A New Sampling Protocol and Applications to Basing Cryptographic Primitives on the Hardness of NP

Iftach Haitner
Microsoft Research
New England Campus
Cambridge, USA
Email:iftach@microsoft.com

Mohammad Mahmoody
Computer Science Department
Princeton University
Princeton, USA
Email:mohammad@cs.princeton.edu

David Xiao
LRI
Université Paris-Sud
Paris, France
Email: dxiao@lri.fr

***Abstract*—We investigate the question of what languages can be decided efficiently with the help of a recursive collision-finding oracle. Such an oracle can be used to break collision-resistant hash functions or, more generally, statistically hiding commitments. The oracle we consider, Sam_d where d is the recursion depth, is based on the identically-named oracle defined in the work of Haitner et al. (FOCS '07). Our main result is a constant-round public-coin protocol "AM–Sam" that allows an efficient verifier to emulate a Sam_d oracle for any constant depth $d = O(1)$ with the help of a $\mathbf{BPP}^{\mathbf{NP}}$ prover. AM–Sam allows us to conclude that if L is decidable by a k-adaptive randomized oracle algorithm with access to a $\mathsf{Sam}_{O(1)}$ oracle, then $L \in \mathbf{AM}[k] \cap \mathbf{coAM}[k]$.**

The above yields the following corollary: assume there exists an $O(1)$-adaptive reduction that bases constant-round statistically hiding commitment on $\mathbf{NP}$-hardness, then $\mathbf{NP} \subseteq \mathbf{coAM}$ and the polynomial hierarchy collapses. The same result holds for any primitive that can be broken by $\mathsf{Sam}_{O(1)}$ including collision-resistant hash functions and $O(1)$-round oblivious transfer where security holds statistically for one of the parties. We also obtain non-trivial (though weaker) consequences for k-adaptive reductions for any $k = \mathrm{poly}(n)$. Prior to our work, most results in this research direction either applied only to non-adaptive reductions (Bogdanov and Trevisan, SIAM J. of Comp. '06 and Akavia et al., FOCS '06) or to one-way permutations (Brassard FOCS '79).

The main technical tool we use to prove the above is a new constant-round public-coin protocol (SampleWithSize), which we believe to be of interest in its own right, that guarantees the following: given an efficient function f on n bits, let D be the output distribution $D = f(U_n)$, then SampleWithSize allows an efficient verifier Arthur to use an all-powerful prover Merlin's help to sample a random $y \leftarrow D$ along with a good multiplicative approximation of the probability $p_y = \Pr_{y' \leftarrow D}[y' = y]$. The crucial feature of SampleWithSize is that it extends even to distributions of the form $D = f(U_{\mathcal{S}})$, where $U_{\mathcal{S}}$ is the uniform distribution on an efficiently decidable subset $\mathcal{S} \subseteq \{0,1\}^n$ (such D are called efficiently samplable with *post-selection*), as long as the verifier is also given a good approximation of the value $|\mathcal{S}|$.

***Index Terms*—sampling protocols; collision-resistant hash functions; constant-round statistically hiding commitments; black-box lower bounds;**

I. Introduction

The ability to sample from efficiently decidable sets (i.e., membership in such a set can be decided efficiently, but sampling from the set might be hard) is an extremely powerful computation resource, to the point that having such ability for *any* decidable set implies $\mathbf{P} = \mathbf{NP}$. In this work we study less powerful samplers, which only agree to sample from more carefully chosen sets. We show that while these samplers can be used to break certain cryptographic primitives, they seem not to be strong enough to decide arbitrary $\mathbf{NP}$ languages. We then use this fact to give negative evidence on the possibility of basing such primitives on $\mathbf{NP}$ hardness.

Consider the sampler that gets a circuit C over $\{0,1\}^n$ as input, and outputs two random values x and x' in $\{0,1\}^n$ conditioned that $C(x) = C(x')$. Such a sampler is known as a "collision finder", and breaks the security of any family of collision-resistant hash functions [62].[1] We consider the following generalization of the above sampler: the sampler Sam_d, where $d \in \mathbb{N}$, gets up to d recursive calls, each of the form $(C_1, \ldots, C_i, x)$, where $i \leq d$, each of the C_j's is a circuit over $\{0,1\}^n$ and $x \in \{0,1\}^n$. Sam_d answers depth 1 calls, $(C_1, \cdot)$, with a random element in $\{0,1\}^n$. For depth $i > 1$ calls, Sam_d first checks that it was previously queried with $(C_1, \ldots, C_{i-1}, \cdot)$ and answered with x (otherwise, it aborts). If the check passes, then Sam_d answers with a random element in $C_1^{-1}(C_1(x)) \cap \ldots \cap C_i^{-1}(C_i(x))$. (Note that Sam_2, is equivalent to the "collision finder" we described above). Such a sampler is very powerful, as it can be used for breaking the binding of any d-round statistically hiding commitments [64, 31].

Commitment schemes are the digital analogue of a sealed envelope. In such a scheme, a sender and a receiver run an interactive protocol where a sender commits to a bit b. In case the commitment is statistically hiding, then the protocol guarantees that from the receiver's point of view there exists roughly equal chance that the sender has committed to $b = 0$ or $b = 1$ (hence the bit b is hidden from the receiver information-theoretically). In addition, the scheme guarantees that a computationally-bounded sender can only find one way to decommit. Statistically hiding commitments are widely used

[1] A family of collision resistant hash functions is a family of, efficient, compressing functions with the following security guarantee: given a random function h in the family, it is hard to find $x \neq x'$ satisfying $h(x) = h(x')$.

throughout all of cryptography, with applications including, but not limited to, constructions of zero-knowledge protocols [12, 50, 22, 6, 32], authentication schemes [14], and other cryptographic protocols (e.g., coin-tossing [45]). Hence, it is highly important to study the minimal assumptions required for building them. Since Sam_d breaks any d-round statistically hiding commitments, it is very informative to learn what hardness assumptions Sam_d does *not* break (in particular, we have little hope to base d-round statistically hiding commitments on such assumptions). The following theorem shows that for a constant d, Sam_d is not "too powerful".

We write $L \in \mathbf{BPP}^{\mathcal{O}[k]}$ to mean that L can be decided by $A^{\mathcal{O}}$, where A is a k-*adaptive* (randomized) oracle-aided algorithm using an oracle $\mathcal{O}$: A 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 or later rounds. We say A is non-adaptive if $k = 1$.

Theorem I.1 (Main theorem, informal)**.** *For any $d = O(1)$ and any efficient oracle-aided algorithm A, there exists an interactive protocol* $\mathsf{AM{-}Sam}$ *with the following guarantee: either the output of the efficient verifier is statistically close to the output of A^{Sam_d}, or (if the prover cheats) the verifier aborts with high probability. Furthermore, the round complexity of* $\mathsf{AM{-}Sam}$ *is the same as the adaptivity of the oracle queries of A, and the honest prover strategy has complexity $\mathbf{BPP}^{\mathbf{NP}}$ (while the protocol remains sound against unbounded cheating provers).*

We apply this theorem to understand what languages can be efficiently decided by randomized oracle-aided algorithms with oracle access to $\mathsf{Sam}_{O(1)}$, where the strength of the implication is a result of the adaptivity of the calls to $\mathsf{Sam}_{O(1)}$ made by the algorithm. Theorem I.1 yields a k-round protocol for any language $L \in \mathbf{BPP}^{\mathsf{Sam}_{O(1)}[k]}$. Since $\mathbf{BPP}^{\mathsf{Sam}_{O(1)}[k]}$ is closed under complement, the above implies the following corollary.

