Electronic Colloquium on Computational Complexity, Report No. 205 (2023)

(Inefficient Prover) ZAPs from Hard-to-Invert Functions

Marshall Ball[1] and Dana Dachman-Soled[2⋆]

[1] New York University
[2] University of Maryland

Abstract. A ZAP is a witness-indistinguishable two-message public-coin interactive proof with the following simple structure: the verifier sends a uniformly random string, the prover responds, and the verifier decides in polynomial time whether to accept or reject.
We show that one-way functions imply the existence of ZAPs for NP where the prover runs in time 2^{n^ϵ} for arbitrarily small constant $\epsilon > 0$ (where n denotes the length on the NP instance). Moreover, it suffices to simply assume there exist functions that are hard to invert, but valid image/preimage pairs can be efficiently recognized. Such functions need not be efficiently computable and hence are not known to imply even one-way functions. Prior to this work such ZAPs were only known from one-way permutations [Ball et al., CRYPTO'20].

⋆ supported in part by NSF grants CNS-2154705 and CNS-1933033.

ISSN 1433-8092

1 Introduction

Since shortly after the notion of zero-knowledge interactive proofs was introduced [13], a goal has been constructing such protocols with the *minimal round-complexity* necessary under the *minimal assumptions* for every language in NP. Ideally, such a protocol would be completely non-interactive, consisting of a single message from the prover to the verifier. Or, barring that, a protocol that is comprised of a single round of interaction (two messages): one message from the verifier and a single response from the prover.

Unfortunately, it was discovered relatively early on that non-trivial zero-knowledge is unachievable with such limited interaction patterns [12]. While non-interactive zero-knowledge is possible in the presence of a trusted setup (public common reference string, or CRS), such trust assumptions are not always feasible. Fortunately, weaker notions of "zero-knowledge" exist that do not suffer from such limitations.

A *witness-indistinguishable* (WI) proof for an NP language has the property that the verifier cannot distinguish which witness was used to generate the proof. In particular, for any two witnesses for the NP language, w_0, w_1, it should be hard to distinguish the case that the transcript was generated from witness w_0 or from witness w_1. This notion is weaker than zero-knowledge and is sometimes even trivial: for example, if the language has unique witnesses. Nonetheless, witness-indistinguishability has many applications (such as constructing multi-theorem NIZKs, witness-hiding proofs, deniable authentication, and more), has desirable properties not inherent in zero-knowledge proofs (such as security under parallel composition), and is interesting in its own right. WI proofs are often used as a critical building block in zero-knowledge proofs with robust properties, as well as other relaxations of zero-knowledge such as witness-hiding proofs.

Witness-indistinguishability avoids the round-complexity barriers for zero-knowledge. *ZAPs*, introduced by Dwork and Naor [9], are two message public coin witness indistinguishable proofs. Dwork and Naor showed ZAPs for NP could be constructed assuming (enhanced) trapdoor permutations, or more generally efficient-prover non-interactive zero-knowledge for NP (NIZK) in the *uniform random string* (URS) model. Barak, Ong, and Vadhan [6] showed that fully non-interactive witness-indistinguishable proofs (NIWI), consisting of only a single message and no setup, follow from ZAPs and certain derandomization assumptions.

Towards understanding the minimal assumptions necessary for (non-trivial) ZAPs, Ball et al. [5] showed that one-way permutations suffice to construct ZAPs for NP with provers that run in subexponential time. (Note that NIWI/ZAP is trivial to achieve if the prover can run in exponential time: the prover that simply outputs the lexicographically first witness is witness indistinguishable. However, witness indistinguishability becomes nontrivial if the prover is forced to run in arbitrarily small subexponential time, assuming the subexponential-hardness of NP.) Moreover, they additionally observed that applying Barak, Ong, Vadhan's compiler [6] in this setting yields a NIWI with subexponential time prover if one additionally makes derandomization assumptions.

However, it was not clear whether one-way permutations were required to construct such inefficient prover ZAPs. In this work, we show that not only do one-way functions suffice to construct such ZAPs, but an even weaker notion suffices: something we call "hard-to-invert functions." These are functions that are hard to efficiently invert but are not necessarily easy to compute. Instead, it is only required that one can efficiently recognize the set of valid preimage-image pairs.

1.1 Our Results

Our final result for ZAPs is proved by first constructing an Offline-NIZK (oNIZK) in the URS model from hard-to-invert functions, and then applying a transformation of [5] that converts oNIZK with inefficient provers to ZAPs. We therefore begin by introducing the notion of hard-to-invert functions and reviewing the notion of oNIZK introduced in [5].

Hard-to-Invert Functions. We introduce a relaxation of one-way functions we call *hard-to-invert functions*, or HIF. These are functions, f, such that it is infeasible to find a pre-image of $f(\mathcal{U}_\kappa)$ with non-negligible probability, and yet one can efficiently recognize pre-image/image pairs (with a proof). In otherwords, f is hard to invert and $\{(x, f(x)\} \in \mathsf{NP}$. Unlike one-way functions, f need not be efficiently computable. See Definition 2.1 for more details.

HIFs follow from assumptions not known to imply OWFs. For example, if there is a problem P in TFUP (TFNP with unique witnesses) that is hard-on-average relative to the uniform distribution, then HIFs

exist (and moreover *injective* HIF exist).[3] Rosen, Segev, and Shahaf showed that there can be no black-box construction of a one-way function from the average-case hardness of a particular problem in TFUP [16].

We note that we believe HIF to actually be a more robust assumption than average-case hardness of TFUP, because HIF are not required to be injective. In fact, a main technical contribution of our work is to deal with precisely the case where images of the HIF can have different numbers of preimages. It is unclear if HIF are implied by the average-case hardness of TFNP, or NP for that matter.

