and *hiding protocol*– are novelties of this work. The CBC protocol is applicable in settings where some (approximate) statistics on the size of sets S drawn from D is known to the verifier. The hiding protocol is applicable in settings where there is another distribution $\widetilde{D}$ which is statistically close to D such that there is an $\mathcal{AM}$ protocol for proving upper bound for sets $\widetilde{S}$ drawn from $\widetilde{D}$.

B.2.1 Aiello-Hastad Upper Bound Protocol

For completeness, we briefly describe the upper bound protocol of Aiello and Hastad [Fo, AiHa]. The work of Aiello and Hastad showed that if the verifier is able to sample uniformly a member x within the set, then the verifier can also upper-bound the size of the set. The idea is again to use hash functions: pick a random hash function h mapping $\{0,1\}^n$ to a range Γ of size slightly smaller than the claimed size, s. Given $h(x)$, let the prover guess a short list of candidates for x. If $|S| = s$, there exists such short list with high probability. On the other hand, if $|S|$ is much larger than s, there should many z's in S with $h(z) = h(x)$, and thus the prover has a not-very-high chance to output a list containing x.

This protocol uses significantly the fact it has a uniform x in S and the fact this x is *private* (*i.e.*, not known by the prover). By the work of Goldwasser and Sipser [GoSi], if the Aiello-Hastad protocol is plugged into an *AM* protocol in a setting where the specific $\mathcal{NP}$ set S can indeed be sampled, the private sampling can be replaced by usage of public coins.

The exact parameters of the protocol (taken from [BoTr]) are detailed in the following theorem:

Theorem 6 (Aiello-Hastad [AiHa]). *$\forall n, \rho, \delta, k > 0$ and $\mathcal{NP}$ set $S \subseteq \{0,1\}^n$, there is an $\mathcal{AM}$ protocol that, given: (i) common input $(1^n, (S, s))$, and (ii) (secret) verifier's input z s.t. $z \in_R S$: for honest prover, if $s \geq |S|$, then the verifier accepts, w.p. at least $1 - \frac{9}{\rho^2 k}$; and conversely, for any prover's strategy, if $s < (1-\rho)|S|$, the verifier rejects, w.p. at least $\frac{1}{6} - \frac{9}{k}$ (where the probabilities are taken over the random coin tosses of the verifier, as well as over the z uniformly distributed within S).*

Note that the verifier may erroneously accept with a very high probability (roughly $\frac{5}{6}$). Nevertheless, it suffices for our needs. In the settings we consider, there are several NP sets, $S_1, \ldots, S_t$ for which the prover claims upper bounds $s_1, \ldots, s_t$; we'd like the verifier to reject if $s_i < (1-\rho)|S_i|$ w.p. at least p where the probability is taken over a random choice of a set S_i as well as the coins of the protocol. The next corollary shows that such an $\mathcal{AM}$ protocol exists.

Corollary 7. *$\forall n, \rho < \frac{1}{3}, \delta$ and $\mathcal{NP}$ sets $S_1, \ldots, S_t \subseteq \{0,1\}^n$, there is an $\mathcal{AM}$ protocol that, on common input $(1^n, \{(R_{S_i}, s_i)\}_{i=1}^t)$ (where R_{S_i} is the $\mathcal{NP}$-relation corresponding to S_i), and (secret) verifier input $\{z_i\}_{i=1}^t$ s.t. $z_i \in_R S_i$, the following holds: For honest prover, if $s_i = |S_i|$ $\forall i$, the verifier accepts, w.p.*

at least $1-\delta$; and converesely, for any prover strategy, if $s_i > (1+\rho)|S_i|$ for more than $\log_{\frac{18}{17}} \frac{1}{\delta}$ of the S_i's, the verifier rejects, w.p. at least $1-\delta$.

Proof. The $\mathcal{AM}$ protocol runs the Aiello-Hastad upper bound protocol t times (in parallel, with independent randomness) with parameter $k \geq \frac{9t}{\rho^2\delta}$, and accept iff the verifier accepts on each of the parallel executions. We show that completeness and soundness hold. For honest prover, if $s_i \geq |S_i|\ \forall i$, then the verifier accepts w.p. at least $(1-\frac{9}{\rho^2 k})^t \geq 1-t\cdot\frac{9}{\rho^2 k} \geq 1-\delta$. Conversely, for any prover's strategy, denote $t_{bad} = \log_{\frac{18}{17}} \frac{1}{\delta}$, and let $s_i < (1-\rho)|S_i|$ for at least t_{bad} of the S_i's. Then the verifier rejects w.p. at least $1-(1-\frac{1}{18})^{t_{bad}} \geq 1-\delta$, because on each of the t_{bad} (independent) executions of the Aiello-Hastad upper bound protocol, the verifier rejects w.p. at least $\frac{1}{6}-\frac{9}{k} \geq \frac{1}{18}$ (where the last inequality is by the choice of k, $\delta \leq 1$, $t \geq 1$ and $\rho < 1/3$). □

B.2.2 Confidence By Comparison (CBC) Protocol for Proving Sizes of $\mathcal{NP}$-sets