Corollary I.2 (Limits of languages decidable using oracle access to $\mathsf{Sam}_{O(1)}$)**.** *It holds that $\mathbf{BPP}^{\mathsf{Sam}_{O(1)}[k]} \subseteq \mathbf{AM}[k] \cap \mathbf{coAM}[k]$. In particular, every $L \in \mathbf{BPP}^{\mathsf{Sam}_{O(1)}[k]}$ has a k-round interactive proof where the honest prover has complexity $\mathbf{BPP}^{\mathbf{NP}}$. Furthermore, if L is $\mathbf{NP}$-complete, then the following consequences hold.*

- $k = \mathrm{poly}(n)$*: $\mathbf{co\text{-}NP}$ has a public-coin $O(k)$-round interactive proof with honest prover complexity $\mathbf{BPP}^{\mathbf{NP}}$.*
- $k = \mathrm{polylog}(n)$*: the quasipolynomial hierarchy collapses to its third level (by [55]).*
- $k = O(1)$*: $\mathbf{PH} = \Sigma_2$ (by [10]).*

Since the polynomial hierarchy is widely conjectured not to collapse, it follows that $\mathbf{NP}$-complete languages are unlikely to be in $\mathbf{BPP}^{\mathsf{Sam}_{O(1)}[k=O(1)]}$. For $k = \mathrm{polylog}(n)$, the collapse is less understood, but it is still reasonable to conjecture that such a collapse does not occur. For $k = o(n)$ the consequence may not be implausible, but would nevertheless lead to surprising progress on the long-standing open question of reducing the round complexity of interactive proofs for $\mathbf{co\text{-}NP}$ [46]. Finally for $k = \mathrm{poly}(n)$, as pointed out to us by Holenstein [39], it would answer a long-standing open question of Babai et al. [5] about reducing the complexity of the prover in interactive proofs for $\mathbf{co\text{-}NP}$ from $\mathbf{BPP}^{\#\mathbf{P}}$ to $\mathbf{BPP}^{\mathbf{NP}}$ (in fact this question is even open for multi-prover interactive proofs). Thus, depending on the adaptivity k, Corollary I.2 gives an indication of either the implausibility or the difficulty of proving that $\mathbf{NP}$-complete languages can be decided using the help of $\mathsf{Sam}_{O(1)}$.

A. Application to Basing Cryptography on $\mathbf{NP}$-Hardness

Much of modern cryptography relies on computational intractability assumptions; starting with seminal works of Diffie and Hellman [17] and Goldwasser and Micali [28], 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 [57, 18, 1]; unfortunately, the existence of sub-exponential algorithms for factoring [13] and of efficient quantum factoring algorithms [61] 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.

The holy grail of this search would be to base cryptography on the minimal assumption of $\mathbf{P} \neq \mathbf{NP}$; 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 almost all other cryptographic primitives [41, 53]), the former is a "worst-case" assumption while the latter is of "average-case" nature, hence making the first assumption much more desirable. In fact, this goal dates back to the seminal paper by Diffie and Hellman [17].

Most constructions and proofs in the cryptographic literature are black-box, so it is worthwhile to understand whether black-box reductions can base cryptographic primitives on $\mathbf{NP}$-hardness. A black-box reduction (also known as, black-box proof of security) from the security of a cryptographic primitive to $\mathbf{NP}$-hardness, is an efficient randomized oracle algorithm R such that given any oracle $\mathcal{O}$ that breaks the security of the cryptographic primitive, $R^{\mathcal{O}}$ solves SAT. The question of whether black-box reductions can be used to base cryptography on $\mathbf{NP}$-hardness has been previously studied in [11, 19, 8, 21, 4, 54].

Since $\mathsf{Sam}_{O(1)}$ breaks the security of d-round statically hiding commitments, it also breaks the wide variety of cryptographic primitives that yield such commitments via constant-adaptive black-box reductions. This list includes: collection of claw-free permutations with an efficiently-recognizable index set [22], collision-resistant hash functions [16, 49], (singly) homomorphic encryption [42], constant-round protocols for oblivious-transfer and private information retrieval schemes where the security of one of the parties holds information theoretically [31], the average-case hardness of $\mathbf{SZKP}$ [52], constant-round statistically *binding* commitments secure against selective opening attacks [65], and constant-round inaccessible entropy generators [33]. The following corollary states that if any of the above primitives can be based on $\mathbf{NP}$-hardness via a black-box reduction R, then $R^{\mathsf{Sam}_{O(1)}}$ decides SAT.

Corollary I.3 (immediate by Corollary I.2)**.** *Let P be a cryptographic primitive whose security can be broken by $\mathsf{Sam}_{O(1)}$. Let R be a k-adaptive reduction that bases the existence of P on $\mathbf{NP}$-hardness. Then $\mathrm{SAT} \in \mathbf{AM}[k] \cap \mathbf{coAM}[k]$, where the honest provers that realize this containment are in $\mathbf{BPP}^{\mathbf{NP}}$. The various consequences for different k given in Corollary I.2 also hold.*

We remark that previous results studying the analogous question of basing (general) one-way functions on $\mathbf{NP}$-hardness were restricted to non-adaptive reductions [19, 8, 4]. Other works do consider adaptive reductions, but with respect to more structured primitives [11, 21, 4]. See Section I-C1 for the description of previous works.

B. Main Tool — A New Sampling Protocol

Our main tool for proving Theorem I.1, which is also our main technical contribution, is a new constant-round public-coin sampling protocol that we believe to be of independent interest. A distribution D is called *efficiently samplable* if it is the output distribution of an efficient function $f : \{0,1\}^n \to \{0,1\}^*$ (i.e., $D = f(U_n)$). A distribution is efficiently samplable *with post-selection* if $D = f(U_{\mathcal{S}})$ where $U_{\mathcal{S}}$ is the uniform distribution over an efficiently decidable set $\mathcal{S} \subseteq \{0,1\}^n$. Such distributions have also been studied in the context of randomized algorithms [37]. We emphasize that although $\mathcal{S}$ is efficiently decidable, it is not necessarily possible to efficiently sample uniform elements of $\mathcal{S}$.

Our "Sample With Size" protocol takes $f, \mathcal{S}$, and a good approximation of $|\mathcal{S}|$ as input, and enables an efficient verifier to sample a uniform $y \in f(\mathcal{S})$, along with a good approximation of the value $\left|f^{-1}(y) \cap \mathcal{S}\right|$.

Lemma I.4. *(Sampling With Size protocol, informal) There exists a constant-round public-coin protocol* $\mathsf{SampleWithSize}$*, where the parties get as a common input an efficiently decidable set $\mathcal{S} \subseteq \{0,1\}^n$, an efficiently computable function $f : \mathcal{S} \to \{0,1\}^*$ and a good approximation (i.e., within $(1 \pm \frac{1}{\mathrm{poly}(n)})$ factor) of $|\mathcal{S}|$, and has the following guarantees:*

Either $\mathsf{V}_{\mathsf{SWS}}$ outputs a pair (x, s_x) such that 1) x is distributed $(1/\mathrm{poly}(n))$-statistically close to the uniform distribution over $\mathcal{S}$, and 2) s_x is a good approximation for $\left|f^{-1}(f(x)) \cap \mathcal{S}\right|$, or (if the prover cheats) the verifier aborts with high probability. Furthermore, the honest prover has complexity $\mathbf{BPP}^{\mathbf{NP}}$, while the cheating prover may be unbounded.