A somewhat related notion of "quasi-one-way functions" was introduced by Bogdanov et al. [7], however, our definition differs from theirs in a variety of aspects. Most importantly: quasi-one-way functions have a much weaker restriction on the verifier (allowing it to accept strings that are not necessarily preimage/image pairs, so long as it remains hard to find a "preimage" that it will accept with a random image).[4] This condition means that $\{(x, f(x)\}$ may not be in NP if f is quasi-one-way, whereas this must be the case if f is hard-to-invert.

Offline NIZK (oNIZK). The standard notion of zero knowledge in the non-interactive setting requires the existence of a single simulator Sim, that for any statement $x \in \mathcal{L}$ produces a distribution over CRS's and proofs (CRS', π') that is computationally indistinguishable from honest CRS's and proofs (CRS, π). In contrast, the Offline Non-Interactive Zero-Knowledge (NIZK) notion requires existence of a distribution $\mathcal{D}_{\mathsf{Sim}}$ over small circuit simulators Sim, such that for any statement $x \in \mathcal{L}$, the distribution over (CRS', π') obtained by drawing Sim from $\mathcal{D}_{\mathsf{Sim}}$ and outputting $(\mathsf{CRS}', \pi') \leftarrow \mathsf{Sim}(x)$ is computationally indistinguishable from honest CRS's and proofs (CRS, π). Note that this definition does not trivialize the zero knowledge property, since the draw of Sim from $\mathcal{D}_{\mathsf{Sim}}$ is independent of the statement x. Another way to view this definition, is that we allow the simulator Sim to perform an expensive "offline" pre-processing step that is independent of the statement x (corresponding to the draw of the circuit from the potentially inefficiently samplable distribution $\mathcal{D}_{\mathsf{Sim}}$).

Our construction of oNIZK in the URS model consists of two steps. In the first step we obtain an oNIZK in which the Verifier requires non-uniform advice about the hard-to-invert function $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$. (Note that even if the algorithm verifying the hard-to-invert function f is uniform, the Verifier still requires this non-uniform advice which provides statistical information about the distribution of $f(U_\kappa)$.) Specifically, we show:

Theorem 1.1 (Informal). *Assume the existence of a hard-to-invert function verifiable by a non-uniform family of poly-sized circuits. There exists an oNIZK proof system in the URS model with* non-uniform *verifiers and prover complexity* 2^κ *for all of* NP.

In the second step, we remove the requirement for non-uniform advice by the Verifier. Specifically, as long as the hard-to-invert function f is itself uniformly verifiable, the Verifier can leverage the inefficient prover to learn the required information about $f(U_\kappa)$. Specifically, we obtain the following:

Theorem 1.2 (Informal). *Assume hard-to-invert functions exist. There exists an oNIZK proof system in the URS model with* uniform *verifiers and prover complexity* $\mathsf{poly}(\kappa) \cdot 2^\kappa$ *for all of* NP.

Finally, applying a theorem of [5], which transforms oNIZK in the URS model with inefficient provers to ZAPs, we obtain the following:

Corollary 1.3 (Informal)**.** *Assume hard-to-invert functions exist. There exists a ZAP system with* uniform *verifiers and prover complexity* $\mathsf{poly} \cdot 2^\kappa$ *for all languages in* NP.

[3] Consider $f(x, r) = (x, y)$ where $y = (r \oplus w,)$ where w is the unique witness such that $V(x, w) = 1$ (and V verifies the hard problem $P \in \mathsf{TFUP}$). Given $(x, r), (x, y)$ one can easily verify that $f(x, r) = (x, y)$, by checking $V(x, y \oplus r) = 1$. Thus, if one can invert the function, one can solve the search problem.

[4] Three more differences: (1) Quasi-one-way functions have the fine-grained promise that they can be computed in polynomial in the forward direction and security holds only for probabilistic adversaries running in (smaller) fixed polynomial time. (2) Bogdanov et al. allow quasi-one-way functions to be randomized. Unlike OWF, it is not clear how to derandomize quasi-one-way functions unconditionally. However, Bogdanov et al. observe that they can be derandomized assuming incompressible functions [8,2]. (3) Some of their techniques do not extend to security against non-uniform algorithms. Our reductions are non-uniform and hence we require security against non-uniform adversaries.

1.2 Technical Overview

Our main technical contribution is the construction of a new primitive *Verifiable Hardcore of a Random String* (VHCR) from any hard-to-invert function (See Section 4 for the formal definition). A VHCR consists of a tuple $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$ of algorithms as well as a hardcore predicate $\mathsf{DecRand}$. Gen is a deterministic function that on input security parameter, outputs public parameters PP that in our construction will consist of statistics about the hard-to-invert function f. The public parameters PP will then be inputted to each of the other algorithms $\mathsf{Open}, \mathsf{Verify}$, and $\mathsf{DecRand}$. Since Gen is an inefficient algorithm, we will alternately view its output PP as non-uniform advice given to the poly-time verification algorithm Verify. We explain the functionality of the $\mathsf{Open}, \mathsf{Verify}$ and $\mathsf{DecRand}$ algorithms alongside a discussion of our target application for the VHCR primitive, which is oNIZK in the URS model.

At a high level, our oNIZK construction in the URS model (See Section 5 for our oNIZK construction from VHCR) will proceed by splitting the URS ρ into blocks $[\rho_{i,j}]_{i\in[n'],j\in[n]}$. For each $i \in [n']$, we will use the sequences of bits $[\mathsf{DecRand}(\mathsf{PP}, \rho_{i,j})]_{j\in[n]}$ as the bits inputted to a hidden bits NIZK proof system $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$. To allow us to argue the zero knowledge property of the oNIZK construction, we therefore require that strings that decode to 0 or 1 under $\mathsf{DecRand}$ are indistinguishable to poly-time adversaries. This is what we mean by saying that $\mathsf{DecRand}$ is a hardcore predicate. The honest Prover will run $\mathsf{Open}(\mathsf{PP}, \rho_{i,j})$ to obtain an opening to a hidden bit that will convince the Verifier, running Verify, that the claimed bit is correct. Thus, we require that Open runs in time $\mathsf{poly}(\kappa) \cdot 2^{\kappa}$, that Verify runs in polynomial time, and we require the completeness property, which says that honestly opened bits can be correctly verified. On the other hand, in order for the oNIZK construction to be sound, we define a corresponding soundness property of the VHCR which says that for a given $\rho_{i,j}$ there should not exist two openings that are both accepted by the Verify algorithm, and that correspond to openings to the bits 0 and 1, respectively.

