A constant-round public-coin protocol for sampling with size, and applications (working draft)

Iftach Haitner* and Mohammad Mahmoody-Ghidary† and David Xiao†

August 31, 2009

Abstract

We present a *constant-round public-coin* protocol between an efficient verifier and an all-powerful (but possibly cheating) prover that achieves the following. Both parties take as input an efficiently verifiable set S, an approximation $s \approx |S|$, and an efficiently computable partition $f : S \to \{0,1\}^*$ where $f(x) = f(x')$ if and only if x, x' belong to the same set of the partition. The verifier is guaranteed to output with high probability a uniformly random $x \leftarrow_{\mathrm{R}} S$ along with a number s_x that approximates the size of the set $|f^{-1}(f(x))|$.

Our main application is to rule out certain black-box reductions from the worst-case hardness of SAT to the task of breaking the binding of a constant-round statistically hiding commitment scheme. We show that the existence of a k-adaptive randomized reduction for the above task implies that **co-NP** has a $O(k)$-round public-coin interactive protocol, namely **co-NP** $\subseteq$ **AM**$[k]$. As corollaries, we derive the following:

1. A $O(1)$-adaptive reduction implies that the *Polynomial Hierarchy* collapses to the second level (Boppana et al., Inf. Proc. Letters 1987).
2. A polylog(n)-adaptive reduction implies the *Exponential Hierarchy* collapses to its third level (Pavan et al., Theor. Comp. Sci. 2007).
3. A $o(n)$-adaptive reduction improves the round-complexity of the best known interactive proof for UNSAT (the language of unsatisfiable formulas).

Our impossibility result holds even if the hiding property of the commitment scheme is only guaranteed to hold against honest receivers. Naturally, our result extends to any primitive that implies constant-round statistically hiding commitment via a black-box proof of security. Most notably, this includes *collision resistant hash functions*, as well as constant-round protocols for *single-server private information retrieval* and *oblivious transfer* with statistical security for one of the parties.

1 Introduction

Much of modern cryptography relies on computational intractability assumptions; starting with seminal works of Diffie and Hellman [DH76] and Goldwasser and Micali [GM84], the security of many if not most modern cryptosystems rests on the assumption that some underlying computational problem is hard to solve efficiently. Often the underlying problem is a concrete number-theoretic or algebraic problems [RSA78, ElG84, Rab79]; unfortunately the existence of sub-exponential algorithms for factoring [BLZ94] and of efficient quantum factoring algorithms [Sho97] have thrown into question whether many of these underlying assumptions are viable, and indeed faster factoring algorithms often translate into better attacks on the cryptosystems based on factoring. In light of this, there has been a search for more robust underlying intractability assumptions.

Besides searching for more robust concrete problems, one way to make the foundations of cryptography more secure is to construct systems based on abstract primitives. This way, cryptosystems can be built via any "instantiation" of the primitive, removing the dependence on any specific intractability assumption. The prime example is one-way functions, which are known to be equivalent to many cryptographic primitives [IL89]. Other examples of such abstract primitives include collision resistant hash functions or public-key cryptosystems, which we believe are strictly more powerful than one-way functions (at least in a black-box fashion [IR89, Sim98]).

*Microsoft Research, New England Campus. E-mail: iftach@microsoft.com

†Department of Computer Science at Princeton University. E-mail: {mohammad,dxiao}@cs.princeton.edu

The holy grail of the project of securing the foundations of cryptography would be to base cryptography on this minimal assumption: namely to show that $\mathbf{P} \neq \mathbf{NP}$ implies the existence of one-way functions, or, even more desirably, the existence of stronger cryptographic primitives such as collision-resistant hash functions or public-key cryptosystems. Other than the fact that $\mathbf{P} \neq \mathbf{NP}$ is necessary for the existence of one-way functions (and all other cryptographic primitives), the former is a "worst-case" assumption while the latter is of "average-case" nature, hence making the first assumption more desirable. In fact, this goal dates back to the seminal paper by Diffie and Hellman [DH76].

The current paper continues the line of research investigating the relation between the existence of cryptographic primitives and **NP**-hardness. As with many previous works, we study whether *black-box reductions* are useful for proving that **NP**-hardness implies the existence of certain cryptographic primitives.

A black-box reduction from **NP**-hardness to the security of a cryptographic primitive is an efficient randomized oracle algorithm R such that the following holds: given *any* oracle A that breaks the security of the cryptographic primitive, the algorithm R^A solves SAT, the language of satisfiable formulas. More precisely, for any x it holds that $\Pr[R^A(x) \neq 1_{\mathsf{SAT}}(x)] \leq 2^{-n}$ over the random coins of R, where $1_{\mathsf{SAT}}(x) = 1$ iff $x \in \mathsf{SAT}$. We say that R is *k-adaptive* if R makes k adaptive rounds of queries to its oracle; each round may consist of many queries, but all of the queries in one round can be computed without looking at the oracle responses to any of the other queries in the same round. We say R is non-adaptive if $k = 1$.

As most constructions and proofs in cryptographic literature are black-box, it is worthwhile to understand whether they can show **NP**-hardness implies one-way functions. As with previous works regarding the existence of black-box reductions from **NP**-hardness to various cryptographic primitives, our results will rule out certain kinds of reductions by showing that their existence implies some implausible consequence, such as $\mathbf{NP} = \mathbf{co\text{-}NP}$ or that the polynomial hierarchy collapses.

In previous work, Brassard [Bra79] observed that if there exists a deterministic black-box reduction from **NP**-hardness to inverting a one-way permutation, then $\mathbf{NP} = \mathbf{co\text{-}NP}$. Bogdanov and Trevisan [BT06], building on earlier work of Feigenbaum and Fortnow [FF93], showed that if there exists a *non-adaptive* randomized black-box reduction from **NP**-hardness to inverting a one-way function, then $\mathbf{NP} \subseteq \mathbf{AM} \cap \mathbf{co\text{-}AM}/\mathrm{poly}$. Here, **AM** is a randomized analogue of **NP** (see Section 2 for definitions), and it is known that if $\mathbf{NP} \subseteq \mathbf{co\text{-}AM}/\mathrm{poly}$, then the polynomial hierarchy to collapse to the third level [Yap83]. Akavia *et al.* [AGGM06] improved this result to show that the same hypothesis implies the uniform conclusion $\mathbf{NP} \subseteq \mathbf{AM} \cap \mathbf{co\text{-}AM}$ (which implies that the polynomial hierarchy collapses to the second level [BHZ87]). [AGGM06] also showed that if there exists an arbitrarily adaptive black-box reduction from **NP**-hardness to inverting one-way functions with *efficiently decidable range and efficiently computable preimage size*, then $\mathbf{co\text{-}NP} \subseteq \mathbf{AM} \cap \mathbf{co\text{-}AM}$. They point out, however, the latter type of one-way functions is "significantly different" than general one-way functions.[1] Finally, Pass [Pas06] takes a different route and showed that if a *specific type* of *witness-hiding* protocol exists, then an arbitrarily adaptive reduction from **NP**-hardness to the existence of one-way functions implies that $\mathbf{co\text{-}NP} \subseteq \mathbf{AM} \cap \mathbf{co\text{-}AM}$. As recently point out by Haitner *et al.* [HRS09], however, it is unlikely that known witness-hiding protocols are of the type required by [Pas06].

Besides the work of [Pas06], common to all these works is a tradeoff between allowing for the most general kind of reduction (ideally highly adaptive ones) while ruling out the weakest kind of primitives (ideally one-way functions). Ruling out (general) one-way functions is clearly a worthwhile goal, but what about adaptivity? It turns out that while most cryptographic reductions we know of are non-adaptive, there are a few notable exceptions, in particular security reductions for building interactive protocols [NOVY98], pseudorandom number generators [HILL99], and certain lattice-based cryptosystems [Ajt03, MR04]. One may hope in particular that lattice problems might someday be used to prove that $\mathbf{P} \neq \mathbf{NP}$ implies one-way functions, since they already exhibit a worst-case to average-case hardness reduction.[2] [BT06, AGGM06] do not rule out the possibility that any of these techniques (or some other adaptive technique)

In order to speak about such adaptive reductions, we need to remove the non-adaptivity restriction from the results of [BT06, AGGM06]. Besides Brassard's observation [Bra79] for the case of one-way permutations, our current results are the first ruling out k-adaptive black-box reductions from **NP**-hardness to many *standard* abstract cryptographic primitives (although, unfortunately, not one-way functions) for the case $k > 1$.[3]

[1] The proof of [AGGM06] also rules out reductions from worst-case problems in **NP** to the worst-case hardness of one-way functions with efficiently computable preimage size. In contrast, observe that (general) worst case hard one-way functions can be based on worst-case problems in **NP** via a black-box reduction.

[2] In particular, the adaptivity of the lattice-based schemes seems essential for giving the best known approximation-ratio required in the worst-case hard lattice problem. Unfortunately, even in the best known reductions the starting worst-case hard problem in the $\mathbf{NP} \cap \mathbf{co\text{-}NP}$.

[3] Notice that [IR89, Sim98] separating public-key encryption and collision-resistance from one-way functions does not say anything about how these primitives relate to **NP**-hardness. This is because their reductions are in an even stronger model called fully-black-box (see for example [RTV04]).

1.1 Statistically Hiding Commitments

Our main result shows that the existence of "moderately adaptive" black-box reductions from **NP**-hardness to constant-round statistically hiding commitments (and, by extension, collision-resistant hash functions) would imply unlikely consequences such as **co-NP** $\subseteq$ **AM** $\cap$ **co-AM**.

A *commitment scheme* is the cryptographic analogue of a safe. It is a 2-party protocol between a Sender algorithm and a Receiver algorithm that consists of two stages. The *commit stage* corresponds to putting an object in a safe and locking it, "committing" the sender to a private message; throughout this work we assume *w.l.o.g.* that the message is a single bit b. The commit stage may be interactive and we measure the round complexity of a commitment scheme by the round complexity of the commit stage.[4] The *reveal stage* corresponds to unlocking and opening the safe, "revealing" the message b to the receiver and "proving" that it was the value committed to in the commit stage. Without loss of generality the reveal phase is non-interactive and the sender simply sends to the receiver the committed message b and the coin tosses it used to generate the transcript of the commit stage.

Commitment schemes have two security properties. The *hiding* property informally says that at the end of the commit stage, an adversarial receiver has learned nothing about the message b, except with negligible probability. The *binding* property says that after the commit stage, an adversarial sender cannot produce valid openings for both b and $1-b$, except with negligible probability. Both of these security properties come in two flavors — *statistical*, where we require security even against a computationally unbounded adversary, and *computational*, where we only require security against feasible (*e.g.*, polynomial-time) adversaries.

Statistical security is preferable to computational security, but it is impossible to have commitment schemes that are both statistically hiding and statistically binding, so we have to settle for one of the two properties being statistical and the other being computational. Statistically hiding (and therefore computationally binding) commitments can be built from one-way functions but only with polynomial round complexity [HR07, HHRS07]. In this paper we focus on *constant-round* statistically hiding commitment (SHC) schemes, which can be based (via constant-adaptive black-box reductions) on specific number-theoretic assumptions [BKK90, BCC88], or, more generally, on any collection of claw-free permutations with an efficiently-recognizable index set [GK96], collision-resistant hash functions [DPP98, NY89], and constant-round protocols for oblivious-transfer and private information retrieval schemes where the security of one of the parties holds information theoretically [HHRS07].

1.2 Our Results

We prove the following:

Theorem 1.1. *If there exists a k-adaptive black-box reduction from* **NP***-hardness to breaking the binding of a constant-round statically-hiding commitment, then* **co-NP** $\subseteq$ **AM**$[k]$.

Here, **AM**$[k]$ is a public-coin interactive proof system between an unbounded prover and an efficient verifier with $O(k)$ rounds of interaction, and **AM** = **AM**$[O(1)]$ (see Section 2 for formal definitions). Our result immediately extends to any cryptographic primitive that implies via black-box constant-adaptive reduction the existence of constant-round SHC, such as collision-resistant hash functions and the other primitives mentioned above.