The confidence by comparison protocol is applicable when there are many $\mathcal{NP}$ sets $S_1,\ldots,S_t$ drawn out of some distribution D for which some statistics on the sizes of these sets is known. Specifically, the statistics we use are (an approximation for) the expectation $\mathbb{E}_{S\sim D}[|S|]$, when the sets S are of polynomial size, and (an approximation for) expectations of the form $\mathbb{E}_{S\sim D}[\lfloor(\log_{1+\rho}(|S|))\rfloor]$ (or, more precisely, $\mathbb{E}_{S\sim D}[\lfloor(\log_{1+\rho}\frac{|S|}{1+\frac{\rho}{P}\ell})\rfloor]\ \forall \ell \in 0,\ldots,P-1$ for $P = poly(n)$ large enough), when the sets S are of super polynomial size. We stress that the CBC protocol proves (appoximate) sizes of sets, namely, it simultaneously proves both a lower and an upper bound.

At the heart of the CBC protocol lies the following basic statistical principal.

Lemma 8 (Technical Lemma)**.** *Let $n_1,\ldots,n_t \in \{0,\ldots,M\}$ be random variables of expectation μ. Let $s_1,\ldots,s_t$ be integers such that $s_i \leq n_i$. Then, $\forall\delta > 0$, $\eta \geq \frac{2M}{t\sqrt{\delta}}$, w.p. at least $1-\delta$, it holds that: if $s_i = n_i\ \forall i$, then $\mathbf{Avg}[s_i] \geq \mu-\eta$, and conversely, if $\mathbf{Avg}[s_i] \geq \mu-\eta$, then $s_i \neq n_i$ for at most 2η-fraction of the n_i's (where $\mathbf{Avg}[s_i] \stackrel{def}{=} \frac{1}{t}\sum_{i=1}^{t} s_i$).*

Proof. By Chebyshev inequality, $\Pr[|\mathbf{Avg}[n_i]-\mu| > \eta] < \frac{\sigma^2}{\eta^2} \leq \delta$ (where the last inequality is derived by bounding the variance of the n_i by $\frac{4M^2}{t^2} \leq \delta\eta^2$). So, if $s_i = n_i\ \forall i$, then $\mathbf{Avg}[s_i] > \mu-\eta$, w.p. at least $1-\delta$. Conversely, if $s_i \neq n_i$ for more than 2η-fraction of the n_i's, then $\mathbf{Avg}_i[s_i] \leq \mathbf{Avg}_i[n_i]-2\eta$ (because $s_i \leq n_i - 1$ whenever $s_i \neq n_i$). Applying Chebyshev, again, we get that $\mathbf{Avg}_i[s_i] < \mu+\eta-2\eta \leq \mu-\eta$, w.p. at least $1-\delta$. □

Remark 9. *For settings where we only know μ' that approximates the expectation μ, we shall use the following variant of the above lemma. Let n_i, s_i, δ, η as above, and μ' s.t. $|\mu'-\mu| < \eta$, then w.p. at least $1-\delta$, it holds that: if $s_i = n_i\ \forall i$, then $\mathbf{Avg}[s_i] \geq \mu'-2\eta$, and conversely, if $\mathbf{Avg}[s_i] \geq \mu'-2\eta$, then $s_i \neq n_i$ for at most 4η-fraction of the n_i's.*

For settings where –in addition to the above– we are only guaranteed that $(1-p)$-fraction of the s_i's satisfy $s_i \leq n_i$, while the rest of the s_i's are arbitrary in $\{0,\ldots,M\}$, we use the following variant of the above lemma. Let n_i, δ, η, μ' as above, and $s_i \in \{0,\ldots,m\}$ s.t. $s_i \leq n_i$ for at least $(1-p)$-fraction of the s_i's, then, w.p. at least $1-\delta$, it holds that: if $s_i = n_i\ \forall i$, then $\mathbf{Avg}[s_i] \geq \mu'-2\eta$, and conversely, if $\mathbf{Avg}[s_i] \geq \mu'-2\eta$, then $s_i \neq n_i$ for at most $(4\eta+pM)$-fraction of the n_i's

Our CBC protocol for estimating sizes for $\mathcal{NP}$ sets relates to the above principal in the sense that, when the prover can only *underestimate* the size of $S_1,\ldots,S_t$, then cheating on many of those sets results in a noticible gap between known statistics on $|S_i|$ and averages over the sizes claimed by the provers. In the following, we first consider sets of polynomial size (that is, sets for which the prover can efficiently send all their elements to the verifier), and later consider sets of arbitrary size. For polynomial size sets, we give a protocol that proves the *exact* sizes, while for arbitrary size sets, the protocol we give only prover *approximate* sizes.