Given the above informal description of the functionality of the VHCR, we next describe the high level ideas of our construction of VHCR from any hard-to-invert function (See Section 4.1 for the formal construction). Let $f : \{0,1\}^{\kappa} \to \{0,1\}^{\kappa'}$ be a hard-to-invert function and let $H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}$ and $h : \{0,1\}^{\kappa} \to \{0,1\}^{m_2}$ be ℓ-wise independent hash functions (for sufficiently large but polynomial ℓ). H will be used to hash $y = f(U_\kappa)$ to obtain y' (which we will argue is close to uniformly distributed), while h will be used to hash the preimage of y' (i.e. the set $f^{-1}(H^{-1}(y'))$) down to a polynomial-size set. Let S_h be the set of $x \in \{0,1\}^{\kappa}$ such that $h(x) = 0^{m_2}$.

We begin by viewing the random input $\rho_{i,j}$ to $\mathsf{DecRand}$ as a sequence of uniformly random blocks of the form $(y', H, h, u) \sim \{0,1\}^{m_1+2\tilde{\kappa}+B\cdot\kappa}$. Consider the set of pre-images $S_{y',H,h} := (H \circ f)^{-1}(y', H, h) \cap S_h$ of size at most $B \in \mathsf{poly}(\kappa)$ (we show in Section 3.1, Claim 3.4 that a slight modification of $S_{y',H,h}$ has size at most B with all but negligible probability).

We first consider obtaining a hardcore predicate from a single random block of the form (y', H, h, u). We would like to show that if f is a hard-to-invert function, then when (y', H, h, u) is chosen uniformly at random, $\langle S_{y',H,h}, u\rangle$ is hardcore (see Section 3.2 and Lemma 3.6). By this notation, we mean to take a lexicographical ordering of the at most B elements of $S_{y',H,h}$ and view this as a $\{0,1\}$-vector of dimension $B \cdot \kappa$, and then take the inner product modulo 2 with vector u. By applying the Goldreich-Levin hardcore theorem, it is sufficient to argue that computing the set $S_{y',H,h}$ is hard for every non-uniform, polynomial time adversary A. To show this, we must create a reduction from inverting f to computing the set $S_{y',H,h}$. The natural thing for such a reduction to do is to obtain $y = f(x)$ for uniformly random x, choose H, h at random, and submit $(y' = H(y), H, h)$ to adversary A. If A is successful, it returns the set $S_{y',H,h}$. The hope is that $S_{y',H,h}$ contains at least one inverse of y–i.e. at least one x value such that $f(x) = y$.

For this approach to work, there are two main obstacles. First, the (y', H, h) submitted to A which is sampled as above may not be uniformly random, so the adversary A may fail on this distribution, while succeeding with respect to uniformly random (y', H, h).

Second, even though y is a preimage $H^{-1}(y')$, there is no guarantee that $y \in f(S_{y',H,h})$, since it is possible that none of the pre-images of y hash to 0^{m_2} under h, and therefore are not contained in $S_{y',H,h}$.

To overcome the first obstacle, we show that there exists a distribution $\mathcal{D}'$ such that any event that occurs relative to $\mathcal{D}'$ with probability p, occurs relative to $f(U_\kappa)$ with probability at least $p/2$. Further, when $y \sim \mathcal{D}'$, the distribution $(y' = H(y), H, h)$ has small *relative error* with respect to the uniform distribution

(See Section 3.1, Claim 3.1 and Lemma 3.2). This guarantees that the adversary $A((y' = H(y), H, h))$ succeeds with non-negligible probability when y is drawn from $\mathcal{D}'$, and therefore must also succeed with non-negligible probability when y is drawn from $f(U_\kappa)$.

To overcome the second obstacle, we show that for every y with pre-image size satisfying the following bounds: $\frac{2^\kappa}{2^{k+2}} \leq |f^{-1}(y)| \leq \frac{2^\kappa}{2^k}$, the pre-images of y make up a poly-sized fraction of the set $(H \circ f)^{-1}(H(y))$. Thus, by calibrating the setting of m_2, we obtain that for every such y, a pre-image of y hashes to 0^{m_2} under h and is therefore contained in the set $S_{y',H,h}$. Further, by our construction of $\mathcal{D}'$, elements y with pre-image sizes satisfying the above bounds must make up at least $1/\mathsf{poly}$-fraction of the weight of the distribution $\mathcal{D}'$. The above implies that if we first sample $(y', H, h) \sim (H(\mathcal{D}'), H, h) : H, h \sim U_{\tilde{\kappa}}$, run $A(y', H, h)$, and if A successfully returns $S_{y',H,h}$ (with non-negligible probability), *then* sample $y \sim \mathcal{D}'$ conditioned on (y', H, h), then a pre-image of the post-sampled y is contained in the returned set $S_{y',H,h}$ with non-negligible probability. Since this simply changes the order of sampling, it means that our suggested reduction which first samples $y \sim \mathcal{D}', H, h \sim U_{\tilde{\kappa}}$, runs $A(y', H, h) \to S$, checks whether $y \in f(S)$, and if yes, returns a pre-image x of y, also succeeds with non-negligible probability.