C. Related Work

1) $\mathbf{NP}$-hardness and cryptography: Brassard [11] showed that if there exists a deterministic black-box reduction from $\mathbf{NP}$-hardness to inverting a one-way permutation, then $\mathbf{NP}$= $\mathbf{co}$-$\mathbf{NP}$. Bogdanov and Trevisan [8], building on earlier work of Feigenbaum and Fortnow [19], showed that if there exists a *non-adaptive* randomized black-box reduction from $\mathbf{NP}$-hardness to inverting a one-way function (or more generally, to a hard on the average problem in $\mathbf{NP}$), then $\mathbf{NP} \subseteq \mathbf{coAM}/\mathrm{poly}$, which is considered implausible since the polynomial hierarchy would collapse to the third level [66]. Akavia et al. [4] improved this result for the case of reductions to inverting one-way functions, to show that the same hypothesis implies the uniform conclusion $\mathbf{NP} \subseteq \mathbf{coAM}$, which implies that the polynomial hierarchy collapses to the second level [10]. Goldreich and Goldwasser [21] showed that adaptive reductions basing public-key encryption schemes with the special property that the set of invalid ciphertexts is verifiable in $\mathbf{AM}$, on $\mathbf{NP}$-hardness would also imply that $\mathbf{NP} \subseteq \mathbf{coAM}$. Finally, Pass [54] takes a different route and showed that if a *specific type* of *witness-hiding* protocol exists, then an arbitrarily adaptive reduction from $\mathbf{NP}$-hardness to the existence of one-way functions implies that $\mathbf{co}$-$\mathbf{NP} \subseteq \mathbf{AM} \cap \mathbf{coAM}$. As recently pointed out by Haitner et al. [34], however, it is unlikely that known witness-hiding protocols are of the type required by [54].

We remark 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 [50], pseudorandom number generators [38, 40, 36], and certain lattice-based cryptosystems [3, 47]. One may hope in particular that lattice problems might someday be used to prove that $\mathbf{P} \neq \mathbf{NP}$ implies one-way functions or collision-resistant hash functions, since they already exhibit a worst-case to average-case hardness reduction.[2] Previous work such as [8, 4] do not rule out the possibility that any of these (or some other adaptive) techniques may succeed.

2) The oracle Sam*:* Simon [62] studied collision finders that break collision-resistant hash functions. In our language of Sam, a collision finder is the same as Sam_2, i.e., Sam where queries of at most depth 2 are allowed. Simon [62] considered the sampler Sam_2^{π} — a generalization of Sam_2 that gets circuits with π-gates, where π is a random permutation oracle. He showed that while Sam_2^{π} breaks any collision-resistant hash functions relative to random permutation π (i.e., the hash function is allowed to use π-gates), it cannot invert π. Continuing this line of research, Haitner et al. [31] showed that Sam_d^{π} breaks all d-round statistically hiding commitments, even those implemented using π, but Sam_d^{π} does not help to invert π if $d = o(n/\log n)$. As a consequence, the above results rule out the possibility of basing $o(n/\log n)$-round

[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 $\mathbf{NP} \cap \mathbf{co}$-$\mathbf{NP}$.

statistically hiding commitments on the existence of one-way functions/permutations, using *fully-black-box* reductions — a reduction from (the security of) a primitive to one-way function is fully-black-box, if the proof of security is black-box (in the sense of Corollary I.3), and *in addition* the construction uses the one-way function as a black-box (i.e., as an oracle). Note that these results are incomparable to the result stated in Corollary I.3. On one hand, they rule out all fully-black-box reductions *unconditionally* without restrictions on adaptivity, and the reductions they consider are starting from one-way functions rather than $\mathbf{NP}$-hardness (and thus "harder" to refute). On the other hand, their results do not apply to constructions that may use the code of the one-way function (or, in our case, the structure of the $\mathbf{NP}$-complete language). In contrast, Corollary I.3 also applies to reductions where the construction is non-black-box, which permits, for example, the construction to exploit the fact that YES instances of $\mathbf{NP}$ languages have efficiently verifiable witnesses. In other words, Corollary I.3 only requires that the *security analysis* be black-box. We refer the reader to Reingold et al. [56], which, although not focused on our case where the construction is non-black-box but the security analysis is black-box, is useful for understanding the distinctions between various notions of reductions.

a) Sam *and zero knowledge.:* In recent work, Gordon et al. [30] observe that our main result is useful in the context of understanding zero-knowledge proofs. In particular, they prove using Theorem I.1 that if a language L has a constant-round black-box computational zero-knowledge proof based on one-way permutations with a k-adaptive simulator, then $L \in \mathbf{AM}[k] \cap \mathbf{coAM}[k]$. Their result suggests that reducing the round complexity of known constructions of zero-knowledge proofs based on one-way permutations for $\mathbf{NP}$ (e.g., [25, 7]) (all of which have super-constant round complexity) to a constant number of rounds is implausible (if the simulator must be $O(1)$-adaptive) or at least difficult to prove (regardless of the simulator's adaptivity).

3) Efficiently samplable distributions with post-selection:

a) Estimating statistics.: Estimating statistical properties *efficiently samplable* distributions has long been studied in theoretical computer science [29, 20, 2, 51, 59, 23]. Typically, estimating interesting parameters of samplable distributions (and therefore also of samplable distributions with post-selection) such as entropy or statistical difference is hard (e.g., $\mathbf{SZKP}$-hard). Nevertheless, for samplable distributions it was known that an efficient verifier can estimate various parameters in constant rounds with the help of an all-powerful prover.

b) Bounding set-size protocols.: The constant-round public-coin lower bound protocol of Goldwasser and Sipser [29] can be used to lower-bound the size of efficiently decidable sets. Namely, on input an efficiently decidable set $\mathcal{S}$ and a value s, the prover makes the verifier accept iff $|\mathcal{S}| \geq s$. Fortnow [20] (see also Aiello and Håstad [2]) gives a constant-round protocol that upper-bounds the sizes of efficiently decidable sets $\mathcal{S}$ where *in addition* the verifier has a uniform element of $\mathcal{S}$ that is *unknown* to the prover.

These protocols are related to our protocol SampleWithSize. For example, one can estimate with respect to $D = f(U_n)$ the integer $s_y = |f^{-1}(y)|$ for a random $y \leftarrow D$ by lower-bounding and upper-bounding the set $|f^{-1}(y)|$. In particular, the upper bound [20, 2] can be applied in this case, since the verifier can sample $x \leftarrow U_n$, compute $y = f(x)$ and ask the prover for an upper bound on the size of the set $f^{-1}(y)$ without revealing x. This is one way to prove SampleWithSize for the special case $\mathcal{S} = \{0,1\}^n$.

We cannot necessarily apply, however, the upper bounds of [20, 2] to do the same thing with *post-selected* distributions $D = f(U_{\mathcal{S}})$; even though $f^{-1}(y)$ is efficiently decidable, it may not be possible to efficiently generate $y \leftarrow f(U_{\mathcal{S}})$ and $x \in f^{-1}(y)$ such that x is hidden from the prover. As we discuss in paragraph II-B2b, handling post-selected distributions is necessary to obtain Theorem I.1. Although one-sided lower-bound estimates can be obtained from the lower-bound protocol of [29], it is unknown how to get two-sided estimates using the upper bound protocol of [20, 2], where the difficulty is to obtain secret samples from $U_{\mathcal{S}}$. In contrast, SampleWithSize *does* guarantee a two-sided bound for $|f^{-1}(f(x)) \cap \mathcal{S}|$ for a random x in $U_{\mathcal{S}}$.

c) Sampling.: Using an all-powerful prover to help sample is an old question in computer science, dating at least to the works of Valiant and Vazirani [63] and Impagliazzo and Luby [41]. In building SampleWithSize, we use a sampling protocol from Goldreich et al. [27]. This constant-round public-coin protocol takes as input an efficiently decidable set $\mathcal{S}$ and a good approximation of $|\mathcal{S}|$, and outputs a nearly-uniform element of $\mathcal{S}$. Our protocol SampleWithSize uses their sampling protocol and extends it by also giving set size information about the sample that is generated.