We can use known results about interactive proofs for **co-NP** to conclude that:

Corollary 1.2. *If there exists an efficient k-adaptive black-box reduction from* **NP***-hardness to breaking the binding of a constant-round statically-hiding commitment, then:*

1. *If $k = O(1)$, then the Polynomial Hierarchy collapses to the second level. [BHZ87]*
2. *If $k = \text{polylog}(1)$, then the Exponential Hierarchy collapses to the third level. [PSSV07]*
3. *If $k = o(n)$, then this improves on the best-known round complexity for an interactive proof for* UNSAT.[5]

As before, the corollary also holds for collision-resistant hash functions and any of the primitives above implying constant-round SHC. It is widely believed that the Polynomial Hierarchy does not collapse, especially not to a low level. The Exponential Hierarchy is less well understood, but it is reasonable to conjecture that it does not collapse to a low level. Whether or not UNSAT has sublinear-round interactive proofs is open, but such a result would be surprising

[4] The round complexity is arguably the most important efficiency measure of the protocol and protocols whose number of rounds is not dependent on the security parameter (*i.e.* constant rounds) are specially interesting.

[5] This statement is sensitive to the particular problem encoding; for a statement that is insensitive to the problem encoding, it suffices to consider $k = n^{o(1)}$.

as there has been no improvement on the round complexity of UNSAT since the original arithmetization protocol of Lund *et al.* [LFKN90]. Thus we interpret Corollary 1.2 as saying that such moderately adaptive reductions are either unlikely to exist or at least very difficult to prove, depending on how much adaptivity is allowed.

Public-coin size-estimating protocols. Our main technical contribution is a new constant-round public-coin size-estimating protocol, which we believe to be of independent interest. Our protocol performs the following sampling task: given an efficiently verifiable set S and an efficient function f, the verifier outputs an almost-uniform sample x from S and a number s_x approximating the number of colliding elements $x' \in S$ such that $f(x) = f(x')$.

We could lower-bound s_x using the Goldwasser-Sipser (GS) public-coin protocol [GS89] (described in Section 2), and we could conceivably upper-bound s_x using the private-coin protocol of Aiello-Håstad [AH91], but the Aiello-Håstad protocol requires the verifier to be able to efficiently sample a random x' colliding with x. In general, even if S is efficiently decidable, it may be hard to sample a colliding x', and so applying the Aiello-Håstad protocol is impossible. Our protocol gives a way to upper-bound set sizes in a context where the Aiello-Håstad protocol cannot be used. Note, however, that rather than estimating the number of collisions for any given $x \in S$, our protocol only guarantees a good estimate for a *randomly chosen* $x \in S$. Our result can be stated as follows:

Theorem 1.3. *(informal) There exists a constant-round public-coin protocol* SampleWithSize *whose parties get, as a common input, an efficiently verifiable set $S \subset \{0,1\}^n$, an efficiently computable function $f : \{0,1\}^n \to \{0,1\}^*$ and a good approximation (i.e., within $(1 \pm \frac{1}{\mathrm{poly}(n)})$ factor) of $|S|$ and has the following guarantee: given that V does not abort, then V outputs a pair (x, s_x), such that x is distributed $(1/\mathrm{poly}(n))$-close to the uniform distribution over S, and s_x is a good approximation for $|f^{-1}(f(x)) \cap S|$.*

In our application, we apply the SampleWithSize protocol recursively. We start with $S_1 = \{0,1\}^n$ and a function $f_1 \colon \{0,1\}^n \to \{0,1\}^*$, to get a random sample $x_1 \in \{0,1\}^n$ and an approximation s_{x_1} for $\left|f_1^{-1}(f_1(x_1)) \cap S\right|$. We then use the approximation s_{x_1} to apply the protocol on $S_2 = f_1^{-1}(f_1(x_1)) \cap S_1$ and a function f_2, and so on. When considering the exact statement of Theorem 1.3 (see Theorem 4.1), it follows that we can only repeat the above process for constant number of times before the accumulated error becomes too large.

1.3 Our Technique

Let us first review the paradigm used for proving the impossibility results of [FF93, BT06, AGGM06]. Assume that we are given a reduction R from deciding SAT to (say) inverting a one-way function f.[6] Note that such a reduction immediately yields a reduction $\overline{R}$ from deciding UNSAT to inverting f (simply emulate R, then flip the output bit). We use $\overline{R}$ to design the following **AM** protocol for UNSAT, which implies $\mathsf{SAT} \in \mathbf{AM} \cap \mathbf{co\text{-}AM}$. In order to decide x, the verifier emulates a random execution of $\overline{R}^A(x)$, where the prover plays the role of the oracle A. When the emulation ends, the verifier outputs the same output that $\overline{R}^A(x)$ does. That the honest prover causes the verifier to accept $x \in \mathsf{UNSAT}$ is straight-forward. Proving that no cheating prover can trick the verifier into accepting $x \notin \mathsf{UNSAT}$, however, is more involved — the prover might deviate arbitrarily from the way any fixed inverter A would act, making $\overline{R}^A$ (and thus the verifier) mistakenly accept.

The previous impossibility results mentioned above take essentially two different approaches for catching cheating provers. The first approach is to equip the verifier with an efficient procedure for checking whether A (played by the prover) returns the right answer or not. Clearly this approach is only applicable in settings where such a procedure exists, for example in the efficiently computable preimage size one-way functions case (second result of [AGGM06]), and under the assumption that a specific type of witness-hiding protocol exists ([Pas06]).

A different approach is needed when the verifier cannot perform such a check. Assume for the sake of simplicity in this discussion that the oracle A responds with single bits (for example a hard-core bit) instead of many-bit strings. [FF93] considered the case where the distribution of $\overline{R}^A(x)$'s queries (induced by its random coins) is uniformly distributed and independent of x. The verifier emulates the execution $\overline{R}^A(x)$ as above, where in addition it checks that the fraction of times the prover answers 1 is approximately the probability $\rho = \Pr_{x \leftarrow_R \{0,1\}^n}[A(x) = 1]$, where the verifier gets ρ as a single non-uniform advice for all x of length n. More accurately, the verifier emulates in parallel many executions of $\overline{R}^A$ (each time with fresh random coins), checks that overall the prover answers 1 approximately a ρ fraction of the time, and if so then the verifier picks one of these executions at random and outputs its answer.

[6] We note that the results of [FF93, BT06] are given in the (more general) settings of worst-case to average-case hardness reductions. Yet, both result can be cast as a worst-case to one-way functions reduction.

This approach was generalized by [BT06] for arbitrary query distributions possibly depending on x (and [AGGM06] gave a version of it without needing advice). Their basic idea is to first learn statistics about the query distribution of $\overline{R}^A(x)$ using statistics about uniform queries. In order to learn this query distribution, the verifier chooses many random queries, and for each query q it invokes an **AM** *size-estimating* protocol for learning the weight of q (i.e., the number of random-coins for the reduction $\overline{R}$ that result in $\overline{R}^A$ asking q). This size-estimating protocol is implemented using the two protocols we saw above: the GS *lower-bound protocol* (Lemma 2.1), and the AH *upper-bound protocol*.

While the lower bound protocol is public-coin, the AH upper bound protocol is *private-coin*: in order to guarantee soundness, the verifier needs to know *private* random-coins leading to q. Such private coins are available to the verifier in the first adaptive round of the reduction, but if the reduction has a second adaptive round the verifier can no longer sample such private coins. This is because the queries in the second round depend on the answers the reduction received for the first round, and since the verifier needs the prover to provide the answers for the queries of the first round, this might give the prover some information about the second round queries. In fact, all the second round queries might be completely determined by the first round queries. Thus, the results of [BT06, AGGM06] (for general one-way functions), hold only for non-adaptive reductions, and there is no known extension to even two rounds of adaptivity.

Using Theorem 1.3 to achieve our main result. We have two advantages in the setting of constant-round SHC that allow us to use SampleWithSize of Theorem 1.3 to bypass the difficulties in [BT06, AGGM06]. First, by the statistically hiding property, we know that given any arbitrary receiver strategy, the honest sender will produce a transcript that can be decommitted to both $b = 0$ and $b = 1$. Second, *any* two revealings suffice to break the binding; there is no additional structure imposed on the binding (for example, the revealings do not have to be related to the receiver's messages in any way except that it should be consistent with the transcript). These two conditions enable us to use SampleWithSize fruitfully in our setting, but unfortunately they do not hold in the case of one-way functions.[7]

More formally, let $\mathsf{Com} = (\mathsf{Sender}, \mathsf{Receiver})$ be a c-round statistically hiding commitment for $c = O(1)$. Let $\overline{R}$ be a black-box reduction that given an oracle A breaking the binding of Com decides UNSAT. As in previous works, we first describe a "canonical adversary", which we call Sam, that breaks Com, which by the correctness of $\overline{R}$ implies that $\overline{R}^{\mathsf{Sam}}$ decides SAT. We then construct **AM** protocol for UNSAT by having the verifier emulate $\overline{R}$ and using the prover to answer the reduction's queries. In order to prevent cheating, we describe an **AM** sub-protocol (using SampleWithSize) that forces the prover to respond as Sam would or else get caught cheating.

The canonical adversary Sam responds to queries as an honest sender committing to 0 would respond. The following interpretation of an honest sender's behavior will be useful: given a history of messages $\tau = (q_1, a_1, \ldots, q_i, a_i)$ (where q_j are receiver messages and a_j are sender messages), and given a new receiver message q_{i+1}, the honest receiver responds by sampling sender random coins ω consistent with τ (namely they would produce the same sender responses as those in τ) and responding with a next message a_{i+1} computed using the history τ and q_{i+1} with random coins ω and secret bit $b = 0$. This is "backwards" from the way things really happen (the honest sender first samples ω once for the entire transcript) but it is not hard to see the distribution over transcripts defined is identical. In the reveal phase, Sam is given a full transcript $\tau = (q_1, a_1, \ldots, q_c, a_c)$, and responds by sampling one set of random coins ω^0 that reveals τ to 0 and another ω^1 that reveals τ to 1.

To emulate Sam, our **AM** protocol will emulate the "backwards sampling" described above using the SampleWithSize protocol. In order to keep the notation simple, we describe the case where the reduction does not rewind Sam. Given a history of receiver messages $Q_i = (q_1, \ldots, q_i)$ and random coins ω, let $f_{Q_i}(\omega) = (a_1, \ldots, a_i)$ be the sender's responses on $q_1, \ldots, q_i$ using random coins ω and secret bit $b = 0$. When $\overline{R}^{\mathsf{Sam}}(x)$ makes a query q_{i+1} to Sam given history $(q_1, a_1, \ldots, q_i, a_i)$, the verifier invokes the SampleWithSize protocol with $S \stackrel{def}{=} f^{-1}_{Q_i}(a_1, \ldots, a_i)$ and $f \stackrel{def}{=} f_{Q_{i+1}}$. The approximation s for $|S|$ is obtained from the previous invocation of SampleWithSize for the query q_i, which output ω_i and $\tilde{s}$ approximating $|f^{-1}_{Q_i}(a_1, \ldots, a_i)|$. SampleWithSize outputs ω_{i+1} and an approximation for $|f^{-1}_{Q_{i+1}}(a_1, \ldots, a_{i+1})|$, which can be used for the next query q_{i+2}. The verifier outputs $f_{Q_{i+1}}(\omega_{i+1})$ as the output of the simulated Sam. A similar sampling procedure allows the verifier to sample ω^0, ω^1 breaking binding for a complete commit-stage transcript.

Theorem 1.3 guarantees that as long as $i = O(1)$ (which is always the case since Com is constant round), the above emulation is accurate enough. It follows that the above is a constant-round public-coin protocol for emulating Sam, and so if the verifier simulates $\overline{R}$ and uses the above protocol to answer oracle queries, this gives an $\mathbf{AM}[k]$ protocol for UNSAT.

[7] Specifically, the first condition does not hold with OWF because given a query y in the image of a OWF f, the prover could fallaciously claim that there exists no x such that $f(x) = y$ and the verifier would not be able to check this claim. The second condition also does not hold for OWF, but to see it one needs to speak of universal one-way hash functions (which are equivalent to OWF [Rom90]), and we defer this discussion to the appendix.