We would now like to view $(y', H, h, u) \sim \{0,1\}^{m_1+2\tilde{\kappa}+B\cdot\kappa}$. as a random string that decodes to $\langle S_{y',H,h}, u\rangle$ under the hardcore predicate. Recall that in order for this to be used to generate bits for the hidden bits proof system, we must produce $\mathsf{Open}, \mathsf{Verify}$ agorithms and argue their completeness and soundness properties, in addition to the "indistinguishability" property argued above. As a first attempt, we can have the Prover run an Open algorithm to prove a decoding is correct by producing all the preimages $x \in S_{y',H,h}$, and it can be checked in polynomial time by the Verifier running Verify that for all $x \in S_{y',H,h}$, $h(x) = 0^{m_2}$ and $H \circ f(x) = y'$. Clearly, the suggested decoding satisfies the completeness property, which says that a string that decodes to a bit b can be successfully opened to b. However, soundness, which says that a string that decodes to a bit b cannot also be successfully opened to $b' \neq b$ is problematic. Concretely, the Prover can produce as its opening a set S which omits elements from the true pre-image set $S_{y',H,h}$. Choosing *which* elements to omit allows the prover to control the value of the decoded bit $b' = \langle S, u\rangle$, and thus soundness is not achieved.

To achieve soundness, we draw inspiration from the techniques of Akavia et al [1], which make use of known statistics of a distribution to ensure the Prover's honest behavior. Instead of defining the hardcore predicate with respect to a single block (y', H, h, u) as above, we define $\mathsf{DecRand}$ to take as input a sequence of blocks $(y'_1, H_1, h_1, u_1), \ldots, (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$. The Open algorithm will now be required to provide openings for each of the blocks in the sequence. Further, the Verifier running Verify will be given statistics about the expected size of $S_{y',H,h}$ via the public parameters PP and it will check that the Prover's reported sizes in the opening deviate by at most a small amount from the expectation. This will ensure that the Prover can only lie about a small number of the sizes of the pre-images of $(y'_1, H_1, h_1, u_1), \ldots, (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$, which means it can lie only about a small fraction of the hardcore bits $b'_1, \ldots, b'_{t'}$. We then use the *majority* of $b'_1, \ldots, b'_{t'}$ as the final decoded bit, and we prove that the majority of a series of hardcore bits is itself a hardcore bit. Intuitively, because the majority function is "stable," the Prover should not be able to flip a decoding from 0 to 1 by lying about only a small fraction of the hardcore bits $b'_1, \ldots, b'_{t'}$. Thus, our final $\mathsf{DecRand}$ algorithm takes as input $(y'_1, H_1, h_1, u_1), \ldots, (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$, computes the hardcore bits $b_1 = \langle S_{y'_1,H_1,h_1}, u_1\rangle, \ldots, b_{t'} = \langle S_{y'_{t'},H_{t'},h_{t'}}, u_{t'}\rangle$ and outputs $\mathsf{Maj}(b_1, \ldots, b_{t'})$. This ensures that the Prover can only produce an opening that flips the decoded bit corresponding to a sequence of blocks $(y'_1, H_1, h_1, u_1), \ldots, (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$ with small $1/\mathsf{poly}$ probability. Setting $1/\mathsf{poly}$ to be sufficiently smaller than the number of hidden bits needed for a single NIZK proof in the hidden bits model, we have that the probability that the Prover can flip any of the hidden bits in a single proof is at most $1/3$, so we achieve soundness of $1/3$ (which can then be boosted to $\mathsf{negl}(\kappa)$ via parallel repetition).

A caveat is that the size of $S_{y',H,h}$ can actually be very high with low $1/\mathsf{poly}$, probability, thus skewing the statistics and allowing an adversary to lie about too many openings. We therefore fix a threshold P such that the size of $S_{y',H,h}$ is at most P with $1 - 1/\mathsf{poly}$ probability (See Section 3.1, Corollary 3.3). Given a sequence of blocks $(y'_1, H_1, h_1, u_1), \ldots, (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$ the Open algorithm proceeds in two stages: In the first stage, the Prover running the Open algorithm specifies which blocks have pre-image size greater than P (and provides at least $P + 1$ pre-images in this case). After this stage, t blocks remain. Those blocks

are opened as above. The Verifier running Verify is given statistics via the public parameters PP on (1) the fraction μ' of blocks with pre-image size greater than P and (2) the expected pre-image size μ for blocks with pre-image size at most P. Knowledge of these statistics allows us to guarantee (1) that t is a sufficiently large fraction $(1 - 1/\mathsf{poly})$ of t', and (2) that $\mathsf{MAJ}(b'_1, \ldots, b'_t)$ can deviate by adding or flipping a small number of bits from the true decoding $\mathsf{MAJ}(b_1, \ldots, b_{\hat{t}})$, where $\hat{t}$ is the true number of blocks with per-image size at most P. Thus, we obtain the same guarantee as above that the Prover can only flip a decoded bit from 0 to 1 or vice versa with small $1/\mathsf{poly}$ probability.

Note that the runtime of the Prover is in $\mathsf{poly}(t', \kappa) \cdot T$, where T is the time it takes to produce the set $S_{y',H,h}$ for a single block (y, H, h, u). Moreover, producing $S_{y',H,h}$ takes at most time 2^κ, since in the worst case, one can loop over every possible $x \in \{0,1\}^\kappa$ and check in polynomial time whether $x \in S_{y',H,h}$, which occurs if and only if $H \circ f(x) = y' \wedge h(x) = 0^{m_2}$.