Another protocol that seems related to SampleWithSize is the random selection protocol of Goldreich et al. [24]. Their protocol accomplishes a goal similar to the protocol of [27], allowing a verifier to select a random element of a set. Their protocol, however, cannot be applied in our context as it requires super-constant round complexity. Other related work include random selection protocols arising in the study of zero knowledge [15, 26, 60], but none of these protocols provides the size information that is provided by SampleWithSize.

D. $\mathsf{Sam}_{O(1)}$ *Vs.* Sam_2

It is worthwhile noting that Theorem I.1 for non-recursive collision finders (i.e., Sam_2), can be proven via a straightforward application of the lower-bound protocol of Goldwasser and Sipser [29] and the upper-bound protocol of [20, 2]. See Section II-A for an illustration of these easier proofs.

Various evidence suggests, however, that $\mathsf{Sam}_{O(1)}$ is more powerful than Sam_2. There is no known way to "collapse" the depth (i.e., to show that Sam_2 suffices to emulate Sam_d for $d > 2$), and under various assumptions there exist problems solvable using $\mathsf{Sam}_{O(1)}$ but not Sam_2 (for example, the average-case hardness of $\mathbf{SZKP}$ [52] and constant-round parallelizable zero-knowledge proofs for $\mathbf{NP}$ [33], both imply constant-round statistically hiding commitment, but not collision-resistant hash functions). Therefore, we do not focus on the (admittedly simpler) proof of Theorem I.1 for the case of Sam_2, and rather we build our machinery of SampleWithSize in order to prove Theorem I.1 for the case of $\mathsf{Sam}_{O(1)}$.

E. *Contrast to previous work*

Sam_d is in a sense a "canonical recursive collision finder". Similarly, one could consider a "canonical function inverter" that takes as input a circuit C and a value y and outputs a random element of $C^{-1}(y)$. Such an oracle would break all one-way functions. One could then ask whether it is possible to construct some kind of "AM-Inv" that emulates this canonical inverter. Such a result would strengthen our main theorem, since an inverter can in particular find collisions.

Unfortunately, it is not known how to build such an AM-Inv. The main difficulty is handling cheating provers, who claim that the given query is not invertible. Notice that for the problem of inverting a function, it is possible to ask queries (C, y) where y is not in the image of C.[3] In this case the oracle must say that the query is invalid. Since there is no efficient way to verify that y is *not* in the image of C, a cheating prover can claim, say, that none of the verifier's queries are in the image of C even when some are valid queries. In general, it is not known how to catch this kind of cheating, since proving that y is not in the image of C is a **co-NP** statement.

As already mentioned in Section I-C, various works have gotten around this difficulty using additional restrictions either on the way the inverting oracle is called (e.g., non-adaptivity) or on the kinds of functions that the oracle inverts (e.g., one-way permutations). The main reason we are able to build AM–Sam whereas building "AM-Inv" seems out of reach, is that in our setting, unlike the inverting oracle, Sam_d can never respond "failure" to a query that passes the sanity checks (since these checks ensure that collisions always exist).

Organization

We give an high level description of our techniques in Section II. Where the formal statements and proofs of our result can be found in [35].

II. Our Techniques

In this section we overview our proof for Theorem I.1. As a warmup, we start with the much simpler case of Sam_2 (i.e., $d = 2$), and then move to any constant d.

This overviews presented here assume familiarity with the lower-bound protocol of [29], the upper-bound protocol of [2], and the uniform-sampling protocol of [27] (see **??** for formal definitions).

A. *The Case of* Sam_2

Given an efficient oracle-aided algorithm A, we construct an **AM** protocol that emulates A^{Sam_2} as follows: the protocol's high-level strategy is standard; the verifier tries to emulate the execution of A^{Sam_2} by picking random coins for the reduction A, and whenever A asks an oracle query to Sam_2, the verifier engages the prover in a protocol such that the distribution of the output is close to what Sam_2 would output, or else the verifier rejects.

d) Depth 1 queries:: a query $(C_1, \perp)$ is answered as follows:

1) The verifier samples $x \leftarrow \{0,1\}^n$ at random and send $y = C_1(x)$ to the prover.
2) The prover responds with $s = |C_1^{-1}(C_1(x))|$.
3) Using the lower-bound protocol of [29] and the upper-bound protocol of [2], the verify checks that $s \approx |C_1^{-1}(C_1(x))|$.[4]
4) The verifier stores (x_i, s_i) in a lookup table, and returns x.

e) Depth 2 queries:: On query (C_2, x), the verifier checks that C_1 was asked before and was answered with x (if not it rejects). The it looks up the value of s_x previously stored and uses it to sample a random member of the set $\mathsf{Sib}(x)$ using the sampling lemma of [27]. It easily follows that this sample is close to uniformly distributed in $C_1^{-1}(C_1(x))$.

Assuming that the prover does not cause the verifier to reject with high probability, each query of A (or rather each adaptive round of parallel queries) is answered correctly (up to some small statistical deviation), and the proof follows.

B. *The Case of* $\mathsf{Sam}_{O(1)}$

We start by showing how to generalize the above approach for using protocol SampleWithSize to implement protocol AM–Sam, and then give details on the implementation of protocol SampleWithSize itself.

Let us start with a more precise description of Sam_d. On input $(C_1, \ldots, C_i, x)$, where $x \in \{0,1\}^n$ and each C_j is a circuit over $\{0,1\}^n$, Sam_d performs the following "sanity check": it checks that $i \leq d$, and if $i > 1$ then it also checks that it was previously queried on $(C_1, \ldots, C_{i-1}, x')$ (for some $x' \in \{0,1\}^n$) and answered with x. If any of these checks fail, Sam_d returns "failure". Otherwise, Sam_d returns a random element x' in $\mathcal{S}(C_1, \ldots, C_{i-1}, x) := \{x' \in \{0,1\}^n \colon \forall 1 \leq j \leq i-1, C_j(x') = C_j(x)\}$ (if $i = 1$, it returns a random $x' \in \{0,1\}^n$). Viewed differently, x' is a random collision with x for depth $i-1$ with respect to $C_1, \ldots, C_{i-1}$ (since it satisfies $C_j(x') = C_j(x)$ for every $1 \leq j \leq i-1$).

In protocol AM–Sam, the verifier chooses A's random coins at random and then emulates A^{Sam_d}, while answering each query $(C_1, \ldots, C_i, x)$ to Sam_d using the following subprotocol: the verifier first performs (using the data stored during previous executions, see below) the sanity check of Sam_d, and aborts and rejects in case any of these tests fail. Otherwise it does the following:

[3] Actually, as pointed out by [9], asking such queries might be very useful.

[4] Actually, the upper-bound protocol of [2] does not give a useful upper bound for *single* query, but only guarantees good upper bound for most queries from a large enough set of queries. Nevertheless, the above approach can be slightly modified to go through, by choosing many x's, applying the set upper and lower bounds protocol on each of them, and finally picking one of them at random.