Theorem 10 (CBC proving *exact* sizes for polynomial size sets)**.** *$\forall n, t$, let $S_1, \ldots, S_t$ be $\mathcal{NP}$ sets drawn from a distribution D_n over $S \subseteq \{0,1\}^n$; denote by R_{S_i} the corresponding $\mathcal{NP}$ relations. $\forall \delta > 0, \eta \geq \frac{2\max|S_i|}{t\sqrt{\delta}}$, there is an $\mathcal{AM}$ protocol that, given common input $(1^n, \{(R_{S_i}, s_i)\}_{i=1}^t, \mu')$, if $|\mu' - \mathbb{E}_{S\sim D_n}[|S|]| < \eta$, then the following hold:*

- *Completeness: For honest prover, if $s_i = |S_i|$ $\forall i$, then the verifier accepts, w.p. at least $1-\delta$*
- *Soundness: For any prover, if $s_i \neq |S_i|$ for at least a 4η-fraction of the S_i's, then the verifier rejects, w.p. at least $1-\delta$.*

(where the probabilities are taken over the coins of the protocol as well as the random $S_i \sim D_n$).

Proof. We first give the AM protocol and the prove its correctness.

The Protocol

1. P: $\forall S_i$, send all pairs (x, w) of elements $x \in S_i$ with their $\mathcal{NP}$ witnesses w.
2. V: Accept iff the two following conditions hold:
   - For each S_i, and each (x, w), w is indeed a witness for $x \in S_i$, and
   - For each S_i, denote by s_i the number of pairs (x, w) sent for S_i, then

$$\mathbf{Avg}[s_i] \geq \mu' - 2\eta$$

   (where $\mathbf{Avg}[s_i] \overset{def}{=} \frac{1}{t}\sum_{i=1}^t s_i$)

We rely on the Lemma 8 for proving the correctness of this protocol (where the s_i's correspond to the s_i's in the lemma, and the true sizes $|S_i|$'s correspond to the n_i's in the lemma). The conditions of the lemma hold: s_i's are integers s.t. $s_i \leq |S_i|$, $|\mu' - \mathbb{E}_{S\sim D_n}[|S|]| \leq \eta$, and $\eta \geq \frac{2\max|S_i|}{t\sqrt{\delta}}$. Therefore, if $s_i = |S_i|$ $\forall i$, then the verifier accept, w.p. at least $1-\delta$; and conversely, if $s_i \neq |S_i|$ for at least 4η-fraction of the S_i's then the verifier rejects, w.p. at least $1-\delta$. □

We now address $\mathcal{NP}$ sets of (potentially) super-polynomial size. Note that for such sets, the prover can no longer send all their elements to the verifier. Our result for these sets is weaker than the one in Theorem 10 in the sense that we prove only *approximate* sizes (w.h.p.), and not *exact* sizes.

We now point out two difficulties that arise in generalizing the above protocol as to handle sets of super polynomial size, and explain our approach for resolving them. The first difficulty arises already when considering the completeness of the above protocol. For completeness to hold, we need the averages to be close to the expectation (w.h.p.). However, when we have polynomially many sets of super polynomial size, their average size $|S_i|$ is not guaranteed to be close enough to the expected size. This difficulty can be resolved fairly easily. Observe that it suffices to know $\lfloor(\log_{1+\rho}|S_i|)\rfloor$ for obtaining $(1+\rho)$-approximate value for $|S_i|$. Therefore, to have good completeness, we –essentially– augument the above protocol as to compare averages of $\lfloor(\log_{1+\rho}|S_i|)\rfloor$ to their expected value (instead of comparing averages over $|S_i|$). Note that this requirs changing the statistics that are given to our protocol as input to be $\mathbb{E}_{S\sim D_n}[\lfloor(\log_{1+\rho}|S|)\rfloor]$ (instead of $\mathbb{E}_{S\sim D_n}[|S|]$). We actually use a slightly different translation of $|S_i|$ to smaller values that allows us to also handle the second difficulty, as we describe next.

The second difficulty is that we can no longer prevent the prover from somewhat *overestimating* $|S_i|$. This is because now only know how to establish *approximate* lower bounds $s \leq (1+\rho')\,|S|$ using the Goldwasser-Sipser protocol. This is in contrast to the *exact* lower bounds that we established in the former case by having the prover send *all* elements in the set to the verifier. To resolve this problem, recall that it suffices to know sizes of the form $\lfloor(\log_{1+\rho}|S|)\rfloor$ (and not the exact size $|S|$); and observe that, as long as $\lfloor(\log_{1+\rho}|S|)\rfloor = \lfloor(\log_{1+\rho}(1+\rho')\,|S|)\rfloor$, the potential overestimation of $|S|$ makes no difference. Therefore, we only need to handle the case that $|S|$ falls in this "bad interval" in which $\lfloor(\log_{1+\rho}|S|)\rfloor \neq \lfloor(\log_{1+\rho}(1+\rho')\,|S|)\rfloor$. To avoid having $S_1,\dots,S_t$ fall in a bad interval we *randomize the location of those bad intervals* so that with high probability no $S \in S_1,\dots,S_t$ falls in a bad interval. Specifically, we translate the sizes $|S|$ to $\lfloor(\log_{1+\rho}\frac{|S|}{1+\ell\frac{\rho\delta}{t}})\rfloor$ for *random* $\ell \in 0,\dots,\frac{t}{\delta}-1$.