Proving Theorem 1.3. Since we want our protocol to be public coin, we cannot simply approximate $\left|f^{-1}(f(x)) \cap S\right|$ using the GS public-coin lower-bound protocol (Lemma 2.1) together with the AH private-coin upper-bound protocol (as done in [BT06, AGGM06]). Rather, our approach for the public-coin sampling is somewhat more "holistic" than the above. We do not prove a public-coin protocol to upper-bound $\left|f^{-1}(f(x)) \cap S)\right|$ for *every* x; rather, we show that for an almost-*random* x drawn from S, we can upper-bound $\left|f^{-1}(f(x)) \cap S)\right|$. Sampling an almost-uniformly random (in statistical distance) $x \in S$ using an approximation of $|S|$ can be done using poly(n)-wise independent hash functions (as shown by Goldreich *et al.* [GVW01]). Hence, the challenge is to obtain an estimate of $\left|f^{-1}(f(x)) \cap S\right|$.

The reason we can bypass the need for private coins in our setting is because an upper bound on many preimage sets in the domain of f translates into a *lower bound* on the *image* of f. As a simple example, consider the task of computing k when given a function $f : \{0,1\}^n \to \{0,1\}^n$ that is promised to be k-to-1 but for an unknown k (for simplicity let us work over the entire domain, *i.e.* $S = \{0,1\}^n$). Given a claimed regularity k', we can be convinced that $k' \leq k$ by taking any x and using the GS lower bound protocol to prove $k' \leq |f^{-1}(f(x))|$. On the other hand, notice that the image of a k-to-1 function has size $2^n/k$, therefore to show $k' \geq k$ we can use the GS lower-bound protocol to prove that $2^n/k' \leq |f(\{0,1\}^n)|$.

This idea can be generalized to non-regular f, where we compute a more refined description of the regularity structure of f. Namely, our protocol will learn the *histogram* of f with respect to the uniform distribution over S. The histogram is a set of numbers b_z for each $z \in [0,1]$ (appropriately discretized), where $b_z = \Pr_{x \leftarrow S}\left[\frac{\left|f^{-1}(f(x)) \cap S\right|}{|S|} = z\right]$. Notice that a histogram satisfies $\sum_z b_z = 1$, and also implicitly says the size of the image of f is $\sum_z \frac{b_z}{z}$. We will obtain (an approximation of) the histogram using the prover's help, using the same idea as above: to lower-bound a set $|f^{-1}(f(x)) \cap S|$ we can directly apply the GS lower-bound protocol, while to upper-bound sets will lower-bound the image of f. The argument showing how to use the lower-bound on the image to give upper-bounds on the pre-image sets is delicate, but essentially the above intuition above can be formalized to show that some invocation of the GS lower-bound protocols will abort if the prover cheats too much in either direction and gives us a bad approximation of the histogram of f. Once we have a good histogram, it is straightforward to sample a random $x \leftarrow_{\mathrm{R}} S$ and obtain an estimate of $|f^{-1}(f(x)) \cap S|$.

Note that previous works have also used histograms to estimate set sizes, and a related protocol appears in [GVW01]. We emphasize that their protocol accomplishes a different task that is incomparable to ours.[8] To the best of our knowledge, our protocol SampleWithSize has not appeared before in the cryptographic literature.

1.4 Organization

2 Preliminaries

For a random variable X taking values in a finite set $\mathcal{U}$, we write $x \leftarrow \mathcal{U}$ to indicate that x is selected according to the uniform distribution over $\mathcal{U}$. Given two probability distributions X, Y over a common universe $\mathcal{U}$, let $\Delta(X,Y)$ denote their statistical distance, where $\Delta(X,Y) = \max_{T \subseteq \mathcal{U}} |\Pr[T(X)=1] - \Pr[T(Y)=1]|$. Two ensembles of distributions $\{X_n\}, \{Y_n\}$ are *statistically indistinguishable* if $\Delta(X_n, Y_n) \leq n^{-\omega(1)}$ for sufficiently large n. Let $[t] = \{1, \ldots, t\}$. A family of functions $\mathcal{F}_{n,t} = \{f : \{0,1\}^n \mapsto [t]\}$ is d-wise independent if for any $x_1, \ldots, x_d \in \{0,1\}^n$ and $y_1, \ldots, y_d \in [t]$, $\Pr_{f \leftarrow \mathcal{F}_{n,t}}[f(x_1) = y_1, \ldots, f(x_d) = y_d] = t^{-d}$. It is possible to efficiently sample d-wise independent functions for any $d = \text{poly}(n)$, for example by using a random univariate degree $d-1$ polynomial (for better parameters see for example [ABI86]).

2.1 AM protocols

A language L is in $\mathbf{AM}[k]$ if there exists an interactive protocol between an efficient verifier V and an unbounded prover P such that the following hold: for every $x \in L$ it holds that $\Pr[\langle P, V\rangle(x) = 1] \geq 2/3$ and for every $x \notin L$ and any possibly cheating prover P^* it holds that $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/3$. Furthermore, the prover and verifier exchange at most $O(k)$ messages and the verifier's messages are sampled uniformly at random (namely its random coins are "public" to the prover). If $k = O(1)$, then we simply write $L \in \mathbf{AM}$. In general, we will say a protocol (possibly not for a decision problem) is an $\mathbf{AM}$ protocol if it has constant rounds and public coins.

[8] In [GVW01], the protocol lower-bounds the size of a set S that is *efficiently verifiable via a low-communication interactive protocol* but not efficiently decidable. To do so, they recursively refining the histogram such that, if the prover lies about $|S|$ and gives an over-estimate, then at the base of the recursion the verifier catches the cheating by noticing that some parts of the histogram are empty while the prover must claim they are non-empty in order to be consistent with previous answers.

2.1.1 Set estimation and sampling protocols.

Lemma 2.1 (Lower-bound protocol [GS89]). *Let $S \subset \{0,1\}^n$ be a set whose membership can be verified in* $\mathrm{poly}(n)$ *time. Then there exists an* **AM** *protocol with verifier V that gets as input a security parameter 1^n, δ (as the soundness parameter), ϵ (as the approximation parameter), and s (as size of S) running in time* $\mathrm{poly}(n, 1/\epsilon, \log 1/\delta)$ *such that:*

- *If $|S| \geq s$, then there is a prover that makes V accept with probability at least $1-\delta$.*
- *If $(1+\epsilon)|S| \leq s$, then for any (unbounded) prover, V rejects with probability at least $1-\delta$.*

The statement of the lemma holds even if the membership in S can be verified only through an **AM** *protocol itself. We call the latter case the "extended" version of the lemma.*

Lemma 2.2 (Uniform sampling protocol [GVW01, AGGM06]). *Let $S \subset \{0,1\}^n$ be a set whose membership can be verified in* $\mathrm{poly}(n)$ *time. Then there exists an* **AM** *protocol with (an efficient) verifier V that gets as input a security parameter 1^n, $\delta = 1/\mathrm{poly}(n)$ (as the approximation and soundness parameter), $s \in \mathbb{N}$ (as size of S) such that assuming $s \in [(1 \pm \delta/5)|S|]$, then:*

1. *For any (unbounded) prover P either V aborts or outputs some $x \in S$. Moreover, either:*
   - *V aborts with probability at least $1-\delta$, or*
   - *Conditioned on not aborting: x is distributed δ-close to uniform over S.*
2. *There is a prover strategy (the honest prover) such that V aborts with probability at most δ.*

2.2 Statistically Hiding Commitments

A commitment protocol is given by a pair of randomized algorithms $\mathsf{Sender}, \mathsf{Receiver}$. The Sender is given an input bit b and both parties are given 1^n where n is the security parameter. The interaction is divided into two stages: a commit stage and a reveal stage. We say the CSHC protocol is c-round if in the commit stage there are at most $2c$ messages exchanged, and say without lost of generality that the first message is sent by the receiver and the last message is sent by the prover. $\mathsf{Sender}(q_1, \ldots, q_i, \omega, b)$ outputs sender messages $(a_1, \ldots, a_i)$, where $i \leq c$, $q_1, \ldots, q_i$ are receiver messages, ω is a choice of random coin tosses, and b is the committed bit, and likewise $\mathsf{Receiver}(a_1, \ldots, a_i, \omega')$ outputs $(q_1, \ldots, q_{i+1})$.[9] Without loss of generality we assume the reveal stage consists simply of the sender sending his random coin tosses ω, b to the receiver. For notational convenience let us define $\mathsf{Transcript}(q_1, \ldots, q_i, \omega, 0) = (q_1, a_1, \ldots, q_i, a_i)$ where $(a_1, \ldots, a_i) = \mathsf{Sender}(q_1, \ldots, q_i, \omega, 0)$. Let $\langle \mathsf{Sender}, \mathsf{Receiver} \rangle(b)$ denote the distribution of the commit-stage transcript obtained when Sender commits to b with $\mathsf{Receiver}$.

Definition 2.3. A commitment protocol is statistically hiding if it satisfies the following:

1. Hiding: let $\mathsf{Receiver}^*$ be any (possibly computationally unbounded) receiver strategy. Then the two distributions $\langle \mathsf{Sender}, \mathsf{Receiver}^* \rangle(0)$ and $\langle \mathsf{Sender}, \mathsf{Receiver}^* \rangle(1)$ are statistically indistinguishable.
2. Binding: let Sender^* be any computationally efficient sender strategy. Then the probability that Sender^* can decommit $\langle \mathsf{Sender}^*, \mathsf{Receiver} \rangle(b)$ to the opposite bit $1-b$ is negligible.

2.2.1 The canonical adversary Sam ([HHRS07])

For any fixed statistical hiding commitment (SHC) protocol ($\mathsf{Sender}, \mathsf{Receiver}$) (with arbitrary number of rounds), there exists the following (inefficient) canonical adversary Sam (similar in spirit to that of Haitner et al. [HHRS07]).[10] For any partial transcript $\tau = (q_1, a_1, \ldots, q_i, a_i)$ of SHC (where $1 \leq i \leq c$) and $b \in \{0,1\}$, let $\mathsf{Sib}(\tau, b) \stackrel{def}{=} \{\omega \in \{0,1\}^r \mid \mathsf{Sender}(q_1, \ldots, q_i, \omega, b) = (a_1, \ldots, a_i)\}$, where $\mathsf{Sib}(\perp, b) \stackrel{def}{=} \{0,1\}^r$. In words, $\mathsf{Sib}(\tau, b)$ is the set of random coins ω for which the sender responses generated using ω on secret bit b are consistent with τ.

In the commit stage Sam behaves as follows. Given a query q_i, Sam samples uniformly at random $\omega \in \mathsf{Sib}(\tau = (q_1, a_1, \ldots, a_{i-1}, q_{i-1}, 0)$, assuming that Sam was previously asked on $q_1, \ldots, q_{i-1}$ answering $a_1, \ldots, a_{i-1}$. Sam returns

[9] We often find it convenient to think of the behavior of the sender in a stateful way: the sender samples a random ω once for the entire commit stage, and then responds to a query q_i by remembering the previous queries $q_1 \ldots q_{i-1}$ and computing its response a_i according to ω and its secret bit.

[10] Since the following Sam breaks a *specific* SHC ($\mathsf{Sender}, \mathsf{Receiver}$), it will actually be considerably simpler than the oracle of [HHRS07].

a_i from the response $(a_1, \ldots, a_i) = \mathsf{Sender}(q_1, \ldots, q_i, \omega, 0)$ back to $\mathsf{Receiver}$. If $i = c$, Sam also samples uniformly at random $\omega' \in \mathsf{Sib}((\tau, q_i, a_i), 1)$, and includes (ω, ω') in the output.

As [HHRS07] showed, the statistically hiding property of the SHC yields that Sam breaks the binding of Com with save but negligible probability.