In case $i = 1$: The verifier sets $\mathcal{S} = \{0,1\}^n$, $s = 2^n$, and $f = C_1$ and runs SampleWithSize to get a random sample $x_1 \in \{0,1\}^n$ and an approximation $s_1 \approx |\{x' \in \{0,1\}^n : C_1(x_1) = C_1(x')\}|$. The verifier stores an entry $((C_1, x_1), s_1)$ in its memory, and returns x_1 to A as the query's answer.

In case $i > 1$: The verifier looks up the entry $((C_1, \ldots, C_{i-1}, x), s_{i-1})$ from its memory (the sanity checks guarantee that such an entry must exist, since x was the answer for a previous query $(C_1, \ldots, C_{i-1}, \cdot)$). Run SampleWithSize on $\mathcal{S} = \mathcal{S}_i = \{x' \in \{0,1\}^n : \forall 1 \leq j \leq i-1,\ C_j(x') = C_j(x)\}$, $f = C_i$, and s_{i-1} in order to obtain $x_i \in \mathcal{S}$ and the approximation $s_i \approx |\{x' \in \mathcal{S} : C_i(x_i) = C_i(x')\}|$. As in the case $i = 1$, the verifier stores an entry $((C_1, \ldots, C_i, x_i), s_i)$ in its memory, and returns x_i.

To see that AM–Sam indeed behaves like A^{Sam_d}, we first note that Lemma I.4 yields that for depth 1 queries, SampleWithSize returns x_1 that is (close to) uniform in $\{0,1\}^n$, which is what Sam_d would answer. In addition, SampleWithSize outputs a good approximation s_1 for $|C_1^{-1}(C_1(x_1))|$, which can be used as input for depth 2 queries to AM–Sam. Since s_1 is a good approximation, this means that a depth 2 query will be answered by SampleWithSize with x_2 where x_2 is a near-uniform element of $C_1^{-1}(C_1(x_1))$, again, just as Sam_d would answer. SampleWithSize also outputs a good approximation $s_2 \approx |C_2^{-1}(C(x_2))|$, which can be used for depth 3 queries, and so on.

The above is done in parallel for each of the k adaptive rounds of oracle queries. The approximation error of s_i grows as the depth increases, and from the formal statement of Lemma I.4 it follows that we can repeat the above process a constant number of times. Unfortunately, the accumulated error becomes super-polynomial for any $d = \omega(1)$.

1) The protocol SampleWithSize*:* The underlying idea behind the soundness proof of SampleWithSize is to force the prover to behave "correctly", by using an accurate estimate of the average preimage size of $y \leftarrow D := f(U_{\mathcal{S}})$. Here, the average preimage size is defined as $\mu(D) := \mathrm{E}_{y \leftarrow f(U_{\mathcal{S}})}[\log |f^{-1}(y)|]$, where $f^{-1}(y) := \{x \in \mathcal{S} : f(x) = y\}$. (Note that this is the average on the log-scale; using this scale is crucial for the correctness of our protocol, see below).

The above estimate is then used to force the prover to give many tuples (y_i, s_i) such that $y_i \leftarrow D$ and most of the s_i are good approximations for $|f^{-1}(y_i)|$. We let $\mathsf{VerifyMean} = (\mathsf{P_{VM}}, \mathsf{V_{VM}})$ denote the protocol that guarantees an accurate estimate of the average preimage size, as stated in the following:

Lemma II.1 (Verifying Average Preimage Size, informal)**.** *There is a constant-round public-coin protocol* VerifyMean *that on input $(f, \mathcal{S}, s)$, as in the statement of Theorem I.4, and a real number μ', guarantees the following assuming that $s \approx |\mathcal{S}|$:*

Completeness: if $\mu' = \mu(D)$, then the verifier accepts (when interacting with the honest prover) with high probability.

Soundness: if μ' is far from $\mu(D)$, then the verifier rejects (when interacting with any prover) with high probability.

a) Proving SampleWithSize *using* VerifyMean*.:* we first show how to use VerifyMean to prove SampleWithSize, then discuss how to prove VerifyMean in the next section. On input $\mathcal{S}, f$ and s, where $s \approx |\mathcal{S}|$, the parties do the following:

1) The prover sends to the verifier a real number μ'. The parties run the VerifyMean protocol to verify that μ' is close to $\mu(D)$.
2) The verifier uses the sampling protocol of [27] to sample many uniform points $x_1, \ldots, x_\ell$ in $U_{\mathcal{S}}$, and sets $y_i = f(x_i)$ for all i.
3) The prover sends $s_1 = |f^{-1}(y_1)|, \ldots, s_\ell = |f^{-1}(y_\ell)|$ to the verifier. The parties engage in the [29] lower bound to ensure that indeed $|f^{-1}(y_i)| \geq s_i$ for all i.
4) The verifier computes $\mu'' = \frac{1}{\ell} \sum_{i=1}^{\ell} \log s_i$ and checks whether $\mu' \approx \mu''$. If they are too far apart, it aborts. Otherwise, it outputs (x_i, s_i), for a random $i \in [\ell]$.

Since completeness is straightforward to observe from the definition of the protocols, in the following we focus on describing how to prove the soundness properties of the SampleWithSize using VerifyMean. Intuitively, the lower bound in Step 3 means that if the prover wants to cheat, it can only claim s_i to be smaller than $|f^{-1}(y_i)|$. On the other hand, by a Chernoff bound we know that for large enough ℓ, the empirical average $\frac{1}{\ell} \sum_{\ell} \log |f^{-1}(y_i)|$ will be close to $\mu(D) \approx \mu'$. Therefore, if the prover consistently under-estimates $|f^{-1}(y_i)|$ for many i, then μ'' will much smaller than μ', and we will catch him in Step 4. Together, this implies that $s_i \approx |f^{-1}(y_i)|$ for almost all i, and so outputting (x_i, s_i) for a random i is good with high probability.[5]

2) Verifying the average preimage size: As a warmup, we first give such a verification protocol for the simple case of efficiently samplable sets. We then give an high level description of the seemingly much more complicated case of efficiently decidable sets.

a) Warmup — Efficiently samplable sets.: As already mentioned in Section I-D, proving VerifyMean for the case $\mathcal{S} = \{0,1\}^n$, or more generally for an efficiently samplable $\mathcal{S}$, is a straightforward application of the lower-bound protocol of [29] and the upper bound protocol of [20, 2]). In particular, the following simple protocol suffices:

1) The verifier samples many uniform random samples $x_1, \ldots, x_\ell$ from $\mathcal{S}$, computes $y_i = f(x_i)$ for all i, and he sends $y_1, \ldots, y_\ell$ to the prover.
2) The prover responds with $s_1, \ldots, s_\ell$, where $s_i = |f^{-1}(y_i)|$.
3) In parallel for all $i \in [\ell]$, the verifier engages the prover in the set lower and upper bound protocols to check that

[5]Here the log-scale is crucial. Assume that we would have defined $\mu(D) := \mathrm{E}_{y \leftarrow f(U_{\mathcal{S}})}[|f^{-1}(y)|]$. In this case, the standard deviation "allows" a $\Omega(2^{n/2})$ deviation from the expectation. Hence, the prover can under count the value $|f^{-1}(y)|$ for *all* the (polynomially many) samples without being caught.

$s_i \approx |f^{-1}(y_i)|$. Notice that the verifier is able to run the upper-bound protocol because he has the secret sample x_i.