Theorem 11 (CBC proving *approximate* sizes)**.** *$\forall n,t$, let $S_1,\dots,S_t$ be $\mathcal{NP}$ sets drawn from a distribution D_n over $S \subseteq \{0,1\}^n$; denote by R_{S_i} the corresponding $\mathcal{NP}$ relations. $\forall\delta,\rho>0$ and $\eta \geq \frac{2n\log_{1+\rho}2}{t\sqrt{\delta}}$, there is an $\mathcal{AM}$ protocol that, given common input $(1^n,\{(R_{S_i},s_i)\}_{i=1}^t,\{\mu'_\ell\})$, if $\mu'_\ell = \mathbb{E}_{S\sim D_n}[\lfloor(\log_{1+\rho}\frac{|S|}{1+\ell\frac{\delta\rho}{t}})\rfloor]$ $\forall\ell$ $0,\dots,\lceil(\frac{t}{\delta})\rceil-1$ (or, more generally, $\left|\mu'_\ell - \mathbb{E}_{S\sim D_n}[\lfloor(\log_{1+\rho}\frac{|S|}{1+\ell\frac{\delta\rho}{t}})\rfloor]\right| < \eta$ $\forall\ell$), then the following hold:*

- *Completeness: For honest prover, if $s_i = |S_i|$ $\forall i$, then the verifier accepts, w.p. at least $1-\delta$*
- *Soundness: For any prover, if $s_i \notin (1\pm\rho)\,|S_i|$ for at least a 4η-fraction of the S_i's, then the verifier rejects, w.p. at least $1-3\delta$.*

(where the probabilities are taken over the coins of the protocol as well as the random $S_i \sim D_n$).

Proof. We first give the AM protocol and then prove its correctness.

The Protocol

1. V: Choose random $\ell \in 0,\dots,P-1$ for $P = \lceil(\frac{t}{\delta})\rceil$ uniformly at random.
2. P,V: For each S_i, run the Goldwasser-Sipser lower bound protocol to prove $s_i \leq (1+\rho')\,|S_i|$ for $\rho' \leq \frac{\rho}{P(1+\rho)}$, with confidence $1-\frac{\delta}{t}$.
3. V: Let $I_\ell(s_i) = \lfloor(\log_{1+\rho}\frac{s_i}{1+\ell\frac{\rho}{P}})\rfloor$. The verifier accepts iff

$$\mathbf{Avg}_i[I_\ell(s_i)] \geq \mu'_\ell - 2\eta$$

We remark that to help visualize the situation, for each $\ell = 0,\dots,P-1$ we think of a partition of the range $\{0,\dots,2^n\}$ of possible sizes into buckets $\mathcal{B}^\ell_k = [\tau^\ell_{k-1},\tau^\ell_k)$ with thresholds $\tau^\ell_k = (1+\rho)^k(1+\ell\frac{\rho}{P})$. The $I_\ell(s)$ are then simply the index k of the bucket $\mathcal{B}^\ell_k$ into which s falls.

Completeness: Note that $s_i = |S_i|$ implies $I_\ell(s_i) = I_\ell(|S_i|)$. Therefore, applying Lemma 8 (on the $I_\ell(s_i)$'s and the $I_\ell(|S_i|)$'s), we see that if $s_i = |S_i|$ $\forall i$, then the verifier accepts, w.p. at least $1-\delta$.

Soundness: Assume the verifier accepts, and that $s_i \leq (1+\rho')\,|S_i|$ (this holds w.p. at least $1-\delta$ by the soundness of the Goldwasser-Sipser protocol). We show $s_i \in (1\pm\rho)\,|S_i|$ for at least $(1-4\eta)$-fraction of the S_i's, w.p. at least $1-3\delta$. To prove $s_i \in (1\pm\rho)\,|S_i|$, it suffices to show that $I_\ell(s_i) = I_\ell(|S_i|)$ (because $I_\ell(s_i) = I_\ell(|S_i|)$ implies $s_i \geq (1-\rho)\,|S_i|$, whereas, $s_i \leq (1+\rho)\,|S_i|$ is proven by the Goldwasser-Sipser protocol). To show $I_\ell(s_i) = I_\ell(|S_i|)$, consider first the case where the prover can only underestimate $I_\ell(S_i)$, namely, $I_\ell(s_i) \leq I_\ell(|S_i|)$ $\forall i$. In this case, by Lemma 8, $I_\ell(s_i) = I_\ell(|S_i|)$ for at least $(1-4\eta)$-fraction of the S_i's, w.p. at least $1-\delta$. We next show that indeed the prover can only underestimate $I_\ell(|S_i|)$, w.p. at least $1-\delta$. Observe that the prover can overestimate $I_\ell(|S_i|)$ only if $I_\ell(|S_i|) \neq I_\ell((1+\rho')\,|S_i|)$ (because in this case, the slight overestimation in the Goldwasser-Sipser protocol may lead to a false value: $I_\ell(s_i) \neq I_\ell(|S_i|)$). By Lemma 11.1 below, for each fixed S_i, this happens with probability at most $\frac{1}{P}$ (where the probability is taken over the random choice of ℓ), thus, the prover cannot overestimate any $S_i \in S_1, \ldots, S_t$, w.p. at least $1 - t\frac{1}{P} \geq 1 - \delta$. Put together, we conclude that $I_\ell(s_i) = I_\ell(|S_i|)$ for at least $(1-4\eta)$-fraction of the S_i's, w.p. at least $1-3\delta$, and therefore, $s_i \in (1 \pm \rho)\,|S_i|$ for at least $(1-4\eta)$-fraction of the S_i's, w.p. at least $1-3\delta$