2.3 Black-box Reductions

A black-box reduction from deciding a language L to breaking the binding of a SHC is an oracle-aided algorithm R with the following guarantee: given as oracle a *deterministic* and *stateless* adversary $\mathcal{O}$ that breaks the binding of a statistically hiding commitment SHC, $R^{\mathcal{O}}$ decides L (*i.e.*, $\Pr[R^{\mathcal{O}}(x) = 1_L(x)] \geq 1 - 2^{-n}$). We say that R is k-adaptive if it makes k adaptive rounds of queries to its oracle; each round may consist of many queries, but all of the queries in one round can be computed without looking at the oracle responses to any of the other queries in the same round. Notice that if we have such R, then we also have a reduction $\overline{R}$ deciding $\overline{L}$ using $\mathcal{O}$, which simply emulates R and then flips the output bit.

3 Basing Statically Hiding Commitment on NP-Hardness

In this Section we prove our main result.

Theorem 3.1. *Suppose there exists an efficient k-adaptive black-box reduction R using an adversary breaking the binding property of a constant-round* SHC *in order to decide L. Then $\overline{L} \in \mathbf{AM}[k]$.*

Proof of Theorem 3.1. Fix the constant-round $\mathsf{SHC} = (\mathsf{Sender}, \mathsf{Receiver})$ and reduction R. It follows that there us also a reduction $\overline{R}$ for $\overline{L}$. We construct an $\mathbf{AM}$ protocol for $\overline{L}$ by making the verifier emulate $\overline{R}$ and forcing the prover to emulate the canonical adversary Sam. We show that the prover cannot deviate far from Sam without getting caught, an therefore the resulting protocol is an $\mathbf{AM}[k]$ protocol for $\overline{L}$ by the correctness of the reduction $\overline{R}$. The verifier is able to emulate $\overline{R}$ by himself except for the oracle queries. Thus it suffices to exhibit a constant-round public-coin protocol such that an honest prover produces a response similar to Sam's response, and a cheating prover is caught with high probability. Of course, queries that the reduction makes in parallel are also run in parallel in the emulation. We call the sub-protocol $\mathsf{AM\text{-}Sam}$, as it emulates exactly the behavior of Sam except the sampling is done using the $\mathsf{SampleWithSize}$ protocol of Theorem 4.1.

Moving to the formal proof, one should first take care of the fact that while R (and thus $\overline{R}$) expects a stateless deterministic oracle, Sam is statefull and randomized. Consider the following variant of Sam (which is in the spirit of the Goldreich-Krawczyk [GK92] cheating verifier's strategy): rather then flipping its random coin uniformly at random, each time Sam needs fresh random coins it draws them from $\pi(q_1, \ldots, q_i)$, where π is a random function (chosen by Sam before the interaction starts), and $q_1, \ldots, q_i$ is the query history (including the current one). It is clear that also this version of Sam breaks the binding of Com with save but negligible probability.

An average argument yields that by fixing of π (used by the above Sam) to all but negligible fraction of possible values, we get a deterministic algorithm that breaks the binding of Com. In particular, for any x it holds that $\overline{R}^{\mathsf{Sam}_\pi}$ decides x correctly for most fixing of π. It follows that $\overline{R}^{\mathsf{Sam}}$ decides x with save but negligible probability, also when we implement Sam as follows: on a query q_i, Sam checks if it was already asked on q_i with the same query history (*i.e.*, same value of $q_1, \ldots, q_{i-1}$). If the answer is positive, Sam answers exactly the same answer it gave on reply to this previous call. Otherwise Sam acts normally, but where each time it needs fresh random coins, it samples them uniformly at random. In the following we address the latter randomized variant of Sam.

It left to give a stateless version of Sam. Following [HHRS07], there exists a stateless variant of Sam such that the following hold. This stateless Sam breaks the binding of Com with save but negligible probability, and the "interaction" of any efficient algorithm with the stateless Sam, can be simulated (with at most negligible statistical difference) given oracle to the statefull Sam. Having the above, we consider an execution of $\overline{R}^{\mathsf{Sam}}$ where Sam is statefull and randomized (in the above meaning — on each query history Sam chooses fresh random coins).

On input x, the verifier emulates an execution of $\overline{R}^{\mathsf{Sam}}(x)$ as follows: the verifier keeps track of a table Prefix (initially empty). During the course of emulating $\overline{R}$, each time the reduction makes a query q_i and the query history to Sam is $q_1, \ldots, q_{i-1}$, the verifier and the receiver interact in $\mathsf{AM\text{-}Sam}(q_1, \ldots, q_{i-1}, q_i)$ below. If $i < c$ and the verifier's output in $\mathsf{AM\text{-}Sam}$ is $((a_1, \ldots, a_i), s)$, then the verifier returns a_i to $\overline{R}$. Where if $i = c$ and the verifier's output is $(a_i, \omega^0, \omega^1)$, then the verifier returns $(a_i, \omega^0, \omega^1)$ to $\overline{R}$. (Note that R can rewind Sam, and in particular make more than c queries).

The protocol AM-Sam below invokes SampleWithSize as a sub-protocol. In the following let $\gamma_i = (\frac{\delta}{100n})^{9(c+1-i)}$ for $\delta = 1/\text{poly}(n)$ to be determined later. Note that it holds tha $(\frac{\lambda_i}{100n})^8 \leq \gamma_{i+1}$.

Protocol 3.2. AM-Sam $= (P, V)$**.**

Input: $q_1, \ldots, q_i$.

Operation:

1. *If* $\mathsf{Prefix}(q_1, \ldots, q_i)$ *is not empty,* V *outputs* $\mathsf{Prefix}(q_1, \ldots, q_i)$*, and halts.*
2. *Let* $(\tau_{i-1}, s_{i-1}) = \mathsf{Prefix}(q_1, \ldots, q_{i-1})$ *and define* $f(\omega) = a$*, where* a *is such that* $\mathsf{Sender}(q_1, \ldots, q_i, \omega, 0) = (a_1, \ldots, a_{i-1}, a)$*.* V *and* P *interact in* SampleWithSize*, on inputs* $S = \mathsf{Sib}(\tau_{i-1}, 0)$*, approximate set size* s_{i-1}*, function* f *and approximation factor* λ_i*, to obtain a sample* $\omega \in S$ *and an approximate size* s_i *for* $|f^{-1}(f(\omega)) \cap S|$*.*

   Let $a_i = f(\omega)$*, and* $\tau_i = (\tau_{i-1}, q_i, a_i)$*.*
3. *If* $i < c$*,* V *sets* $\mathsf{Prefix}(q_1, \ldots, q_i) = (\tau_i, s_i)$*, output* a_i *and halts.*

   Otherwise, V *and* P *interact in* SampleWithSize*, on inputs* $S = \mathsf{Sib}(\tau_i, 1)$*, approximation set size* s_i*, the trivial function* $f(\omega) = 0$*, and approximation factor* λ_{i+1} *to obtain a sample* $\omega' \in S$*.* V *sets* $\mathsf{Prefix}(q_1, \ldots, q_i) = (a_i, w, w')$ *and outputs* (a_i, w, w')*.*

If AM-Sam aborts, then the verifier rejects. We call $(\tau_i, \tilde{s}_i)$ a ***good pair*** if $\tilde{s}_i \in [(1 \pm \gamma_i)|\mathsf{Sib}(\tau_i, 0)|]$. The following lemma immediately follows from Theorem 4.1.

Lemma 3.3. *For every query* $q_1, \ldots, q_i$*, if* $(\tau_{i-1}, \tilde{s}_{i-1}) = \mathsf{Prefix}(q_1, \ldots, q_{i-1})$ *form a good pair, then the following hold.*

1. *When* V *interacts in AM-Sam with the honest prover, it aborts with probability at most* 2δ*.*
2. *When* V *interacts in AM-Sam with an arbitrary prover, either* V *rejects with probability* $\geq 1 - 2\delta$*, or conditioned on not aborting,* V *finds (in Step* 2*) a good pair* $(\tau_i, \tilde{s}_i)$ *with probability* $\geq 1 - \delta$*, and* $(a_1, \ldots, a_i)$ *is* γ_i*-close to the output distribution of* Sam *(for the same query history, over its random coins). If* $i = c$*, then* $(a_1, \ldots, a_c, \omega_c, \omega'_c)$ *is* 2δ*-close to the output distribution of* Sam*.*

Suppose that $\overline{R}$ makes at most a polynomial $p(n)$ number of queries. By setting $\delta \leq 1/(10np(n))$, all the verifier in all the executions of AM-Sam done through the emulation runs in polynomial time. Lemma Lemma 3.3 yields that the honest prover makes the verifier abort with probability $\leq 2\delta p(n) = 1/(5n)$. On the other hand, for any prover either we reject with probability $> 1 - 2\delta$ in some invocation of AM-Sam, or conditioned on not rejecting in any invocation of AM-Sam the probability that we ever produce a bad pair $(\tau_i, \tilde{s}_i)$ is at most $p(n)2\delta = 1/(5n)$. Lemma Lemma 3.3, yields that conditioned on not having any bad pairs, it holds that all the outputs of AM-Sam are distributed 2δ-close to the distribution of Sam's responses (induced by its random coins). Notice that the above conditioning is only over the randomness used in AM-Sam, and not the random coins of $\overline{R}$. Therefore, by the triangle inequality we have that the output distribution of the protocol differs from the output distribution of the reduction given an oracle Sam by at most $p(n)2\delta = 1/(5n)$. Hence, the completeness and soundness errors of the protocol are bounded by $1/n$ plus the error of $\overline{R}$. ■

4 The Size-estimating Protocol

In this section we prove our main technical lemma.

Theorem 4.1 (restating Theorem 1.3)**.** *There exists a* **AM** *protocol (*SampleWithSize*) that gets as input a security parameter* 1^n*,* $\delta \geq 1/\text{poly}(n)$*, an efficiently verifiable set* $S \subset \{0,1\}^n$*,* $s \in \mathbb{N}$ *(as size of* S*) and an efficiently computable function* $f : \{0,1\}^n \to \{0,1\}^*$*,*[11] *such that the following holds. If* $s \in [(1 \pm (\frac{\delta}{100n})^8) \cdot |S|]$*, then:*

1. *For any (unbounded) prover* P *either* V *aborts or outputs a pair* (x, s_x)*. Moreover, either:*
   - V *aborts with probability at least* $1 - \delta$*, or*

[11] S, f both given as circuits.

- *conditioned on not aborting, x is distributed δ-close to uniform over S and with probability at least $1-\delta$, $s_x \in [(1 \pm \delta)|f^{-1}(f(x)) \cap S|]$.*

2. *There is a prover strategy (the honest prover) such that V aborts with probability at most δ.*

We first explain the high level ideas behind the protocol of Theorem 4.1, then we will give the formal protocol/proof for a simplified case which carries the main technical ideas of the general case, and finally we will show how to extend the proof of the simplified case to the general case.

4.1 Intuition behind the proof: Using the histogram

We give the formal definition of the histogram of f. For some $\epsilon = 1/\text{poly}(n)$, let the i^{th} "bin" be the set $B_i = \{y \mid \Pr_{x \leftarrow_R S}[f(x) = y] \in ((1+\epsilon)^{-i}, (1+\epsilon)^{1-i}]\}$, namely all y such that the probability that $f(x) = y$ is roughly $(1+\epsilon)^{-i}$. Then, the i'th entry of the histogram is $b_i = \Pr_x[f(x) \in B_i]$. Let $\mathsf{bin}(x) = i$ whenever $f(x) \in B_i$. Notice that the small bins (*i.e.* small i) are "heavier", namely for smaller i, the elements $y \in B_i$ occur with greater probability. By definition, $f(x) \in B_i$ yields that $|f^{-1}(f(x)) \cap S| \in (|S|(1+\epsilon)^{-i}, |S|(1+\epsilon)^{1-i}]$. Also note that $\sum_j b_j = 1$, that $\sum_j b_j(1+\epsilon)^j$ is an approximation of the size of the image of f, and that $(1+\epsilon)^m > |S|$ yields that $B_j = \emptyset$ (*i.e.*, $b_j = 0$) for $j > m$.