4) If all the checks pass, then the verifier accepts iff μ' is very close to $\frac{1}{\ell}\sum_{i=1}^{\ell}\log s_i$.

By a Chernoff bound we know that $\mu(D) \approx \frac{1}{\ell}\sum_{i=1}^{\ell}\log|f^{-1}(y_i)|$ with high probability, and so if the verifier accepts in the upper/lower bound protocols, it then also holds that $\mu(D) \approx \frac{1}{\ell}\sum_{i=1}^{\ell}\log s_i$ with high probability. This implies that the verifier accepts if $\mu' = \mu(D)$ and rejects if μ' is far from $\mu(D)$.

b) The general case — Efficiently decidable sets.: The above approach heavily relies on the set upper-bound protocol, which requires a random secret sample from $\mathcal{S}$. While obtaining such a sample is easy for efficiently sample sets, it seems infeasible when the set in hand in only efficiently decidable. Nevertheless, we manage handle the general case by asking the prover for more infirmation about the distribution $D = f(U_{\mathcal{S}})$. Namely, we show that one can verify the *histogram* of D (see Section II-C), and from this histogram the verify compute the value $\mu(D)$ by itself. Finding a more direct protocol for estimating the mean, is an interesting open problem.

C. Histograms

Let D be a distribution over $\{0,1\}^n$, let $p_y = \Pr_{y' \leftarrow D}[y' = y]$, let $\varepsilon \in (0,1]$ and let $m = \log_{1+\varepsilon} 2^n \approx n/\varepsilon$. The ε-histogram of D is a function $h^f : \{0,\ldots,m\} \mapsto [0,1]$, where $h^f(i) = \Pr_{y \leftarrow D}\left[p_y \in (2^{-(i+1)\varepsilon}, 2^{-i\varepsilon}]\right]$. Namely, the histogram tells us the distribution of weights of elements drawn from D. Note that the smaller the ε, the more informative h^f is about D, but h^f's description is larger. Hence, we will consider histograms for small enough $\varepsilon = 1/\operatorname{poly}(n)$.

In the following we define a distance between histograms with the following properties: (1) it is feasible verify whether a claimed histogram is in small distance from the real histogram (without using upper-bound protocols), (2) and given that the claimed histogram is of small distance from the real one, the mean derived by this histogram is close to the real value.

1) Wasserstein distance: We use the 1st Wasserstein distance W1 (also known as Kantorovich distance and Earth Mover's distance) as the distance measure between histograms. This distance is well studied in probability theory [48, 43, 44] and also has application in computer science (e.g., in the realm of image processing [58]). To understand this distance intuitively, think of a histogram h as piles of "earth" on the discrete interval $0,\ldots,m$, where the larger $h(i)$ is, the larger the pile at location i. $\mathsf{W1}(h,h')$ is the *minimal amount of work* that must be done to "push" the configuration of earth given by h to get the configuration given by h'. Recall that in physics, work equals force times distance. For example, pushing a mass of weight 0.1 from bin 2 to bin 3 requires work $0.1 \cdot (3-2) = 0.1$, where pushing the same mass from bin 2 to bin 4 requires $0.1 \cdot (4-2) = 0.2$. The W1 distance for histograms over $\{0,\ldots,m\}$ is defined as:

$$\mathsf{W1}(h,h') = \frac{1}{m} \cdot \sum_{0 \le i \le m} \Big| \sum_{0 \le j \le i} (h(j) - h'(j)) \Big|$$

The intuition is that $|\sum_{0 \le j \le i} h(j) - h'(j)|$ captures the amount of mass "pushed" from the interval $\{0\ldots,i\}$ into the interval $\{i+1,\ldots,m\}$, and taking an integral over all these amounts together gives us the total amount moved.

We first notice that the above distance has the second property we require above (i.e., small variation in the Wasserstein distance implies small difference in the mean), and then, in Section II-C2, explain why is it feasible to verify that a claimed histogram is of small Wasserstein distance from the real one.

From the above definitions it follows that

$$\begin{aligned}
\mu(D) &= \sum_{y \in \operatorname{Supp}(D)} \Pr[D=y] \cdot \log|f^{-1}(y)| \\
&= \log|\mathcal{S}| - \sum_{y \in \operatorname{Supp}(D)} \Pr[D=y] \cdot \log \tfrac{|\mathcal{S}|}{|f^{-1}(y)|} \\
&= \log|\mathcal{S}| - \sum_{y \in \operatorname{Supp}(D)} \Pr[D=y] \cdot \log \tfrac{1}{\Pr[D=y]} \\
&= \log|\mathcal{S}| - \sum_{0 \le i \le m} \\
&\qquad \sum_{y \in \operatorname{Supp}(D) :\, \left\lfloor \log_{1+\varepsilon} \frac{1}{\Pr[D=y]} \right\rfloor = i} \Pr[D=y] \cdot \log \tfrac{1}{\Pr[D=y]} \\
&\approx \log|\mathcal{S}| - \sum_{0 \le i \le m} h^f(i) \cdot \frac{i}{m},
\end{aligned}$$

where the quality of $\approx$ is a function of ε, and $\operatorname{Supp}(D) := \{y \mid \Pr[D=y] > 0\}$. Given an histogram h such that $\mathsf{W1}(h^f,h)$ is small, the following says that the estimate $\mu' = \log|\mathcal{S}| - \sum_{0 \le i \le m} h(i) \cdot i$ is close to $\mu(D)$.

$$\begin{aligned}
|\mu(D) - \mu'| &\approx \frac{1}{m} \cdot \Big| \sum_{0 \le i \le m} (h^f(i) - h(i)) \cdot i \Big| \\
&= \frac{1}{m} \cdot \Big| \sum_{0 \le i \le m} \sum_{0 \le j \le i} (h^f(i) - h(i)) \Big| \\
&\le \frac{1}{m} \cdot \sum_{0 \le i \le m} \Big| \sum_{0 \le j \le i} (h^f(i) - h(i)) \Big| \\
&= \mathsf{W1}(h^f,h).
\end{aligned}$$

2) Verifying histograms: The above tells us that in order to find the mean $\mu(D)$, it suffices to give a protocol that verifies that a histogram h is close to the true histogram h^f of D in W1 distance. Such protocol has to handle not only efficiently samplable distributions $D = f(U_n)$, but also the efficiently samplable distributions with post-selection $D = f(U_{\mathcal{S}})$, where $\mathcal{S}$ is efficiently decidable, as long as the verifier is also given a good approximation of $|\mathcal{S}|$. We prove the following:

Lemma II.2 (Verify Histogram Protocol, informal)**.** *There exists a constant-round public-coin protocol* VerifyHist*, between a prover in* $\mathbf{BPP^{NP}}$ *and an efficient verifier, where the parties get as a common input an efficiently decidable set* $\mathcal{S} \subseteq$

$\{0,1\}^n$*, an efficiently computable function* $f : \mathcal{S} \to \{0,1\}^*$*, a good approximation (i.e., within* $(1 \pm \frac{1}{\mathrm{poly}(n)})$ *factor) of* $|\mathcal{S}|$ *and a claimed* ε*-histogram* $h : \{0,\dots,m\} \mapsto [0,1]$ *of the distribution* $D = f(U_{\mathcal{S}})$*, and has the following guarantees:*

Completeness. If $h = h^f$*, then the verifier (when interacting with the honest prover) accepts with high probability.*

Soundness. If h *is far from* h^f *in the 1st Wasserstein distance (as a function of* ε*), then the verifier (when interacting with any cheating prover) rejects with high probability.*

a) Previous work using histograms.: Previous works have used histograms to estimate set sizes, and a related protocol to VerifyHist appears in Goldreich et al. [27]. We emphasize that their protocol accomplishes a different task that is incomparable to ours.[6]

3) Proving soundness of VerifyHist*.:* Before handling the general case, let us consider the very special case of VerifyHist where f that is promised to be a *regular* function over $\mathcal{S}$ (but with unknown regularity).

a) The case of regular functions.: Assuming that f is k regular, it implies that $|f(\mathcal{S})| = s/k$, and the only non-zero element of the histogram is $h(k/s)$, which has value 1. To verify a claimed value k', the verifier does the following.