Lemma 11.1. *For a fixed S, $I_\ell(|S|) \neq I_\ell((1+\rho')\,|S|)$ w.p. at most $\frac{1}{P}$ (where the probability is taken over the random choice of $\ell \in 0, \ldots, P-1$).*

Proof. We say that a threshold $\tau = \tau_k^\ell$ is *bad with respect to S* if $|S| \leq \tau$ but $(1+\rho')\,|S| > \tau$. Then, it suffices to show that for each S, there is at most one $\ell \in 0, \ldots, P-1$ such that τ is bad with respect to S. To show this, we show that if τ is bad for S then no other threshold τ' is bad for S. Specifically, denote by τ^- the threshold closest to τ from the left, then we show that if τ is bad w.r. to S then $|S| > \tau^-$. This implies that: (i) $\tau_{k'}^{\ell'}$ is not bad w.r. to S for any $\tau_{k'}^{\ell'} < \tau$ (because $|S| \not\leq \tau_{k'}^{\ell'}$ for any $\tau_{k'}^{\ell'} \leq \tau^-$); and (ii) $\tau_{k'}^{\ell'}$ is not bad w.r. to S for any $\tau_{k'}^{\ell'} > \tau$ (because otherwise, $|S| > \tau$ in contradiction to τ being bad w.r. to S). To see that $|S| > \tau^-$ we note that when S is bad for τ then $(1+\rho')\,|S| > \tau$, or equivalently, $|S| > \frac{\tau}{1+\rho'}$. So, it suffices to show that $\tau^- \leq \frac{\tau}{1+\rho'}$. A simple check shows that this is indeed the case (for our choice of $\rho' \leq \frac{\rho}{(1+\rho)P}$). ■ □

B.2.3 Hiding Protocol for Proving Upper Bound

The hiding protocol is used for proving upper bound on a *random* $\mathcal{NP}$ set S that is *not* given as common input, but rather, S is *known only to the verifier*, who draws it out of some distribution D_n. The hiding protocol is applicable when the verifier can also sample sets from another distribution $\widetilde{D}_n$ that has the following two properties: (a) There exists an $\mathcal{AM}$ protocol for proving upper bound on sets drawn from $\widetilde{D}$ (this protocol could be, for example, the Aiello-Hastad protocol, or our CBC protocol), and (b) $\widetilde{D}_n$ is statistically close to D_n in the sense that $\forall S \subseteq \{0,1\}^n$, $\Pr_{S\sim D_n}[S] \leq \lambda \cdot \Pr_{S\sim\widetilde{D}_n}[S]$ for $1 \leq \lambda = O(1)$.

The Hiding Protocol essentially runs as follows. The verifier sends the set of interest S *"hidden"* within many sets drawn from $\widetilde{D}_n$. The prover sends *claims* for the sizes of *all* these sets (including the set S). Subsequently, the verifier reveals which sets were drawn from $\widetilde{D}_n$, and *for those sets* upper bound is *proven* (using the upper bound protocol for sets drawn from $\widetilde{D}_n$).

We now explain why this protocol proves upper bound on the set S drawn from D_n, and not only on the sets drawn from $\widetilde{D}_n$. The heart of the matter is that, for any set S_i received by the prover, the probability that S_i was drawn from $\widetilde{D}_n$ is at least as high as the probability it was drawn from D. (This is true, infomation theoretically.) Therefore, a dishonest prover is *just as likely* to cheat on samples from $\widetilde{D}_n$ as on S. But, when the prover cheats on samples from $\widetilde{D}_n$, the verifier rejects, w.h.p. (by the soundness of the upper bound protocol for sets drawn from $\widetilde{D}_n$); thus, if the prover cheats on S, then the verifier rejects, w.h.p..

For simplicity of the presentation, in the Theorem below, we describe settings that do not work with using the CBC protocol as the upper bound protocol for sets drawn from $\widetilde{D}_n$. (This is because, the $\mathcal{AM}$-protocol for $\widetilde{D}_n$ described there succeed on input of a single set and not many sets as in our CBC protocol). Nonetheless, the theorem can easily be generalized, to work with the CBC protocol as well.