The vector $b = (b_1, \ldots, b_m)$ defined above is called the ϵ-histogram of the function f over S, and it encodes the (approximate) regularity structure of the function f over the domain S. For example if $\epsilon = 1$ and $|S| = 2^{m-1}$, then a 1-to-1 function's histogram has one non-zero entry $b_m = 1$, a 2-to-1 function's histogram has one non-zero entry $b_{m-1} = 1$, while a constant function's histogram has one non-zero entry $b_1 = 1$. Functions with more complex regularity structures would have more complex histograms. If $(1+\epsilon_1)^t = (1+\epsilon_2)$ for $t \in \mathbb{N}$, the ϵ_2-histogram of f can be retrieved from the ϵ_1-histogram of f by simply merging the adjacent bins: $(b_1 + \cdots + b_t, \ldots, b_{m-t+1} + \cdots + b_m)$. We call this "scaling up" the histogram.

Given an approximation of $|S|$, we can use the GVW protocol (Lemma 2.2) to sample an almost uniform $x \in S$. The challenge, however, is to obtain also an estimate for $\left|f^{-1}(f(x)) \cap S\right|$. In the following we show that a good approximation of the histogram, can be used to sample (x, s_x) as promised by Theorem 4.1. First, we want the verifier in $\mathsf{SampleWithSize}$ to obtain a good approximation of the histogram of f using the prover's help, but we must prevent the cheating prover from giving us a bad histogram. Intuitively, there are two main ways the prover can cheat: he can move weight in the histogram from smaller bins to larger ones or vice versa. We will employ the GS public-coin lower-bound protocol in two different ways to prevent both kinds of cheating.

Our **Image Test** prevents shifting weight from larger bins to the smaller ones. The verifier use the GS protocol to lower-bound the size of the *image* of f: if the prover shifted too much weight from larger bins to smaller ones, this means he claims there are too many $y \in f(S)$ that have a small number of pre-images. Recall that the histogram gives a claim about the size of the image of f. Thus, claiming that too many $y \in f(S)$ have a small number of pre-images, yields a claim that the image of $f(S)$ is much larger than it actually is. In this case, the verifier can falsify such a claim by running the GS protocol on $f(S)$.

Our **Preimage Test** prevents shifting weight from smaller bins to larger ones. The verifier samples many instances $x_1, \ldots, x_\ell$ (for $\ell = \text{poly}(n)$) from S almost uniformly at random (using the approximation $s \approx |S|$ and the GVW protocol (Lemma 2.2)), and asks the prover for their bin numbers $\mathsf{bin}(x_i)$. The fraction of the times that the prover answers i should converge to b'_i (and the verifier checks that). In addition, the verifier prevents the prover from giving bin numbers that are higher than the true values (i.e., $\mathsf{bin}'(x) > \mathsf{bin}(x)$), by using the GS lower-bound protocol (Lemma 2.1) over $f^{-1}(f(x_i)) \cap S$.

These two tests together force the prover to tell the true value $\mathsf{bin}'(x) = \mathsf{bin}(x)$ for almost all $x \in \{x_1, \ldots, x_\ell\}$. If the prover passes both tests, the verifier chooses uniformly at random $x \in \{x_1, \ldots, x_\ell\}$ (where $\{x_1, \ldots, x_\ell\}$ is the set generated in the Preimage Test), and outputs $(x, s(1+\epsilon)^{1-\mathsf{bin}'(x)})$. It follows that with high probability, x is a uniform sample and $|f^{-1}(f(x)) \cap S| \approx s(1+\epsilon)^{1-\mathsf{bin}'(x)}$.

Actually proving that these two tests guarantee soundness is more involved, as we need to argue that these two tests suffice to catch the entire range of cheating prover strategies between the two simple extremes of shifting smaller bins to larger ones and vice versa. It turns out that the above verifier cannot ensure that b' is a good approximation of the histogram. We will show, however, that if b' is not a good approximation, either the verifier catches this cheating and aborts, or else the bad ϵ-histogram b' scales up to a good $t\epsilon$-histogram $b'_t = (b'_1 + \cdots + b'_t, \ldots, b'_{m-t+1} + \cdots + b'_m)$ that is a good approximation for the true $t\epsilon$-histogram $b_t = (b_1 + \cdots + b_t, \ldots, b_{m-t+1} + \cdots + b_m)$. By taking $\epsilon' = t\epsilon$ small enough, we can use the ϵ'-histogram (the way explained above) to do the sampling task of Theorem 4.1.

4.2 The Case of "Exact Regularities"

In this section we consider a simplified case with the following properties.

(1) Exact Regularities: For all $y \in f(\{0,1\}^n)$ it holds that $\Pr_x[f(x) = y] = (1+\epsilon)^{-j}$ for some $j \in \mathbb{Z}$.

(2) Examples Set: For $\ell = \text{poly}(n)$ the verifier has access to a "well-distributed" set $X = \{x_1, \ldots, x_\ell\}$ where each x_i is distributed uniformly over S for all $1 \leq i \leq \ell$, and in addition $\frac{|X \cap f^{-1}(B_j)|}{\ell} = b_j$ holds for all $1 \leq j \leq m$. (Note x_i's are not necessarily independent.)

(3) Exact Size: $s = |S|$.

For the above simplified case, we give the following implementation of SampleWithSize.

Protocol 4.2 (protocol for exact regularities)**.**
SampleWithSize $= (P, V)$**.**

Parameters: 1^n, $\delta = 1/\text{poly}(n)$, *granularity parameter* $m = 100n^2/\delta^2$ *and* $\epsilon >$ *such that* $(1+\epsilon)^{m-1} = 2^n$.

Common input: $S \subseteq \{0,1\}^n$, *a function* $f : \{0,1\}^n \to \{0,1\}^*$, *size estimate* s *and an example set* $\{x_1, \ldots, x_\ell\}$.

Description:

1. V *asks* P *for* $\mathsf{bin}(x_i) \in [m]$ *for each* $1 \leq i \leq \ell$. *Let* $\mathsf{bin}'(x_i)$ *be* P*'s responses.*
2. ***Preimage Test.*** *For all* $1 \leq i \leq \ell$, V *and* P *interact (in parallel) the GS lower-bound protocol (Lemma 2.1) over the set* $f^{-1}(f(x_i))$ *and set size* $s(1+\epsilon)^{1-\mathsf{bin}'(i)}$ *with* $\epsilon/3$ *and* $\frac{\delta}{100\ell}$ *as the approximation and the soundness parameters. Let* $b'_i = \frac{|\{j | \mathsf{bin}'(x_j) = i\}|}{\ell}$ *be* V*'s estimate of* b_i *it obtained in the above interaction.*
3. ***Image Test.*** *Let* $B_{\leq i} = B_1 \cup \ldots \cup B_i$. V *and* P *interact in (the extended version of) the GS lower-bound protocol (Lemma 2.1) over the set* $B_{\leq i}$, *with* $\sum_{1 \leq i \leq j} b'_i(1+\epsilon)^{i-1}$ *as* V*'s estimate for* $|B_{\leq i}|$ *and* $\epsilon/3, \frac{\delta}{100m}$ *as the approximation and soundness parameters.*[12]
4. V *picks uniformly at random* $i \in [\ell]$, *and outputs* $(x = x_i, s_x = s(1+\epsilon)^{1-\mathsf{bin}'(x_i)})$.

Notice that in step 3, the claim membership in $B_{\leq i}$ is verifiable using an **AM** protocol, and the reason is as follows. The exact regularities condition yields that for every i it holds that $|B_i| = \frac{b_i}{(1+\epsilon)^{1-i}}$, and for every $y \in B_{\leq i}$ and any $y' \in f(\{0,1\}^n) \setminus B_{\leq i}$, it holds that $(1+\epsilon)|f^{-1}(y') \cap S| \leq s(1+\epsilon)^{1-i} \leq |f^{-1}(y) \cap S|$. Hence, the lower-bound protocol (Lemma 2.1) used over the set $f^{-1}(y) \cap S$, set size $(1+\epsilon)^{1-i}$, and approximation and soundness parameters ϵ and 2^{-n}, gives an **AM** protocol for deciding membership in $B_{\leq i}$.

In the rest of this section we prove that Protocol 4.2 has the properties needed for Theorem 4.1 for the simple case of exact regularities. It is clear from the protocol that if the prover is honest, the verifier rejects with probability at most $\ell\frac{\delta}{100\ell} + m\frac{\delta}{100m} < \delta$, and otherwise the output of the protocol is (x, s_x) where x is a uniform member of S and s_x is equal to $|f^{-1}(f(x)) \cap S|$.

Claim 4.3. *Either the verifier rejects with probability at least* $1 - \delta/50$ *or both of the following holds:*

- ***One Directional Shifts:*** *For all* $1 \leq i \leq \ell$, $\mathsf{bin}(x_i) \leq \mathsf{bin}'(x_i)$.
- ***Exponential Sum (ES) Inequalities:*** *For all* $1 \leq j \leq m$, $\sum_{1 \leq i \leq j} b'_i(1+\epsilon)^i \leq (1+\epsilon/3)\sum_{1 \leq i \leq j} b_i(1+\epsilon)^i$.

Proof. Assuming that the prover gives $\mathsf{bin}'(x_i) < \mathsf{bin}(x_i)$ for any i, then the exact regularities condition yields that $(1+\epsilon)|f^{-1}(f(x_i)) \cap S| \leq s(1+\epsilon)^{1-\mathsf{bin}'(x_i)}$, and therefore the verifier aborts in some execution of the GS lower-bound protocol in the Preimage Test (Step 2) with probability at least $1 - \frac{\delta}{100\ell}$. Similarly, if for any j, it holds that $(1+\epsilon/3)|B_{\leq i}| \leq \sum_{1 \leq i \leq j} b'_i(1+\epsilon)^{i-1}$, then the verifier rejects in some execution of the GS lower-bound protocol in the Image Test (Step 3) with probability at least $1 - \frac{\delta}{100\ell}$.

By the union bound, either the verifier rejects with probability at least $1 - \ell\frac{\delta}{100\ell} - m\frac{\delta}{100m} = 1 - \delta/50$ or the conclusion of the lemma holds. ∎

In the following we assume that the inequalities of Claim 4.3 hold.

[12] Note that membership in $B_{\leq j}$ is verifiable using an **AM** protocol

Definition 4.4. For any two vectors $b = (b_1, \ldots, b_m)$ and $b' = (b'_1, \ldots, b'_m)$ let $\Delta(b, b')$ be a vector of the same dimension m such that its j^{th} component is $\Delta(b, b')_j = \sum_{1 \le i \le j} (b_i - b'_i)$.

When b, b' are two histograms, $\Delta(b, b')_j$ is exactly the amount of weight must be pushed from bins $\le j$ to bins $> j$ in order to transform b into b'. Since by Claim 4.3 $\mathsf{bin}(x_i) \le \mathsf{bin}'(x_i)$ holds for all $1 \le i \le \ell$, therefore for all $1 \le j \le m$,

$$\Delta(b, b')_j = \sum_{1 \le i \le j} (b_i - b'_i) = \frac{\{k \in [\ell] \mid \mathsf{bin}(x_k) \le j, \mathsf{bin}'(x_k) > j\}}{\ell} \ge 0$$

If we sum up over all j, we get

$$\ell \sum_{1 \le j \le m} \Delta(b, b')_j = \sum_{1 \le j \le m} |\{i \mid \mathsf{bin}(x_i) \le j, \mathsf{bin}'(x_i) > j\}| = \sum_{1 \le i \le \ell} \mathsf{bin}'(x_i) - \mathsf{bin}(x_i) \ , \tag{4.1}$$

where the last inequality holds since each x_i contributes $\mathsf{bin}'(x_i) - \mathsf{bin}(x_i)$ to the summation.

Intuitively, $\sum_j \Delta(b, b')_j$ captures the total "work" that must be done to transform b into b'. Namely, thinking of each item x_i as having unit mass, $\ell\Delta(b, b')$ sums up the distance that each of these unit masses must travel in order to get from its bin in b to its bin in b'. Of course, if $b = b'$ then no work is necessary. If we could upper bound the amount of work done by the cheating prover to shift b to b', we could also bound how different b, b' are. It turns out that if the amount of work done is sufficiently small, then b' is a good approximation (at least when scaled up) of b for our purposes. This intuition is formalized in the following lemma.