The drawback of the above protocol is that the Verifier requires non-uniform advice of μ', μ. In fact, the Verifier requires additional non-uniform advice about the distribution $\mathcal{D}'$ and its distance from $f(U_\kappa)$ in order to know the setting of the parameters m_1, m_2. We next show that the Verifier can actually use the Prover to provide these values, together with a proof. For purposes of this overview, we will just explain how the Verifier obtains μ', μ from the Prover. The other parameters are obtained in a similar way. (See Sections 6.1 and 6.2) We use a protocol from prior work [14] called $\mathsf{VerifyHist}$ to learn μ', μ. A $(\tilde{\epsilon}, \tilde{t})$ histogram of a probability distribution $H \circ f(U_\mathcal{R})$ is a vector $\mathbf{h}$ of dimension $\tilde{t}$ such that for $i \in [\tilde{t}]$, $h[p] = Pr_{y \sim f(U_\mathcal{R})}[2^{-(i+1)\cdot\tilde{\epsilon}} \geq \Pr_{f(U_\mathcal{R})}[y] < 2^{-i\cdot\tilde{\epsilon}}]$. $\mathsf{VerifyHist}$ allows a Prover to prove that a submitted $(\tilde{\epsilon}, \tilde{t})$-histogram $\mathbf{h}$ is $20/\tilde{t}$ to the true $(\tilde{\epsilon}, \tilde{t})$-histogram $\mathbf{h}^*$ with respect to the 1st Wasserstein distance. Given $\mathbf{h}$, μ' can be computed as $1 - \sum_{i \geq (-\log(P)+\kappa+2\tilde{\kappa}-m_2)/\tilde{\epsilon}} h[i] \cdot \frac{2^{i\cdot\tilde{\epsilon}}}{2^{m_1+2\tilde{\kappa}}}$ while μ can be computed as $\mu = \sum_{i > (-\log(P)+\kappa+2\tilde{\kappa}-m_2)/\tilde{\epsilon}} h[i] \cdot \frac{2^\kappa}{2^{m_1+m_2}}$. We prove that as long as $\mathbf{h}$ and $\mathbf{h}^*$ have 1st Wasserstein distance of at most $20/\tilde{t}$ (see Definition 2.15), then the computed μ', μ are $1/t'$ and $1/t$ additive approximations of the correct statistics $(\mu')^*, \mu^*$. Similar to above, we note that the prover runtime in the $\mathsf{VerifyHist}$ protocol is in $\mathsf{poly}(\tilde{t}, \kappa) \cdot T$, where T is the time it takes to produce the set $S_{y',H,h}$.

At the end of this stage, we obtain oNIZK for NP in the URS model from hard-to-invert function with prover complexity $\mathsf{poly}(\kappa) \cdot 2^\kappa$. We then apply a transformation of [5] to obtain ZAPS for NP from hard-to-invert function with prover complexity 2^κ.

1.3 Open Questions

A number of questions remain in this space. For example, are hard-to-invert functions necessary for subexponential prover ZAPs? Are hard-to-invert functions implied by the average-case hardness of TFNP?

While we know that OWFs are necessary and sufficient for (inefficient-prover) NIZK in the presence of a structured public random string (CRS), it is not known if OWF suffice to construct NIZK in the presence of a uniform random string (URS). The simplest assumption known to imply this object is OWP.

NIZKs in the URS model imply (inefficient prover) ZAPs, so in some sense, our present result is a necessary step on the way to NIZK in the URS model from OWF. That said, the difficulty in extending our result to NIZK is that it is not clear how to simulate the information sent by the prover in our protocol. While it is easy to efficiently simulate a single preimage/image pair, our protocol (roughly) requires the prover to send the verifier all preimages of a given image. It is not clear how to efficiently sample an image with all its preimages (unless one has a OWP).

Recently, Ghosal et al. [10] showed how to construct (inefficient prover) NIZK in the Random Oracle (RO) model. One might hope to adapt their techniques to constructing NIZK in the URS model, but their proof critically relies simultaneously upon the pseudorandomness of the RO and density of the RO. This suggests that constructing an entropy-preserving PRG would suffice, but unfortunately constructing such objects from OWF remains beyond our current techniques.

2 Definitions and Preliminaries

Notation. For a set S, U_S denotes the uniform distribution over S. For a positive integer z, U_z denotes the uniform distribution over bitstrings of length z. For $n \in \mathbb{N}$ we let $(n) := \{0, 1, \ldots, n\}$ and $[n] := \{1, 2, \ldots, n\}$. If $\mathcal{D}$ is a probability distribution, "$x \in \mathcal{D}$" denotes that x is in the support of $\mathcal{D}$.

Definition 2.1 (Hard-to-Invert Function (HIF))**.** *A function* $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$ *is* hard-to-invert *if*

- *There exists a nondeterministic polynomial time algorithm* V *such that* $V(x,y) = 1$ *if and only if* $f(x) = y$.
  (Or equivalently, there exists a polynomial $p(n)$ *and a deterministic verifier* V' *such that there exists* $\pi_{x,y} \in \{0,1\}^{p(|x|+|y|)}$ *where* $V'(x,y,\pi_{x,y}) = 1$ *if and only if* $f(x) = y$.*)*
- *For every non-uniform PPT algorithm* A*, there exists a negligible function* $\epsilon(\kappa)$ *s.t.* $\forall\kappa$*, we have* $\Pr[A(f(U_\kappa)) \in f^{-1}(f(U_\kappa))] \in \epsilon(\kappa)$.

We say a function f *a* non-uniform *hard-to-invert function if item 1 above holds for non-uniform poly-time* V*, and item 2 holds.*

Remark 1. In our constructions, the prover will need to be able to evaluate $f(x)$ on all inputs x in time 2^κ, where f is a hard-to-invert function. Note that we may assume this WLOG given the above definition. Specifically, for a given f, the prover can simply enumerate over all $x \in \{0,1\}^\kappa, y \in \{0,1\}^{\kappa'}$ and check whether $V(x,y) = 1$ for each pair. If the verifier runs in nondeterministic time κ'', then this takes time $2^{\kappa+\kappa'+\kappa''}$. We can then create a new hard-to-invert function f' with input length $\kappa''' = \kappa + \kappa' + \kappa'''$ which evaluates f on the first κ bits (and ignores the rest). In this case, the prover's run time will be $2^{\kappa'''}$–where κ''' is the input length of f'–as desired.

Definition 2.2 (One-Way Function (OWF))**.** *A function* $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$ *is* one-way *if* f *is computable in polynomial time and it is hard-to-invert (Definition 2.1).*

We say a f *is* non-uniform one-way *if it is computable in non-uniform polynomial time and it is non-uniform hard-to-invert.*

The following three definitions follow the exposition in [2].