Theorem 12 (Hiding Protocol)**.** *For any $n, \rho, \delta > 0$, the settings for the Hiding Protocol are as follows. Let $D_n, \widetilde{D}_n$ be two sampleable distributions on $\mathcal{NP}$ sets $S \subseteq \{0,1\}^n$, such that: (a) There exists an $\mathcal{AM}$ protocol that, on input $(1^n, (R_{\widetilde{S}}, \widetilde{s}))$ (where $R_{\widetilde{S}}$ is the $\mathcal{NP}$-relation for an $\mathcal{NP}$ set $\widetilde{S}$ drawn at random from $\widetilde{D}$), for honest prover, if $\widetilde{s} = \left|\widetilde{S}\right|$, the verifier accepts, w.p. at least $1-\delta$, and conversely, for any prover strategy, if $\widetilde{s} < (1-\rho)\left|\widetilde{S}\right|$, the verifier rejects, w.p. at least $1-\delta$. (b) $\widetilde{D}_n$ is statistically close to D_n in the sense that $\forall S \subseteq \{0,1\}^n$, $\Pr_{S\sim D_n}[S] \leq \lambda \cdot \Pr_{S\sim\widetilde{D}_n}[S]$ for $1 \leq \lambda = O(1)$.*

In settings as above, there is an AM protocol that, given common input 1^n, the prover claims and proves an upper bound $s \geq (1-\rho)\,|S|$ for a random set S sampled from D by the verifier, that is:

- *Completeness: For honest prover, $s = |S|$ and the verifier accepts, w.p. at least $1-\delta$*
- *Soundness: For any prover, if $s < (1-\rho)\,|S|$, then the verifier rejects, w.p. at least $1-\delta$.*

Proof. We first give the AM protocol and then prove its correctness. Denote $k = \lceil(\lambda)\rceil$ and $k' = \lceil(\frac{2k}{\delta})\rceil$.

The Protocol

1. V: Sample $S \sim D_n$ and $\widetilde{S}_1, \ldots, \widetilde{S}_{k'} \sim \widetilde{D}_n$; send all $k'+1$ sets in random order.
2. P: Upon receiving $S_1, \ldots, S_{k'+1}$, send sizes $s_1, \ldots, s_{k'+1}$ s.t. $s_i = |S_i|$.
3. V: Send the index i_0 of the set sample from D, *i.e.*, $S_{i_0} = S$.
4. P,V: Run (in parallel) the $\mathcal{AM}$ upper bound protocol for sets sampled from $\widetilde{D}_n$ on the sets $S_i\ \forall i \neq i_0$, with confidence $1 - \frac{\delta}{k'}$. Accept iff all parallel runs accept.

Completeness: Completeness follows from the completeness of the $\mathcal{AM}$ protocol for sets sampled from $\widetilde{D}$: if $s_i = |S_i|\ \forall i$, then the verifier accept, w.p. at least $(1-\frac{\delta}{k'})^{k'} \geq 1-\delta$.

Soundness: We first show that $\Pr[i \neq i_0 | S_i] \geq \frac{1}{2}\ \forall i$, and thus, prior to step 3, the prover cannot identify the target set S (*i.e.*, the set S that was drawn from D). Since $\Pr[i \neq i_0|S_i] + \Pr[i = i_0|S_i] = 1$, it suffices to show $\Pr[i \neq i_0|S_i] \geq \Pr[i = i_0|S_i]$. By Bayes Rule, the former is equivalent to showing that $\Pr[S_i|i \neq i_0]\Pr[i \neq i_0] \geq \Pr[S_i|i = i_0]\Pr[i = i_0]$. Now, noticing that the left hand side is $\Pr_{S_i \sim \widetilde{D}_n}[S_i] \cdot \frac{k'}{k'+1}$ and the right hand side is $\Pr_{S_i \sim D_n}[S_i] \cdot \frac{1}{k'+1}$, and recalling that $k' \geq k$, we conclude that this inequality holds for any $D_n, \widetilde{D}_n$ for which $k \cdot \Pr_{\widetilde{D}_n}[S_i] \geq \Pr_{D_n}[S_i]$.

In view of the above, even if the prover "cheats" only on one set S_i (*i.e.*, sends s_i s.t. $s_i < (1-\rho)\,|S_i|$), then $i \neq i_0$, w.p. at least $1 - \frac{1}{k'+1} \geq 1 - \frac{\delta}{2}$. When this is the case, the verifier rejects, w.p. $1 - \frac{\delta}{k'} \geq 1 - \frac{\delta}{2}$. Put together, the verifier rejects, w.p. at least $1-\delta$. □

C Analysis of the Protocol for the Adaptive Case

We first note that the additional checks added to this protocol have a negligible effect on the *completeness* condition: the probability that either $|f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})| > 4p(n)$ for some i,k or that Eq. (1) is violated for some i is exponentially vanishing.[11] Turning to the soundness condition, we note that the checks performed by the verifier force the prover to use $A_i^{(k)} \subseteq T_i^{(k)} \stackrel{\text{def}}{=} f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})$. Also, with overwhelmingly high probability, for every $i = 1,...,q$, it holds that

$$\frac{1}{m} \cdot \sum_{k=1}^{m} \frac{2^{\ell_i^{(k)}} \cdot |f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})|}{|f^{-1}(y_i^{(k)})|} < 1 + \frac{1}{100q \cdot p(n)} \tag{2}$$

Combining Eq. (1) and Eq. (2), and recalling that $A_i^{(k)} \subseteq T_i^{(k)}$ (and $|f^{-1}(y_i^{(k)})| < 2p(n) \cdot 2^{\ell_i^{(k)}}$), it follows that $(1/m) \cdot \sum_{k=1}^{m}(|T_i^{(k)} \setminus A_i^{(k)}|/2p(n)) < 2/(100q \cdot p(n))$ for every i. Thus, for each i, the probability over a random k that $A_i^{(k)} \neq T_i^{(k)}$ is at most $1/25q$. It follows that for a random k, the probability that $A_i^{(k)} = T_i^{(k)}$ for all i's is at least $1-(1/25)$. In this case, the correctness of the reduction implies the soundness of the foregoing AM-protocol.