Lemma 4.5. *Let $b = (b_1, \ldots, b_m)$ and $b' = (b'_1, \ldots, b'_m)$ be probability distributions (i.e., nonnegative entries and sum equal to one). Assuming that the following hold for $100 \le 1/\epsilon \le m$:*

1. *$\Delta(b, b')_j \ge 0$ for all $1 \le j \le m$ and*
2. *ES inequalities: $\sum_{1 \le i \le j} b'_i (1+\epsilon)^i \le (1+\lambda) \sum_{1 \le i \le j} b_i (1+\epsilon)^i$ for all $1 \le j \le m$,*

then $\sum_j \Delta(b, b')_j \le 8m\lambda$.

Before proving Lemma 4.5, let see how it can be used to prove the correctness of the protocol. Claim 4.3 implies the hypotheses of Lemma 4.5 for $\lambda = \epsilon/3 < 2\epsilon$, by which we get $\sum_j \Delta(b, b')_j \le 16m\epsilon < 16n$.[13]

Let $q_t = |\{i \mid \mathsf{bin}'(x_i) - \mathsf{bin}(x_i) \ge t\}|/\ell$ denote the fraction of instances that the prover cheated on their bin number by at least t bin differences. Since $\sum_{1 \le i \le \ell} \mathsf{bin}'(x_i) - \mathsf{bin}(x_i) \ge \ell t q_t$, Lemma 4.5 and Equation 4.1 imply that $16n \ge \sum_{1 \le j \le 2m} \Delta(b, b')_j \ge t q_t$. Therefore, $q_t \le \frac{16n}{t}$. If $t = 16n/\delta$, then for at least $1 - q_t > 1 - \delta$ fraction of x_i's (for $1 \le i \le \ell$), the prover has not cheated on their bin number by more than t bin differences. By choosing uniformly at random $i \in [\ell]$, with probability at least $1 - \delta$ the verifier gets x_i such that the prover has given her $|f^{-1}(f(x_i)) \cap S|$ with approximation factor $(1+\epsilon)^t$ (which is $(1+\epsilon)^t < (1+\delta)$ for any $\epsilon < \frac{\delta}{2t} = \frac{\delta^2}{32n}$). Since we set the parameters as $(1+\epsilon)^{m-1} \le 2^n$, it follows that $\epsilon m < 2n$, and so $\epsilon < \frac{n}{m} <= \frac{\delta^2}{100n} < \frac{\delta^2}{32n}$.

Proof of Lemma 4.5. We artificially increase the dimension of b, b' to $2m$ by padding m zero coordinates to all of them (e.g. $b = (b_1, \ldots, b_m, 0, \ldots, 0) \in \mathbb{R}^{2m}$). Let us denote $a_j = \Delta(b, b')_j$ and $a = (a_1, \ldots, a_{2m})$.

Elaborating on our interpretation of $a = \Delta(b, b')$ as measuring how to transform b into b', notice that the following procedure incrementally performs such a transformation in $2m - 1$ steps:

- $b^1 = b$.
- For $1 \le j < 2m$, b^{j+1} is the same vector as b^j, with the only difference that it takes the amount of a_j from its j^{th} coordinate, and adds it to its $(j+1)^{\text{th}}$ coordinate: $b^{j+1}_j = b^j_j - a_j, b^{j+1}_{j+1} = b^j_{j+1} + a_j$.

At the end we would get $b^{2m} = b'$. In other words, in the j^{th} step the amount of $a_j = \sum_{1 \le i \le j} b_i - b'_i$ is pushed out from the first j coordinates. Also, for any $k \ge j+1$, the exponential sum $\sum_{1 \le i \le k} b^{j+1}_i (1+\epsilon)^i$ for the vector b^{j+1} is bigger than the exponential sum $\sum_{1 \le i \le k} b^j_i (1+\epsilon)^i$ for the vector b^j by the amount of $a_j(1+\epsilon)^{j+1} - a_j(1+\epsilon)^j = \epsilon a_j (1+\epsilon)^j$.

Now for $1 \le j \le 2m$, let $A_j = \sum_{1 \le i \le j-1} a_i \epsilon (1+\epsilon)^i$ be the net effect of the first $j-1$ steps of the above transformation on the Exponential Sums of length at least j (and in particular $A_1 = 0$). The next lemma compares the exponential sums of b and b' in terms of A_j.

[13] We can use the smaller value of $\lambda = \epsilon/3$ at this moment, but as we will see in Section 4.3, in the general case we can only get the ES inequalities for $\lambda > (1 + \Omega(1))\epsilon$.

Claim 4.6. *It holds that* $\sum_{1\leq i\leq j}(b'_i - b_i)(1+\epsilon)^i = A_j - a_j(1+\epsilon)^j$.

Proof. Note that

$$\begin{aligned} A_j - a_j(1+\epsilon)^j &= \sum_{1\leq i\leq j-1} a_i\epsilon(1+\epsilon)^i - a_j(1+\epsilon)^j \\ &= \sum_{1\leq i\leq j-1} \epsilon(1+\epsilon)^i \sum_{1\leq k\leq i}(b_k - b'_k) - (1+\epsilon)^j \sum_{1\leq k\leq j}(b_k - b'_k) \\ &= (b'_j - b_j)(1+e)^j + \sum_{1\leq i\leq j-1}((b'_i - b_i)((1+\epsilon)^j - (\epsilon \sum_{i\leq k\leq j-1}(1+\epsilon)^k))) \ . \end{aligned}$$

On the other hand $\sum_{i\leq k\leq j-1}(1+\epsilon)^k = \frac{(1+\epsilon)^j-(1+\epsilon)^i}{\epsilon}$ implies that:

$$A_j - a_j(1+\epsilon)^j = (b'_j - b_j)(1+e)^j + \sum_{1\leq i\leq j-1}(b'_i - b_i)(1+\epsilon)^i = \sum_{1\leq i\leq j}(b'_i - b_i)(1+\epsilon)^i \ .$$

■

The next claim shows that the normalized $\frac{A_j}{(1+\epsilon)^j}$ is always at most 2λ. This matches our intuition, since, as Claim 4.6 showed, $A_j(1+\epsilon)^{-j}$ is an approximation of the gap between the j-length exponential sums of b and b', and the ES inequalities say that this gap should be roughly bounded by $O(\lambda)$. We will then use this bound later to get a bound on $\sum_j a_j$.

Claim 4.7. *For every* $1 \leq j \leq 2m$ *we have* $A_j < 2\lambda(1+\epsilon)^j$.

Proof. Let $p_j = \frac{A_j}{(1+\epsilon)^j}$. For $1 \leq j \leq 2m$, let $c_j = \frac{\sum_{1\leq i\leq j} b_i(1+\epsilon)^i}{(1+\epsilon)^j}$ be the normalized form of the exponential sum $\sum_{1\leq i\leq j} b_i(1+\epsilon)^i$. Notice that $c_j \leq 1$ for all j.

We first show that for every $1 \leq j \leq 2m$:

1. $a_j \geq p_j - \lambda c_j$.
2. $p_{j+1} \geq p_j - \frac{\lambda\epsilon c_j}{1+\epsilon}$.

The idea is that on the one hand we know that $a_j = 0$ for all $j > m$ which, by the definition of p_j and A_j, implies that p_{2m} must be small. On the other hand, $p_{j+1} - p_j$ cannot be too large, since otherwise this would open up a large gap between the j-length and $j+1$-length Exponential Sums, contradicting the ES inequalities. These two facts imply that p_j must be small for all j.

More formally, the ES inequalities immediately imply $\sum_{1\leq i\leq j}(b'_i - b_i)(1+\epsilon)^i \leq \lambda\sum_{1\leq i\leq j} b_i(1+\epsilon)^i = \lambda c_j(1+\epsilon)^j$. On the other hand, Claim 4.6 yields that $\sum_{1\leq i\leq j}(b'_i - b_i)(1+\epsilon)^i = A_j - a_j(1+\epsilon)^j$. Therefore, $p_j(1+\epsilon)^j - a_j(1+\epsilon)^j \leq \lambda c_j(1+\epsilon)^j$ and thus $a_j \geq p_j - \lambda c_j$. In addition, since

$$\begin{aligned} p_{j+1}(1+\epsilon)^{j+1} = A_{j+1} &= A_j + a_j\epsilon(1+\epsilon)^j \\ &\geq p_j(1+\epsilon)^j + (p_j - \lambda c_j)\epsilon(1+\epsilon)^j \\ &= p_j(1+\epsilon)^j(1+\epsilon) - \lambda c_j\epsilon(1+\epsilon)^j \ , \end{aligned}$$

it follows that $p_{j+1} \geq p_j - \frac{\lambda\epsilon c_j}{1+\epsilon}$. Using induction one concludes that $p_k \geq p_j - \frac{\lambda\epsilon}{1+\epsilon}\sum_{j\leq i<k} c_j$. Using $a_{2m} = 0$ and $c_{2m} \leq 1$ we get:

$$0 \geq p_{2m} - \lambda c_{2m} \geq (p_j - \frac{\lambda\epsilon}{1+\epsilon}\sum_{j\leq i<2m} c_j) - \lambda \ . \tag{4.2}$$

Notice that $\sum_{1\leq j\leq 2m} c_j < \frac{1+\epsilon}{\epsilon}$, which follows by simply re-ordering of summations:

$$\sum_{1\leq j\leq 2m} c_j = \sum_{1\leq j\leq 2m} \frac{\sum_{1\leq i\leq j} b_i(1+\epsilon)^i}{(1+\epsilon)^j} = \sum_{1\leq i\leq 2m}\sum_{i\leq j\leq 2m} b_i(1+\epsilon)^{i-j} = \sum_{1\leq i\leq 2m} b_i \sum_{0\leq k\leq 2m-i}(1+\epsilon)^{-k} < \sum_{1\leq i\leq 2m} b_i \sum_{0\leq k\leq\infty}(1+\epsilon)^{-k}.$$

Applying the geometric sum formula $\sum_{0\leq k\leq\infty}(1+\epsilon)^{-k}=\frac{1+\epsilon}{\epsilon}$, yields that $\sum_{1\leq j\leq 2m}c_j<\frac{1+\epsilon}{\epsilon}\sum_j b_j=\frac{1+\epsilon}{\epsilon}$. Hence, by Inequality 4.2 we have that

$$0\geq p_j-\frac{\lambda\epsilon}{1+\epsilon}\sum_{j\leq i<2m}c_j-\lambda\geq p_j-\frac{\lambda\epsilon}{1+\epsilon}(\frac{1+\epsilon}{\epsilon})-\lambda=p_j-2\lambda\ ,$$

and therefore $p_j\leq 2\lambda$. ■

We derive Lemma 4.5 from the above Claim 4.7 as follows. Note that

$$2m\cdot 2\lambda\geq\sum_{1\leq j\leq 2m}\frac{A_j}{(1+\epsilon)^j}=\sum_{1\leq j\leq 2m}\sum_{1\leq i<j}\epsilon a_i(1+\epsilon)^{i-j}=\epsilon\sum_{1\leq i<2m}a_i\sum_{i<j\leq 2m}(1+\epsilon)^{i-j}=\epsilon\sum_{1\leq i<2m}a_i\sum_{1\leq k\leq 2m-i}(1+\epsilon)^{-k}\quad .$$

For $i>m$ it holds that $a_i=0$. On the other hand since $100\leq 1/\epsilon\leq m$, then $(1+\epsilon)^{-m}<1/2$ and therefore for $i\leq m$ it holds that

$$\sum_{1\leq k\leq 2m-i}(1+\epsilon)^{-k}\geq\sum_{1\leq k\leq m}(1+\epsilon)^{-k}=\frac{1-(1+\epsilon)^{-m}}{\epsilon}>\frac{1}{2\epsilon}\quad .$$

Since $a_i\geq 0$, it follows that $a_i\sum_{1\leq k\leq 2m-i}(1+\epsilon)^{-k}>\frac{a_i}{2\epsilon}$ and so $4m\lambda\geq\frac{\epsilon}{2\epsilon}\sum_j a_j$, which means $\sum_j a_j\leq 8\lambda m$. ■

4.3 The General Case

In this section we extend the protocol for the special case of exact regularities to the general conditions stated in Theorem 4.1 by the following steps.