Preimage test: The parties run the lower-bound protocol of [29] to verify that $k = |f^{-1}(f(x))| \geq k'$ (here x is an arbitrary element in $\mathcal{S}$).

Image test: The parties run the lower-bound protocol to check that $s/k = |f(\mathcal{S})| \geq s/k'$.

From the guarantee of the lower-bound protocol, it follows that the preimage test prevents the prover from claiming $k' \gg k$. Similarly, the image test prevents the prover from claiming $k' \ll k$, as this would make $|f(\mathcal{S})| = s/k \ll s/k'$ and the lower-bound of the image test would fail. Note that we are able to use the lower-bound protocol in the image test, since $f(U_{\mathcal{S}})$ is efficiently decidable.

b) The general case.: The idea in the regular case above is that by giving a lower bound on the image of the function f, one obtains an upper bound on the preimage. This idea extends to f that are far from being regular, and we generalize the special case of regular functions to a protocol with more image tests over many *subsets* of $f(\mathcal{S})$.

Define the sets $\mathcal{W}_i \subseteq f(\mathcal{S})$ given by $\mathcal{W}_i = \{y \in f(\mathcal{S}) \colon |f^{-1}(y)| \geq i\}$. As in the case of regular functions, where the regularity k determines the size of the image $|f(\mathcal{S})|$, for the general case we observe the histogram h^f determines the sizes of $|\mathcal{W}_i|$ for all i. Let w_i^h be the estimate of $|\mathcal{W}_i|$ that is given by the histogram h. Note that using the lower-bound protocol, one can efficiently verify membership in $\mathcal{W}_i$ (given y, run the lower-bound to verify that $|f^{-1}(y)| \geq i$). Therefore, the set sizes $|\mathcal{W}_i|$ can themselves be lower-bounded using (a generalization of) the lower-bound protocol of [29].

In our VerifyHist protocol, given an input $(\mathcal{S}, f, s)$ and a claimed ε-histogram h for $D = f(U_{\mathcal{S}})$, the verifier first checks that h is a valid histogram (i.e., $\sum_{0 \leq j \leq m} h(j) = 1$), and then the parties are engaged in the following two steps:

Preimage test:

1) The parties run the uniform sampling protocol of [27], to sample ℓ random elements $y_1,\dots,y_\ell$ from $D = f(U_{\mathcal{S}})$.
2) The prover sends $s_1 = \left|f^{-1}(y_1)\right|,\dots,s_\ell = \left|f^{-1}(y_\ell)\right|$ to the verifier.
3) The parties run in the set lower-bound protocol of [29] to verify that $|f^{-1}(y_i)| \geq s_i$ for all i.
4) The verifier constructs the histogram h^{emp} induced by $s_1,\dots,s_\ell$, and aborts if $\mathsf{W1}(h, h^{\mathsf{emp}})$ is too large.

Image test: For $i \in [m]$, the parties run the lower-bound protocol to verify that $|\mathcal{W}_i| \geq w_i^h$.

In the following we assume that the verifier did not reject, and we deduce that h must be close to the true histogram h^f in the 1st Wasserstein distance. As in the case of regular functions explained above, the preimage test prevents the prover from claiming that preimages are significantly larger than they actually are (otherwise, $\mathsf{W1}(h, h^{\mathsf{emp}})$ would be large). This yields that the following holds (ignoring small terms) for every $i \in [m]$:

$$a_i^f := \sum_{0 \leq j \leq i}^{i} h^f(j) \geq a_i := \sum_{0 \leq j \leq i} h(j) \tag{1}$$

The above yields that

$$\sum_{0 \leq i \leq m \colon a_i^f > a_i} (a_i^f - a_i) = \mathsf{W1}(h^f, h),$$

and in particular there exists an index $i^* \in \{0,\dots,m\}$ such that $a_{i^*}^f - a_{i^*} \geq \mathsf{W1}(h^f, h)/m$.[7] Since h is a valid histogram, it holds that $\sum_{0 \leq j \leq m} h(j) = 1 = \sum_{0 \leq j \leq m} h^f(j)$, which together with the above equation yield that

$$\sum_{i^* < j \leq m} (h(j) - h^f(j)) \geq \mathsf{W1}(h^f, h)/m \tag{2}$$

Let h' be an ε-histogram of D and let $0 \leq lb \leq ub \leq m$. Note that $\sum_{lb \leq j \leq up} h'(j) \cdot 2^{j\varepsilon}$ is, essentially, the number according to h' of the elements $y \in \mathrm{Supp}(D)$ with $\log_{1+\varepsilon} \frac{1}{\Pr_D(y)} \in [lb, ub)$. It follows that $\sum_{lb \leq j \leq up} (h^f(j) - h(j)) \cdot 2^{j\varepsilon}$ measures the *difference* between the number of elements in this range according to these two histograms. Equation 1 yields that the following holds.

Claim II.3. *For every* $0 \leq i \leq m$ *it holds that* $\sum_{0 \leq j \leq i} (h^f(j) - h(j)) \cdot 2^{j\varepsilon} \leq 2^{i \cdot \varepsilon} \cdot (a_i^f - a_i)$.

[6] The protocol of [27] lower bounds the size of a set $\mathcal{S}$ that is efficiently verifiable via a low-communication *interactive protocol*, but not efficiently decidable using a circuit. To do so, they recursively refine the histogram such that the following holds: if the prover lies about $|\mathcal{S}|$ (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. The prover, however, must claim they are non-empty in order to be consistent with previous answers.

[7] Throughout this informal presentation we are using rather rough bounds. For tighter analysis, see the formal proof in **??**.

Proof Sketch. By "pushing" the weights in h^f as far up towards the i'th "bin" while maintaining the invariant of Equation 1, we derive an histogram $\widetilde{h^f}$ such that:

1) $\sum_{0\leq j\leq i} h^f(j)\cdot 2^{j\varepsilon} \leq \sum_{0\leq j\leq i} \widetilde{h^f}(j)\cdot 2^{j\varepsilon}$, and
2) $\sum_{0\leq j\leq i}(\widetilde{h^f} - h(j))\cdot 2^{j\varepsilon} = 2^{i\cdot\varepsilon}\cdot(a_i^f - a_i)$.

Hence, $\sum_{0\leq j\leq i}(h^f(j) - h(j))\cdot 2^{j\varepsilon} \leq \sum_{0\leq j\leq i}(\widetilde{h^f} - h(j))\cdot 2^{j\varepsilon} = 2^{i\cdot\varepsilon}\cdot(a_i^f - a_i)$.