Definition 2.3 (ℓ-wise Independent Hashing)**.** *A collection of functions* $\mathcal{H} = \{H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}\}$ *is called an* ℓ*-wise independent hash function if for every* ℓ *distinct strings* $x_1, \ldots, x_\ell$ *the random variables* $H(x_1), \ldots, H(x_\ell)$ *induced by a uniformly random choice of* $h \sim H$ *are uniformly distributed in* $(\{0,1\}^{m_1})^\ell$.

Definition 2.4 (Statistical Distance with Relative-Error)**.** *We say that a distribution* Z *on* $\{0,1\}^{m_1}$ *is* ϵ*-close to uniform with relative error if for every event* $A \subseteq \{0,1\}^{m_1}$, $|\Pr[Z \in A] - \frac{|A|}{2^{m_1}}| \leq \epsilon \cdot \frac{|A|}{2^{m_1}}$.

Definition 2.5 (Relative-Error Extractor)**.** *A function* $E : \{0,1\}^{\tilde{\kappa}} \times \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}$ *is a* (k,ϵ)*-relative error extractor if for every source* X *with min-entropy* $H_\infty(X) \geq k$*, with all but negligible probability over* $r \sim U_{\tilde{\kappa}}$*,* $E(r,X)$ *is* ϵ*-close to uniform with relative error.*

The following definition and theorem follow the exposition in [11].

Definition 2.6 (Proof Systems in the Hidden Bits Model)**.** *A pair of probabilistic machines,* (P,V)*, is called a* hidden-bits proof system for L *if* V *is polynomial-time and the following two conditions hold*

- ***Completeness:*** *For every* $x \in L$

$$\Pr[V(x, R_I, I, \pi) = 1] \geq \frac{2}{3},$$

  where $(I,\pi) := P(x,R)$*,* R *is a random variable uniformly distributed in* $\{0,1\}^{\mathsf{poly}(|x|)}$ *and* R_I *is the sequence of bits at positions* $I \subseteq \{1,2,\ldots,\mathsf{poly}(|x|)\}$*. That is,* $R_I = r_{i_1} \cdots r_{i_n}$*, where* $R = r_1 \cdots r_n$ *and* $I = (i_1, \ldots, i_n)$.
- ***Soundness:*** *For ever* $x \notin L$ *and every machine* B*,*

$$\Pr[V(x, R_I, I, \pi) = 1] \leq \frac{1}{3},$$

  where $(I,\pi) := B(x,R)$*,* R *is a random variable uniformly distributed in* $\{0,1\}^{\mathsf{poly}(|x|)}$ *and* R_I *is the sequence of bits at positions* $I \subseteq \{1,2,\ldots,\mathsf{poly}(|x|)\}$.

– ***Zero Knowledge:*** *There exists a probabilistic polynomial-time algorithm* Sim *such that the ensembles* $\{(x, R_I, P(x,R))\}_{x\in L}$ *and* $\{\mathsf{Sim}(x)\}_{x\in L}$ *are statistically indistinguishable, where* R *is a random variable uniformly distributed in* $\{0,1\}^{\mathsf{poly}(|x|)}$ *and* R_I *is the sequence of bits at positions* $I \subseteq \{1,2,\ldots,\mathsf{poly}(|x|)\}$*, and where* $(I,\pi) = P(x,R)$*.*

Theorem 2.7. *There exists a (perfect) zero-knowledge Hidden Bits proof system for Graph Hamiltonicity with perfect completeness. Furthermore, the prover may be implemented by a polynomial-time machine which gets a Hamiltonian cycle as auxiliary input.*

Definition 2.8 (Non-Interactive Proofs in the URS Model)**.** *A pair of algorithms* $(\mathsf{Prover}, \mathsf{Verifier})$ *is called a non-interactive proof system in the URS model for a language* $\mathcal{L}$ *if the algorithm* $\mathsf{Verifier}$ *is deterministic polynomial-time, there exists a polynomial* $p(\cdot)$ *and a negligible function* $\mu(\cdot)$ *such that the following two conditions hold:*

– ***Completeness:*** *For every* $x \in \mathcal{L}$

$$\Pr[\mathsf{CRS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi) = 1] \geq 1 - \mu(|x|).$$

– ***Soundness:*** *For every* $x \notin \mathcal{L}$*, every algorithm* P^*

$$\Pr[\mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi' \leftarrow P^*(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi') = 1] \leq \mu(|x|).$$

Definition 2.9 (Non-Interactive Zero-Knowledge with Offline Simulation (oNIZK) in the URS Model)**.** *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a non-interactive proof system in the URS model for the language* $\mathcal{L}$*. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is* non-adaptively zero-knowledge with offline simulation *in the URS model if there exists a distribution* $\mathcal{D}_{\mathsf{Sim}}$ *over polynomial-sized circuits* Sim *such that the following two distribution ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of* $|x|$*)*

$$\{(\mathsf{CRS}, \pi) : \mathsf{CRS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{CRS}, x)\}_{x\in\mathcal{L}}$$
$$\{(\mathsf{CRS}', \pi') \leftarrow \mathsf{Sim}(x) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}\}_{x\in\mathcal{L}}.$$

A useful property of oNIZK is the following: Let $\mathcal{D}_{\mathsf{yes}}$ be a distribution over statements $x \in \mathcal{L}$ and let $\mathcal{D}_{\mathsf{no}}$ be a distribution over statements $x \in \overline{\mathcal{L}}$. If $\mathcal{D}_{\mathsf{yes}}$ and $\mathcal{D}_{\mathsf{no}}$ are computationally indistinguishable by polynomial-sized circuits then the following two distribution ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of $|x|$)

$$\{(x, (\mathsf{CRS}, \pi) \leftarrow \mathsf{Sim}(x)) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}, x \leftarrow \mathcal{D}_{\mathsf{yes}}\}$$
$$\{(x', (\mathsf{CRS}', \pi') \leftarrow \mathsf{Sim}(x')) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}, x' \leftarrow \mathcal{D}_{\mathsf{no}}\}.$$