Exact regularity to a regularity gap condition. First we weaken the regularity condition to just keep a multiplicative gap between the probabilities in different bins: for every i and every $y\in B_i$, it holds that $\Pr_x[f(x)=y]\in[(1+\epsilon)^{1/2-i},(1+\epsilon)^{1-i}]$ (where $B_i=\{y\in f(S)\mid\Pr_{x\leftarrow S}[f(x)=y]\in((1+\epsilon)^{-i},(1+\epsilon)^{1-i}]\}$). The gap condition yields that the verifier can still use the lower-bound protocol to make sure that $\mathsf{bin}'(x_i)\geq\mathsf{bin}(x_i)$ for $1\leq i\leq\ell$. The problem is that the verifier can no longer conclude the ES inequalities of Claim 4.3. The reason being that the size of B_i is not determined by b_i exactly, but rather describes it approximately: $\frac{b_i}{(1+\epsilon)^{1-i}}\leq|B_i|\leq\frac{b_i}{(1+\epsilon)^{1/2-i}}$. Using the gaps between the probabilities in different bins, the verifier can still verify membership in $B_{\leq i}$ through an **AM** protocol. So it can still use the (extended version of the) lower-bound protocol on the size of B_i using her *minimum* estimate for it. The inequalities the verifier concludes after running the protocol (in case of not rejecting with probability at least $1-\delta$) are: $\sum_{1\leq i\leq j}b'_i(1+\epsilon)^{i-1}\leq(1+\epsilon/3)\sum_{1\leq i\leq j}b_i(1+\epsilon)^{i-1/2}$ for all j. The reason is that, the prover will be caught (with high probability), if the minimum estimate of $B_{\leq i}$ (that the verifier concludes from prover's claims) is $(1+\epsilon/3)$-larger than the maximum possible value of $B_{\leq i}$. Therefore:

$$\sum_{1\leq i\leq j}b'_i(1+\epsilon)^i\leq(1+\epsilon/3)(1+\epsilon)^{1/2}\sum_{1\leq i\leq j}b_i(1+\epsilon)^i<(1+2\epsilon)\sum_{1\leq i\leq j}b_i(1+\epsilon)^i$$

for all $1\leq j\leq 2m$. In other words, the verifier still gets the ES inequalities of Lemma 4.5 for $\lambda=2\epsilon$, and the conclusions we had before still hold.

Sampling examples set. In this step, instead of having the set of examples X, we only assume that there is an efficient algorithm to sample uniformly from S. In this case, the verifier asks the prover to give her the vector b. Let b' be the prover's claim about b. In addition, the verifier samples $x_1,\ldots,x_\ell$ for $\ell=m^2\log(\frac{100m}{\delta})$ uniformly and independently at random from S, and asks the prover to give her the bin numbers of x_i's. Let $\mathsf{bin}'(x_i)$ be the prover's claim for $\mathsf{bin}(x_i)$. The verifier uses $x_1,\ldots,x_\ell$ and b' as before and runs the protocol.

If we had $a_j=\Delta(b,b')_j\geq 0$ for $1\leq j\leq 2m$, then Lemma 4.5 would still hold. While the latter is not necessarily true, we do know that for large enough ℓ, the fraction of times that the verifier samples from bin i should be near b'_i (as otherwise the verifier can safely reject). As we will see this makes a_j's to behave good enough for our purpose. More formally let $u=(u_1,\ldots u_{2m})$ be such that $u_j=\frac{|\{i|\mathsf{bin}(x_i)=j\}|}{\ell}$ and $u'=(u'_1,\ldots,u'_{2m})$ be such that $u'_j=\frac{|\{i|\mathsf{bin}'(x_i)=j\}|}{\ell}$. For large enough ℓ, the vector u converges to the vector b. In particular, for any fixed j, using Hoeffding's inequality we have $\Pr[|\Delta(b,u)_j|\geq 1/m]<2e^{-\ell/m^2}<\frac{\delta}{100m}$, and so by the union bound, $|\Delta(b,u)_j|<1/m$ holds for all $1\leq j\leq 2m$

with probability at least $1-\delta/50$. This means that the verifier can safely check that $|\Delta(b',u')_j| < 1/m$ holds for all $1 \le j \le 2m$, and otherwise reject. (If the prover is honest, the verifier rejects at this point only with probability at most $\delta/50$.)

Now note that $(a_1,\ldots,a_{2m}) = \Delta(b,b') = \Delta(b,u)+\Delta(u,u')+\Delta(u',b')$. What we really need to prove (rather than the conclusion of Lemma 4.5) is to show that $\sum_j \Delta(u,u') \le O(n)$, where the latter can be used (similarly to the way Lemma 4.5 was used) for bounding the prover's ability to cheat on the bin numbers.

Since the verifier runs the lower-bound protocol on the sets $f^{-1}(f(x_i)) \cap S$, it holds that $\Delta(u,u')_j \ge 0$ for all $1 \le j \le 2m$. In addition we just showed that $|\Delta(b,u)_j + \Delta(u',b')_j| \le |\Delta(b,u)_j| + |\Delta(u',b')_j| \le 2/m$ for all $1 \le j \le 2m$. The last line of the proof of Lemma 4.5 was the only place that we used $a_j \ge 0$. There we showed that:

$$2m \cdot 2\lambda \ge \epsilon \sum_{1\le i<2m} a_i \sum_{1\le k\le 2m-i} (1+\epsilon)^{-k} = \epsilon \sum_{1\le i<2m} (\Delta(u,u')_j + \Delta(b,u)_j + \Delta(u',b')_j) \sum_{1\le k\le 2m-i} (1+\epsilon)^{-k} \ .$$

Since $\epsilon \sum_{1\le i<2m} \Delta(u,u')_j \sum_{1\le k\le 2m-i}(1+\epsilon)^{-k} \ge \frac{\epsilon}{2\epsilon}\sum_j \Delta(u,u')_j$ and $\epsilon \sum_{1\le i<2m}(\Delta(b,u)_j+\Delta(u',b')_j)\sum_{1\le k\le 2m-i}(1+\epsilon)^{-k} > \epsilon(2m)(\frac{-2}{m})(\frac{1}{\epsilon})$, it follows that $4m\lambda \ge \frac{\epsilon}{2\epsilon}\sum_j \Delta(u,u')_j - 4$. Hence, $\sum_j \Delta(u,u')_j < 8(m\lambda+1)$ which for $\lambda < 2\epsilon$ is still $O(n)$.

Using approximation for $|S|$. In this step, we remove the assumption that we know $s = |S|$ exactly, but we assume that we know $s \in [(1\pm\epsilon/20)|S|]$. Let x and y be such that $f(x) \in \mathsf{bin}(i)$ and $f(y) \in \mathsf{bin}(j)$ for $i < j$. Before this step we had $|f^{-1}(f(y)) \cap S| \le s(1+\epsilon)^{1-j}$ and $s(1+\epsilon)^{1/2-i} \le |f^{-1}(f(x)) \cap S|$, Therefore if the prover claims $\mathsf{bin}(x) = j$, he would be caught with high probability if the verifier runs the lower-bound protocol with $s(1+\epsilon)^{1/2-i}$ as the minimum possible size of $|f^{-1}(f(x)) \cap S|$ because $(1+\epsilon/3)|f^{-1}(f(y)) \cap S| \le s(1+\epsilon)^{1/2-i}$. Now s is not exactly equal to $|S|$, but rather $s \in [(1\pm\epsilon/20)|S|]$ which implies that $(1-\epsilon/20)s(1+\epsilon)^{1/2-i} \le |f^{-1}(f(x)) \cap S|$, and therefore the verifier can use $(1-\epsilon/20)s(1+\epsilon)^{1/2-i}$ as her minimum estimate for $|f^{-1}(f(x)) \cap S|$ to run the lower-bound protocol. As we will see in a moment this minimum estimate is still bigger than the maximum possible value of $|f^{-1}(f(y)) \cap S|$ by a factor of $(1+\epsilon)$ and the lower-bound protocols can be used as before. The reason is that now, because of $s \in [(1\pm\epsilon/20)|S|]$ it holds that $|f^{-1}(f(y)) \cap S| \le (1+\epsilon/20)s(1+\epsilon)^{1-j}$, but we have $(1+\epsilon/3)(1+e/20)s(1+\epsilon)^{1-j} < (1-\epsilon/20)s(1+\epsilon)^{1/2-i}$ because $i+1 \le j$.

Therefore by using the lower-bound protocols using the new minimum estimate sizes for the preimage sizes, the verifier can still make sure that $\mathsf{bin}(x_i) \le \mathsf{bin}'(x_i)$ for all i.

Using a similar argument the verifier can also use the lower-bound protocol over the size of $B_{\le i}$ for $1 \le i \le m$, and get all the inequalities of Claim 4.3 (in case she does not reject with probability at least $1-\delta$).

Removing gap condition. In this step we remove the gap condition. Interestingly, the protocol remains exactly the same, where only the analysis gets a bit more involved. The problem is with elements in $f(S)$ whose probabilities lie very close to the borders of the bins. For such element, if the prover claims $\mathsf{bin}'(x_i)$ for $\mathsf{bin}(x_i)$, and the verifier runs the lower-bound protocol (on $|f^{-1}(f(x_i)) \cap S|$) with any approximation factor $\lambda < \epsilon$, the only guarantee that it gets is that $\mathsf{bin}'(x_i) \ge \mathsf{bin}(x_i) - 1$. Also, when the verifier runs the lower-bound protocol on $|B_{\ge j}|$, the prover might use the elements in bin $j+1$ (which are very close to the border with bin j), and the inequalities that the verifier gets are: $\sum_{1\le i\le j} b'_i(1+\epsilon)^i \le (1+2\epsilon)\sum_{1\le i\le j+1} b_i(1+\epsilon)^i$ for all j.

Let $b'' = (b''_1,\ldots,b''_{2m})$ be such that $b''_1 = 0$ and $b''_j = b'_{j-1}$ for $2 \le j \le 2m$, and similarly let $u'' = (u''_1,\ldots,u''_{2m})$ be such that $u''_1 = 0$ and $u''_j = u'_{j-1}$ for $2 \le j \le 2m$. One point is that using b'' (instead of b') we get the ES inequalities of Lemma 4.5:

$$\sum_{1\le i\le j+1} b''_i(1+\epsilon)^i = 0 + \sum_{1\le i\le j} b'_i(1+\epsilon)^i \le (1+2\epsilon) \sum_{1\le i\le j+1} b_i(1+\epsilon)^i \ .$$

Note that $\Delta(u,u'')_j \ge 0$ holds for all j — If one uses $\mathsf{bin}''(x_i) = \mathsf{bin}'(x_i)+1$ instead of $\mathsf{bin}'(x_i)$'s, it would get the vector u'' instead of u', and it holds that that $\mathsf{bin}''(x_i) = \mathsf{bin}'(x_i)+1 \ge \mathsf{bin}(x_i)$.

It also holds that $\Delta(b,b'') = \Delta(b,u)+\Delta(u,u'')+\Delta(u'',b'')$. Since u' is supposed to converge to b' in the limit, u'' also should converges to b'' with the same statistical bounds. The analysis will remain as before with b'' instead of b' and u'' instead of u', and we get $\sum_j \Delta(u,u'')_j = O(n)$. The only difference is that now there is a potential approximation error of $(1+\epsilon)^{-1}$ for the size of the sampled point (as opposed to the previous cases). But this amount of error in the approximation factor is still acceptable.