□

It follows that

$$\begin{aligned}
&|\mathcal{W}_m| - w_m^h \\
&= \sum_{0\leq j\leq m}(h^f(j) - h(j))\cdot 2^{j\varepsilon} \\
&= \sum_{0\leq j\leq i^*}(h^f(j) - h(j))\cdot 2^{j\varepsilon} + \sum_{i^*<j\leq m}(h^f(j) - h(j))\cdot 2^{j\varepsilon} \\
&\leq 2^{i^*\cdot\varepsilon}\cdot(a_{i^*}^f - a_{i^*}) + \sum_{i^*<j\leq m}(h^f(j) - h(j))\cdot 2^{j\varepsilon} \\
&\leq 2^{i^*\cdot\varepsilon}\cdot \mathsf{W1}(h^f,h)/m + \sum_{i^*<j\leq m}(h^f(j) - h(j))\cdot 2^{j\varepsilon}.
\end{aligned}$$

In addition, the "Image test" of the protocol yields (for the right choice of parameters) that $w_m^h - |\mathcal{W}_m| \leq \delta\cdot|\mathcal{W}_m| \leq \delta\cdot 2^{m\varepsilon}$, where $\delta > 1/\operatorname{poly}(n)$ is to be determined below. Hence,

$$\sum_{i^*<j\leq m}(h(j) - h^f(j))\cdot 2^{(j-i^*)\varepsilon} \leq 2^{(m-i^*)\varepsilon}\cdot\delta + \mathsf{W1}(h^f,h)/m \tag{3}$$

In the following we assume without loss of generality that $h(m) - h^f(m) \geq \mathsf{W1}(h^f,h)/m^3$ (otherwise, we carry the same calculation for the maximal $m' < m$ with this property) and assume towards a contradiction that $\mathsf{W1}(h^f,h) \in \Omega(\varepsilon)$. By setting δ sufficiently small (e.g., $O(\Delta\varepsilon/m^4)$), Equation 3 yields that $\sum_{i^*<j\leq m}(h(j) - h^f(j)) < \Delta/m$ in contradiction to Equation 2.

Acknowledgment

The authors would like to thank Boaz Barak, Thomas Holenstein and Salil Vadhan for very useful discussions. The third author also thanks the other authors of [30] for fruitful discussions and their perspectives on the power of Sam.

References

[1] Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Jan. 1979.

[2] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991.

[3] M. Ajtai. The worst-case behavior of schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *STOC '03*, 2003.

[4] A. Akavia, O. Goldreich, S. Goldwasser, and D. 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.

[5] Babai, Fortnow, and Lund. Non-deterministic exponential time has two-prover interactive protocols. *CM-PCMPL: Computational Complexity*, 1, 1991.

[6] B. Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–115, 2001.

[7] M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1987.

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

[9] A. Bogdanov and L. Trevisan. Average-case complexity. *CoRR*, 2006.

[10] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

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

[12] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[13] J. Buchmann, J. Loho, and J. Zayer. An implementation of the general number field sieve. In *CRYPTO '93*, 1994.

[14] I. Damgård. Collision free hash functions and public key signature schemes. In *Advances in Cryptology – EUROCRYPT '87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216. Springer, 1987.

[15] I. Damgård, O. Goldreich, T. Okamoto, and A. Wigderson. Honest verifier vs. dishonest verifier in public coin zero-knowledge proofs. In *Advances in Cryptology – CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 325–338. Springer, 1995.

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

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

[18] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

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

[20] L. Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989.

[21] O. Goldreich and S. Goldwasser. On the possibility of basing cryptography on the assumption that P $\neq$ NP.

Theory of Cryptography Library: Record 98-05, February 1998.

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

[23] O. Goldreich and S. P. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *IEEE Conference on Computational Complexity*, pages 54–73. IEEE Computer Society, 1999.

[24] O. Goldreich, Y. Mansour, and M. Sipser. Interactive proof systems: Provers that never fail and random selection. *FOCS*, 0:449–461, 1987. ISSN 0272-5428.

[25] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[26] O. Goldreich, A. Sahai, and S. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *In Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, 1998.

[27] O. Goldreich, S. Vadhan, and A. Wigderson. On interactive proofs with a laconic prover. In *Proc.* 28*th ICALP*, pages 334–345, 2001.

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

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

[30] S. D. Gordon, H. Wee, A. Yerukhimovich, and D. Xiao. On the round complexity of zero-knowledge proofs from one-way permutations, 2009. Manuscript. Available at http://www.cs.princeton.edu/~dxiao/docs/zk-owp.pdf.

[31] I. Haitner, J. J. Hoch, O. Reingold, and G. 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.

[32] I. Haitner, M. Nguyen, S. J. Ong, O. Reingold, and S. Vadhan. Statistically-hiding commitments and statistical zero-knowledge arguments from any one-way function. November 2007.

[33] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, 2009.

[34] I. Haitner, A. Rosen, and R. Shaltiel. On the (im)possibility of arthur-merlin witness hiding protocols. In *Theory of Cryptography, Fourth Theory of Cryptography Conference, TCC 2009*, 2009.

[35] I. Haitner, M. Mahmoody, and D. Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. Technical Report TR10-001, Electronic Colloquium on Computational Complexity, 2010.

[36] I. Haitner, O. Reingold, and S. Vadhan. Improvements in constructions of pseudorandom generators from one-way functions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, 2010.

[37] Y. Han, L. A. Hemaspaandra, and T. Thierauf. Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997. ISSN 0097-5397.

[38] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[39] T. Holenstein. Private communication. 2009.

[40] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, 2006.

[41] R. Impagliazzo and M. 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.

[42] Y. Ishai, E. Kushilevitz, and R. Ostrovsky. Sufficient conditions for collision-resistant hashing. In *In Proceedings of the 2nd Theory of Cryptography Conference*, pages 445–456, 2005.

[43] L. V. Kantorovich. On the translocation of masses. *Doklady Akademii Nauk SSSR*, 37:227–229, 1942.

[44] L. V. Kantorovich and G. S. Rubinstein. On a space of totally additive functions. *Vestn Lening. Univ*, 13(7): 52–59, 1958.

[45] Y. Lindell. Parallel coin-tossing and constant-round secure two-party computation. *JCRYPTOLOGY*, 2003.

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

[47] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS '04*, 2004.

[48] G. Monge. Mmoire sur la thorie des dblais et des remblais. *Histoire de l'Acadmie des Sciences de Paris*, page 666, 1781.

[49] M. Naor and M. 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.

[50] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998.

[51] T. Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000.

[52] S. J. Ong and S. P. Vadhan. An equivalence between zero knowledge and commitments. In *TCC*, pages 482–500, 2008.

[53] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings*

of the 2nd Israel Symposium on Theory of Computing Systems, pages 3–17. IEEE Computer Society, 1993.

[54] R. 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.

[55] A. Pavan, A. L. Selman, S. Sengupta, and N. V. Vinodchandran. Polylogarithmic-round interactive proofs for conp collapse the exponential hierarchy. *Theor. Comput. Sci.*, 385(1-3):167–178, 2007. ISSN 0304-3975.

[56] O. Reingold, L. Trevisan, and S. 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.

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

[58] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[59] A. Sahai and S. Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2): 196–249, 2003.

[60] S. Sanghvi and S. P. Vadhan. The round complexity of two-party random selection. In *STOC*, pages 338–347, 2005.

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

[62] D. 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.

[63] L. G. Valiant and V. V. Vazirani. Np is as easy as detecting unique solutions. In *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, 1985.

[64] H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *TCC '07*, pages 419–433, 2007.

[65] D. Xiao. (Nearly) optimal black-box constructions of commitments secure against selective opening attacks, 2009. Manuscript.

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