Definition 2.10 (AM Protocol)**.** *A proof system* $\langle P, V\rangle$ *for a language* L *is called an* AM Protocol *if* V *is a polynomial-time algorithm and the protocol consists of 2-messages and is public-coin. Specifically, the first message is sent from* V *to* P *and consists of a sequence of random coins. The second message is sent from* P *to* V*.*

Definition 2.11 (ZAP)**.** *A ZAP is a 2-round (2-message) protocol for proving membership of* $x \in \mathcal{L}$*, where* $\mathcal{L}$ *is a language in* NP*. Let the first-round (verifier to prover) message be denoted* ρ *and the second-round (prover to verifier) response be denoted* π *satisfying the following conditions:*

– ***Public Coins:*** *There is a polynomial* $p(\cdot)$ *such that the first round messages form a distribution on strings of length* $p(|x|)$*. The verifier's decision whether to accept or reject is a polynomial time function of* $x, \rho,$ *and* π *only.*
– ***Completeness:*** *Given* x*, a witness* $w \in w(x)$*, and a first-round* ρ*, the prover generates a proof* π *that will be accepted by the verifier with overwhelming probability over the choices made by the prover and the verifier.*

– ***Soundness:*** *With overwhelming probability over the choice of ρ, there exists no $x' \notin \mathcal{L}$ and second round message π such that the verifier accepts (x', ρ, π).*
– ***Witness-Indistinguishability:*** *Let $w, w' \in w(x)$ for $x \in L$. Then $\forall \rho$, the distribution on π when the prover has input (x, w) and the distribution on π when the prover has input (x, w') are nonuniform probabilistic polynomial time (in $|x|$) indistinguishable, even given both witnesses w, w'.*

Theorem 2.12. *Assume $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ is a non-adaptive oNIZK proof system for language $\mathcal{L}$ with prover complexity $\mathsf{poly}(\kappa) \cdot 2^{\kappa}$ in the URS model. Then there exists a ZAP for language $\mathcal{L}$ with prover complexity $\mathsf{poly}(\kappa) \cdot 2^{\kappa}$.*

The following refinement of Babai and Moran [3] (additionally bounding the growth of prover complexity) is due to the fact that their compiler amounts to parallel composition (on the part of the prover).

Theorem 2.13 (Public-Coin Round Reduction (Babai-Moran [3])). *Let $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle$ be a constant-round, public-coin interactive proof system for a promise problem Π with prover complexity $T(n)$, then there exists 2 message public-coin AM proof system for Π with prover complexity $\mathsf{poly}(n)T(n)$.*

We give the definitions of histograms and Wasserstein distance as given in [14] and [15]. The histogram of a probability distribution $f(U_{\mathcal{R}})$ is a function $h : [0,1] \to [0,1]$ such that $h(p) = Pr_{y \sim f(U_{\mathcal{R}})}[\Pr_{f(U_{\mathcal{R}})}[y] = p]$. The following definition describes a discretized version of this concept.

Definition 2.14 ($(\tilde{\epsilon}, \tilde{t})$-histogram)**.** *Let $f(U_{\mathcal{R}})$ be a probability distribution where $\mathcal{R} \subseteq \{0,1\}^n$ and $f : \{0,1\}^n \to \{0,1\}^{n'}$. Fix $\tilde{\epsilon} > 0$ and $\tilde{t} = \lceil \frac{n}{\tilde{\epsilon}} \rceil$. For $i \in [\tilde{t}]$ we define the i-th interval A_i and the i-th bucket B_i as*

$$A_i := (2^{-(i+1)\tilde{\epsilon}}, 2^{-i\tilde{\epsilon}}], B_i := y : \frac{|f^{-1}(y)|}{|\mathcal{R}|} \in A_i.$$

We then let $h := (h_0, \ldots, h_{\tilde{t}})$, where $h_i := Pr_{y \sim f(U_{\mathcal{R}})}[y \in B_i] = \sum_{y \in B_i}[\Pr_{f(U_{\mathcal{R}})}[y]]$. The tuple h is called the $(\tilde{\epsilon}, \tilde{t})$-histogram of $f(U_{\mathcal{R}})$.

Definition 2.15 (1st Wasserstein Distance over arrays)**.** *Given two distribution vectors $\mathbf{x}$ and $\mathbf{y}$ over $[\tilde{t}]$ we let $a_i = \sum_{j \in (i)} x_j$ and $b_i = \sum_{j \in (i)} y_j$. We let*

$$\overrightarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y}) := \frac{1}{\tilde{t}} \sum_{i \in (\tilde{t}) : a_i > b_i} (a_i - b_i), \qquad \overleftarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y}) := \frac{1}{\tilde{t}} \sum_{i \in (\tilde{t}) : b_i > a_i} (b_i - a_i),$$

and $\mathsf{W1}(\mathbf{x}, \mathbf{y}) := \overrightarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y}) + \overleftarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y})$. $\mathsf{W1}(\mathbf{x}, \mathbf{y})$ is called the 1st Wasserstein distance between $\mathbf{x}$ and $\mathbf{y}$. $\overrightarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y})$ and $\overleftarrow{\mathsf{W1}}(\mathbf{x}, \mathbf{y})$ are called the right and left Wasserstein distance, respectively

Claim 2.16. Let h, h^* be two distribution vectors over $\tilde{t} = n/\tilde{\epsilon}$ and let $\Delta \in [\lfloor 1/\tilde{\epsilon} \rfloor]$. Let $\gamma, \gamma' \in \{0, \ldots, n-1\}$ and $\gamma' > \gamma$. Assume h, h^* have Wasserstein distance at most d.