[11] Recall that here we refer to the case that $A_i^{(k)} = f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})$. Thus, regarding Eq. (1), we note that the l.h.s is the average of m independent random variables, each having constant variance. Applying Chernoff bound, the probability that Eq. (1) is violated is upperbounded by $\exp(-\Omega(m/(100q \cdot p(n))2)) = \exp(-\Omega(n))$.

D Non-Adaptive Reductions (General Functions)

We now turn to outline the proof of our second result.

Theorem 13 (General Functions)**.** *Unless* co$\mathcal{NP} \subseteq \mathcal{AM}$*, there exists no* non-adprtive *reduction from deciding an NP-complete language to inverting a polynomial-time computible function.*

Considering the AM-protocol used in the adaptive case, we note that in the current case the verifier cannot compute (or even directly verify claims about) the size of sets of f-preimages of the reduction's queries. Fortunately, adapting the ideas of [BoTr] to the current setting, allows not only to present a non-uniform AM-protocol for co$\mathcal{NP}$, but even to present a unifom one.

Here R is a *non-adaptive* reduction of some set $L \in \mathcal{NP}$ to the average-case inverting of an arbitrary (polynomial-time computible) function f, and our goal again is to show that $\overline{L} \in \mathcal{AM}$. We may assume, without loss of generality, that the queries of $R(x, \cdot)$ are identically (but typically *not* independently) distributed, and represent this distribution by the random variable R_w; that is, $\Pr[Rx = y] = |\{r \in \{0,1\}^{n'} : R(x,r)\}|/2^{n'}$, where n' denotes the number of coins used by $R(x, \cdot)$.

A simple case (analogous to [FeFo]): We first consider the case that R's queries are distributed identically to $F_n \stackrel{\text{def}}{=} f(U_n)$, where U_n denotes the uniform distribution over $\{0,1\}^n$. In this case, we ask the prover to provide $|f^{-1}(y_i^{(k)})|$ along with each query $y_i^{(k)}$ made in the emulation of $R(x, r^{(k)})$, and ask for lower-bound proofs (cf., [GoSi]) regarding the claimed sizes. To prevent the prover from understating these sizes, we compare the value of $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |f^{-1}(y_i^{(k)})|$ to the expected value of $\log_2 |f^{-1}(f(U_n))|$. Mimicing [FeFo], one may suggest that the latter value (i.e., $\text{Exp}[\log_2 |f^{-1}(F_n)|]$) can be given as a non-uniform advice, but we can di better: We may ask the prover to supply $\text{Exp}[\log_2 |f^{-1}(f(U_n))|]$ and prove its approximate correctness using the following protocol.

> The verifier selects $x_1, ..., x_m$, computes $y_i = f(x_i)$ for every i, sends $y_1, ..., y_m$ to the prover and asks for $|f^{-1}(y_1)|, ..., |f^{-1}(y_m)|$ along with lower and upper bound constant-round interactive proofs. (As usual, the lower-bound AM-protocol of [GoSi] (or [GVW]) can be applied because membership in the corresponding sets can be easily verified.) The upper-bound protocol of [AiHa] can be applied, because the verifier have secret random elements of the corresponding sets.

We note that if the prover understates the set size by more than an $\mathbb{E}$ factor in at least $n/\mathbb{E}$ executions then it gets detected with overwhelmingly high probability. Using a suitable setting of parameters, this establishes the value of $\text{Exp}[\log_2 |f^{-1}(f(U_n))|]$ up to a sufficiently small aditive error, which suffices for our purposes. Specifically, this will force the prover not to understate the value of $|f^{-1}(y_i^{(k)})|$ by more than a $1/10p(n)$ factor for more than $m/10$ of the possible pairs (i,k).

A special case of all light queries : We now allow R_w to depend arbitrarily on w, but restrict our attention to the natural case in which the reduction does not ask a query y with probability that exceeds $\Pr[F_n = y]$ by too much. Specifically, for a threshold parameter t to be determined later, we call a query y t-heavy if $\Pr[R_w = y] > t \cdot \Pr[F_n = y]$. Suppose that all queries of R_w are light. The idea of dividing queries inot light vs. heavy is due to [BoTr], our definition of light/heavy is, however, different. In this case, we modify the foregoing protocol as follows.

> Here it makes no sense to compare $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |f^{-1}(y_i^{(k)})|$ to $\text{Exp}[\log_2 |f^{-1}(F_n)|]$. Instead we should compare the former (empirical) average to $\text{Exp}[\log_2 |f^{-1}(R_w)|]$. Thus, the verifier needs to obtain a good approximation to the latter value. This is done by generating

many y_i's as before (i.e., $y_i = f(x_i)$ for uniformly selected $x_i \in \{0,1\}^n$) along with many more y_i's sampled from R_w, and sending all these y_i's (in random order) to the prover. Specifically, for $t = \max_{y \in \{0,1\}^*}\{\Pr[R_w = y]/\Pr[F_n = y]\}$, we generate t times more y_i's from R_w, and each y_i received by the prover is more likely to come from F_n than from R_w.