Using $|S|$ to sample from S. Finally, instead of being given an efficient algorithm to sample from S, we show how to sample from it using the prover's help and a good approximation for $|S|$. Knowing the size of $|S|$ with approximation γ, the verifier can use the Uniform Sampling protocol (Lemma 2.2) to sample x_i's from S (with the help of the prover) with statistical distance at most 5γ from the uniform. Therefore, if the verifier use $\gamma < \delta/(100\ell)$, then the statistical distance of $(x_1, \ldots, x_\ell)$ from the case that they were all sampled uniformly from S is at most $\delta/20$. So all the statistical checks (about entries of $\Delta(u'', b'')$) that the verifier performs on the sequence $(x_1, \ldots, x_\ell)$ (i.e. checking that $|\Delta(u'', b'')_j| < 1/m$ for all j) should still hold with probability at least $1 - \delta/20$. Since $m = \frac{100n^2}{\delta^2}$ and $\ell = m^2 \log(\frac{100m}{\delta})$, hence the value of γ used in the protocol $\gamma = (\frac{\delta}{100n})^8$ is small enough:

$$(\frac{\delta}{100n})^8 < \frac{\delta^2}{100^2(\frac{100n^2}{\delta^2})^3} = \frac{\delta}{100m^2(\frac{100m}{\delta})} < \frac{\delta}{100m^2\log(\frac{100m}{\delta})} = \frac{\delta}{100\ell} \quad .$$

References

[ABI86] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on np-hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 701–710, 2006.

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991. Preliminary version in *FOCS'87*.

[Ajt03] Miklos Ajtai. The worst-case behavior of schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *STOC '03*, pages 396–406, New York, NY, USA, 2003. ACM.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[BKK90] Joan F. Boyar, Stuart A. Kurtz, and Mark W. Krentel. A discrete logarithm implementation of perfect zero-knowledge blobs. *Journal of Cryptology*, 2(2):63–76, 1990.

[BLZ94] J. Buchmann, J. Loho, and J. Zayer. An implementation of the general number field sieve. In *CRYPTO '93*, pages 159–165, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[Bra79] Gilles Brassard. Relativized cryptography. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 383–391. IEEE Computer Society, 1979.

[BT06] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for np problems. *SIAM Journal on Computing*, 36(4):1119–1159, 2006.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DPP98] Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Transactions on Information Theory*, 44(3):1143–1151, 1998.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. pages 10–18, 1984.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.

[GK92] Oded Goldreich and Hugo Krawczyk. Sparse pseudorandom distributions. *Random Structures & Algorithms*, 3(2):163–174, 1992.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989.

[GVW01] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. In *Proc. 28th ICALP*, pages 334–345, 2001.

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2007.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Preliminary versions in *STOC'89* and *STOC'90*.

[HR07] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*. ACM Press, 2007.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of arthur-merlin witness hiding protocols. In *Theory of Cryptography, Fourth Theory of Cryptography Conference, TCC 2009*, 2009.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989.

[LFKN90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *Proc. 31st FOCS*, pages 2–10. IEEE, 1990.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS '04*, pages 372–381, Washington, DC, USA, 2004. IEEE Computer Society.

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO'92*.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.

[Pas06] Rafael Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on np-hardness. In *IEEE Conference on Computational Complexity*, pages 96–110, 2006.

[PSSV07] A. Pavan, Alan L. Selman, Samik Sengupta, and N. V. Vinodchandran. Polylogarithmic-round interactive proofs for conp collapse the exponential hierarchy. *Theor. Comput. Sci.*, 385(1-3):167–178, 2007.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, January 1979.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 387–394, 1990.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb 1978.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.

[Sim98] Daniel Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.

A The case for CRHF

For the case of CRHF, it is convenient to use the following upper-bound protocol instead of using the more complex SampleWithSize protocol.

Lemma A.1 (Upper-bound protocol [AH91])**.** *Let $S \subset \{0,1\}^n$ be a set whose membership can be verified in* $\mathrm{poly}(n)$ *time. Then there exist an* **AM** *protocol with (an efficient) verifier V that gets as input a security parameter 1^n, $\delta = 1/\mathrm{poly}(n)$ (as the soundness parameter), $\epsilon = 1/\mathrm{poly}(n)$ (as the approximation parameter), s (as size of S), and a* secret *random sample $x \leftarrow_R S$ (which the prover does not have access to) such that:*

- *If $|S| \leq s$, then there is a prover that makes V accept with probability at least $1 - \delta$.*
- *If $|S| \geq s(1+\epsilon)$, then for any (unbounded) prover, V rejects with probability at least $\epsilon/5 - \delta$.*

Definition A.2. A *collision resistant hash function* (CRHF) is (a sequence of) families of functions H_n (for $n \in \mathbb{N}$) such that each $h \in H_n$ is a compressing function $f\colon \{0,1\}^n \to \{0,1\}^{n-1}$ and we have:

- **Sampling.** There is a PPT sampler S which $S(1^n)$ samples some $h \in H_n$.
- **Evaluation.** There is a PPT evaluator A such that given h and $x \in \{0,1\}^n$ computes $h(x)$ in $\mathrm{poly}(n)$ time.
- **Collision Resistance (CR).** For any PPT adversary B we have $\Pr[h(x) = h(y) \mid h = S(1^n), B(h) = (x,y)] \leq n^{-\omega(1)}$.

For simplicity, in the subsequent we drop the index n when the context is clear.

Suppose that H has the Sampling and Evaluation properties and we want to reduce its CR property to $\mathbf{NP} \not\subset \mathbf{BPP}$. The following definition captures the black-box type of reductions (which ignore the code of the adversary B).

Definition A.3. A reduction from CR of H to SAT is an PPT R which for any oracle O that $\Pr_{h=S(1^n)}[O(h) = (x,y), x \neq y, h(x) = h(y)] \geq 1/2$ we have $\Pr[R^O(\phi) = \mathsf{SAT}(\phi)] \geq 2/3$ for any $|\phi| = n$ where the last probability is only over the randomness of R.

The actual reduction should solve SAT even if the oracle provides collisions for $1/\mathrm{poly}(n)$ of the queries, but since our result is of negative form this definition only makes the result stronger.

Theorem A.4. *If there exist a reduction R from CR of H to* SAT *with k rounds of adaptivity, then* **co**-SAT $\in \mathbf{AM}[k]$.

The result of [PSSV07] shows that the existence of such a reduction for $k = \mathrm{polylog}(n)$ implies that the exponential hierarchy collapses to the third level, and the reduction with $k = o(n)$ improves the number of rounds over the best known interactive proof protocol for **co**-SAT.

Proof Sketch of Theorem A.4: The proof of Theorem A.4 is by forcing the prover to provide a *random* answer to each of the queries the reduction makes to its oracle. Since the oracle answers are independent of the randomness of the reduction, the verifier gets to result of simulating one running of the reduction and therefore can decided whether $\phi \in \mathsf{SAT}$ or not.

The oracle that the prover is forced to behave similar to is an oracle that given h, chooses a random x and then picks a random y satisfying $h(x) = h(y)$ as follows. To force the prover to mimic this behavior, the verifier samples $x \in \{0,1\}^n$

at random and sends $h(x)$ to the prover and asks the prover for $|\mathsf{Sib}(x)| = |f^{-1}(x)|$ and receives s. Using the lower-bound protocol (Lemma 2.1), the verifier can make sure that $s(1-\epsilon) \leq |\mathsf{Sib}(x)|$ for arbitrary small $\epsilon = 1/\text{poly}(n)$. Since the verifier has a secret sample from the set $\mathsf{Sib}(x)$ (i.e., x), it can use the upper-bound protocol (Lemma 2.1) to make sure $s(1+\epsilon) \geq |\mathsf{Sib}(x)|$ as well. One point to notice is that the soundness of the upper-bound protocol is small (but not too small), and in order to handle this issue the verifier does what we mentioned for many x instances in parallel, and chooses one of them after running to lower-bound protocol for all of them. (To see why doing it in parallel is not a problem, note that we can pretend that the random x conditioned on the revealed $h(x)$ is chosen at random *after* the prover commits to $\mathsf{Sib}(x)$.

After being convinced of the size of $\mathsf{Sib}(x)$ for arbitrarily good enough precision, the verifier, with the help of the prover, samples a random member of the set $\mathsf{Sib}(x)$ (by using polynomially-independent hash functions — Lemma 2.2). Since there are many collisions, *i.e.*$|\mathsf{Sib}(x)| > 1$ often, if the verifier gets enough number of (x, y) pairs such that $h(x) = h(y)$ as above, then with high probability she gets $x \neq y$ for at least one of the pairs. Therefore each query (or rather each adaptivity round) will be substituted by a constant round protocol between the verifier and the prover and forcing the prover to give us a random answer. That results in a $\mathbf{AM}[O(k)]$ for **co**-SAT. Using the result of [] one can reduce the number of rounds to k.

Lemma A.5. *Let* $h\colon \{0,1\}^n \to \{0,1\}^{n-1}$ *be a hash function. Then* $\Pr_{x,y\leftarrow_R\{0,1\}^n}[x \neq y \mid h(x) = h(y)] \geq 1/2$.

Proof. We show $\Pr_{x,y\leftarrow_R\{0,1\}^n}[x = y \mid h(x) = h(y)] \leq 1/2$. Let $S_1, \ldots, S_r$ be the partition of $\{0,1\}^n$ according to the equivalence relation that $x \approx y$ iff $h(x) = h(y)$. Then $\Pr_{x,y\leftarrow_R\{0,1\}^n}[x = y \mid h(x) = h(y)] = \sum_i \frac{|S_i|}{2^n} \cdot \frac{1}{|S_i|} = \frac{r}{2^n} \leq \frac{2^{n-1}}{2^n} = \frac{1}{2}$. ∎

Let t be the number of queries that the reductions asks (k was the number of adaptivity rounds). The final $\mathbf{AM}$ protocol for the **co**-SAT problem is as follows. The parameters m, ϵ, δ will be determined later. For each query h, the verifier samples m many $x \leftarrow_R \{0,1\}^n$ at random and for each of them asks the prover to give a $1 \pm \epsilon$ good estimate on the size of $\mathsf{Sib}(x)$ by running the upper-bound and lower-bound protocols. Then she chooses n distinct instances $x_1, \ldots, x_n$ out of all those m instances and runs the sampling protocol for all of them to get $y_1, \ldots, y_n$. If for any $1 \leq i \leq n$ she gets $x_i \neq y_i$, she will use that pair as the oracle answer.

Assuming y_i's are random samples of $\mathsf{Sib}(x_i)$'s, by Lemma A.5 with probability at least $1 - 2^{-n}$ one of them is a distinct pair. So we ignore this exponentially small error. In order to keep the statistical distance of all samples small from uniform samples, we want $10tn\epsilon \leq O(1/n)$. So we can take $\epsilon = \frac{1}{n^2 t}$. Let p be the fraction of m instances on which the prover cheats in the upper-bound protocol. If $mp = n/\epsilon$, then the probability that the prover is not caught in any of those interactions (only for simulating the query h) is at most $(1 - 10\epsilon)^{n/\epsilon} = \exp(-n)$, so we can safely ignore this probability as well. The probability that any of those n chosen instances are among these mp "bad" instances is at most np which we want it to be at most $\frac{1}{tn}$. So we will take $p = \frac{1}{tn^2}$, and hence $m = \frac{n}{p\epsilon} = tn^5$. Finally we want δ to be small enough such that the sum of all the soundness errors for the lower-bound protocols be at most $O(1/n)$ (This makes the sum of completeness errors of all the upper-bound and lower-bound protocols to be $O(1/n)$ as well). So we want $m\delta t \leq O(1/n)$. So, we take $\delta = \frac{1}{mtn} = \frac{1}{t^2 n^6}$. By setting the parameters this way, either the verifier rejects, or gets a running of the reduction with an oracle which is at most $O(1/n)$ far from an oracle which is independent of the reduction's randomness and answer *all the queries*[14] of the reduction. So, it can decide the membership in SAT with probability at least $2/3 - O(1/n) \gg 1/2$.

[14]Therefore what we prove in Theorem A.4 is that the task of breaking collision resistance of a hash function can not be **NP**-hard (through randomized Cook reductions) even in the *worst case*.