The prover will be asked to provide all $|f^{-1}(y_i)|$'s along with lower-bound proofs, and afterwards (i.e., only after commiting to the $|f^{-1}(y_i)|$'s) the verifier will ask for upper-bound proofs for those y_i's generated via F_n (for which the verifier knows a random $x_i \in f^{-1}(y_i)$).

Recall that the prover cannot significantly overstate the size of any $|f^{-1}(y_i)|$ (i.e., overstate it by more than an $\mathbb{E} = 1/\text{poly}(n)$ factor). If the prover significantly understates the sizes of too many of the $|f^{-1}(y_i)|$'s, then it is likely to so overestimate also the sizes of many $|f^{-1}(y_i)|$'s such that y_i was generated by sampling F_n. But in this case, with overwhelimingly high probability, the prover will fail at least one of the corresponding upper-bound proofs.

The general case (both light and heavy queries) : We now allow R_w to depend arbitrarily on w, without any restrictions whatsoever. Observe that the probability that F_n is t-heavy is at most $1/t$, and thus modifying an inverting oracle such that it answers t-heavy queries by $\perp$ effects the inverting probability of the oracle by at most $1/t$. Thus, for $t \geq 2$, if we answer t-heavy queries by $\perp$ (and answer other f-images with a preimage), then we emulate a legitimate inverting oracle (which inverts f with probability $1/2$) and the reduction is still supposed to work well. Referring to y as t-light it it is not t-heavery, we note that t-light queries can be handled as in the foregoing special case (provided $t = \text{poly}(n)$), wheraes t-heavy queries are deal by the previous discussion. The problem is to determine whether a query is t-heavy or t-light, and certainly we have no chance of going so if many (reduction) queries are very close to the threshold (e.g., $\Pr[R_w = y] = (t \pm n^{-\log n}) \cdot \Pr[F_n = y]$). Thus, as in [BoTr], we select the threshold at random (say, uniformly in the interval $[2,3]$). Next, we augment the foregoing protocol as follows.

- We ask the prover to provide for each query $y_i^{(k)}$, also the value of $\Pr[R_w = y_i^{(k)}]$, or equivalently the size of $\{r : R(w,r) = y_i^{(k)}\}$. In addition, we ask for lower-bound proofs of these sizes.
- Using lower and upper bound protocols (analogously to the simple case)[12], we get estimates of $\text{Exp}[\log_2 |\{r : R(w,r) = R_w\}|]$. We let the verifier check that this value is sufficiently close to $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |\{r : R(w,r) = y_i^{(k)}\}|$, thus preventing an understating of the sizes of the latter.

  Hence, combining these two items, the verifier gets a good estimate of the size of $\{r : R(w,r) = y_i^{(k)}\}$ for all but few (i,k)'s.
- Assuming that the value of $\Pr[R_w = y_i^{(k)}]$ are approximalely correct, the verifier makes tentative decisions regarding which of the $y_i^{(k)}$'s is t-light.

  Using a protocol as in the special case, the verifier obtains estimates of $\text{Exp}[\log_2 |f^{-1}(R'_w)|]$, where R'_w denotes R_w conditioned on being t-light. The verifier checks that this value is sufficiently close to the average of $\log_2 |f^{-1}(y_i^{(k)})|$, taken only over t-light $y_i^{(k)}$'s.

Recall that the verifier accepts w if and only if all the foregoing checks are satisfied and $R(w, r^{(k)}, a_1^{(k)}, ..., a_q^{(k)}) = 0$ for a uniformly selected $k \in \{1, ..., m\}$.

[12] In the simple case we got a got estimate of $\text{Exp}[\log_2 ||f^{-1}(F_n)|]$, while relying on our ability to generate samples of F_n along with a uniformly distributed member of $|f^{-1}(F_n)|$. Here we rely on our ability to generate samples of R_w along with a uniformly distributed member of $\{r : R(w,r) = R_w\}$.

Ignoring the small probability that we selected a bad threshold t as well as the small probability that we come accross a query that is close to the threshold, we analyze the foregoing protocol as follows. We first note that for almost all t-light queries, we obtain correct estimates of the size of their f-image. Recalling that, for almost all queries y, we obtained correct estimates of the size of $\{r : R(w,r) = y\}$, it follows that we correctly characterize almost all t-light queries as such. As for (almost all) t-heavy queries y, we may wrongly consider them t-light only if we overestimate the size of their preimage (which is highly improbible in light of the lower-bound proofs and recalling that we have a good estimate of $\{r : R(w,r) = y\}$ for these t's). Thus, except for few queries, all decisions made about these queries are correct, where we refer to the decisions of whether or not they are t-light, and for t-light queries the approximate size of their set of preimages. In particular, for a random $k \in \{1, ..., m\}$, with high probability all decisions regarding all $y_i^{(k)}$'s are correct, in which case the correctness of the reduction implies the completeness and soundness of the foregoing AM-protocol.