1. If $\sum_{i \leq \gamma/\tilde{\epsilon}} h[i] \leq \alpha$ then $\sum_{i \leq \gamma/\tilde{\epsilon} - \Delta} h^*[i] \leq \alpha + d \cdot \tilde{t}/\Delta$.
   Equivalently, if $\sum_{i \leq \gamma/\tilde{\epsilon} - \Delta} h[i] \geq \alpha$ then $\sum_{i \leq \gamma/\tilde{\epsilon}} h^*[i] \geq \alpha - d \cdot \tilde{t}/\Delta$.
2. If $\sum_{\gamma/\tilde{\epsilon} - \Delta \leq i \leq \gamma'/\tilde{\epsilon} + \Delta} h[i] \leq \alpha$ then $\sum_{\gamma/\tilde{\epsilon} \leq i \leq \gamma'/\tilde{\epsilon}} h^*[i] \leq \alpha + d \cdot \tilde{t}/\Delta$.
   Equivalently, if $\sum_{\gamma/\tilde{\epsilon} \leq i \leq \gamma'/\tilde{\epsilon}} h[i] \geq \alpha$ then $\sum_{\gamma/\tilde{\epsilon} - \Delta \leq i \leq \gamma'/\tilde{\epsilon} + \Delta} h^*[i] \geq \alpha - d \cdot \tilde{t}/\Delta$.

Proof. Assume that h, h^* have Wasserstein distance at most d. For $i \in [\tilde{t}]$, let $a_i = \sum_{j \in [i]} h[j]$ and $b_i = \sum_{j \in [i]} h^*[j]$.

For item (1): Assume towards contradiction that $\sum_{i \leq \gamma/\tilde{\epsilon}} h[i] \leq \alpha$ but $\sum_{i \leq \gamma/\tilde{\epsilon} - \Delta} h^*[i] > \alpha + d \cdot \tilde{t}/\Delta$. Then for $i \in \{\gamma/\tilde{\epsilon} - \Delta, \ldots, \gamma/\tilde{\epsilon}\}$, $a_i > \alpha + d \cdot \tilde{t}/\Delta$ and $b_i \leq \alpha$. Thus, the contribution to the Wasserstein distance

on this interval is at least

$$\begin{aligned}\frac{1}{\tilde{t}}\sum_{i\in\{\gamma/\tilde{\epsilon}-\Delta,\ldots,\gamma/\tilde{\epsilon}\}} a_i - b_i &> \frac{1}{\tilde{t}}\sum_{i\in\{\gamma/\tilde{\epsilon}-\Delta,\ldots,\gamma/\tilde{\epsilon}\}} d\cdot\tilde{t}/\Delta \\ &> \frac{1}{\tilde{t}}\cdot\Delta\cdot d\cdot\tilde{t}/\Delta \\ &> d,\end{aligned}$$

which is a contradiction to the assumption on the Wasserstein distance of h and h^*.

For item (2): Assume towards contradiction that $B := \sum_{\gamma/\tilde{\epsilon}-\Delta\le i\le\gamma'/\tilde{\epsilon}+\Delta} h[i] \le \alpha$ but $B^* := \sum_{\gamma/\tilde{\epsilon}\le i\le\gamma'/\tilde{\epsilon}} h^*[i] > \alpha + d\cdot\tilde{t}/\Delta$.

Let $A := \sum_{i<\gamma/\tilde{\epsilon}-\Delta} h[i]$, $A^* := \sum_{i<\gamma/\tilde{\epsilon}} h^*[i]$. $C := 1-A-B = \sum_{i>\gamma'/\tilde{\epsilon}+\Delta} h[i]$, $C^* := 1-A^*-B^* = \sum_{i>\gamma'/\tilde{\epsilon}} h^*[i]$. Note that the assumption above implies that $(A-A^*)+(C-C^*) > d\cdot\tilde{t}/\Delta$. However, by (1), $(A-A^*)\le d\cdot\tilde{t}/\Delta$. and $(C-C^*) = (A^*+B^*)-(A+B) = \sum_{i\le\gamma'/\tilde{\epsilon}} h^*[i] - \sum_{i\le\gamma'/\tilde{\epsilon}+\Delta} h[i] \le d\cdot\tilde{t}/\Delta$. So the assumption implies that both $(A-A^*)$ and $(C-C^*)$ must be positive. This means that for $i\in\{\gamma/\tilde{\epsilon}-\Delta-1,\ldots,\gamma/\tilde{\epsilon}-1\}$ $a_i - b_i \ge (A-A^*)$ and for $i\in\{\gamma'/\tilde{\epsilon},\ldots,\gamma'/\tilde{\epsilon}+\Delta\}$ $b_i - a_i \ge (C-C^*)$. Thus, the total contribution to the Wasserstein distance over the intervals $\{\gamma/\tilde{\epsilon}-\Delta-1,\ldots,\gamma/\tilde{\epsilon}-1\}$ and $\{\gamma'/\tilde{\epsilon},\ldots,\gamma'/\tilde{\epsilon}+\Delta\}$ is at least $((A-A^*)+(C-C^*))\cdot\Delta/\tilde{t}$. Thus, $d\ge((A-A^*)+(C-C^*))\cdot\Delta/\tilde{t}$. On the other hand, we have that $((A-A^*)+(C-C^*)) > d\cdot\tilde{t}/\Delta$. This implies $d>d$, which leads to contradiction. □

Lemma 2.17 (VerifyHist Protocol, [14])**.** *For a fixed distribution $f(U_{\mathcal{R}})$ where $\mathcal{R}\subseteq\{0,1\}^n$ and $f:\{0,1\}^n\to\{0,1\}^{n'}$, let h^* denote its $(\tilde{\epsilon},\tilde{t})$-histogram, where $\tilde{t}=\lceil\frac{n}{\tilde{\epsilon}}\rceil$. We define the promise problem $\Pi^{\mathsf{VerifyHist}}$ as*

$$\begin{aligned}\Pi_Y^{\mathsf{VerifyHist}} &:= \{(f,S,\tilde{\epsilon},h) : h = h^*\}\\ \Pi_N^{\mathsf{VerifyHist}} &:= \{(f,S,\tilde{\epsilon},h) : \mathsf{W1}(h,h^*) > 20/\tilde{t}\}.\end{aligned}$$

There exists a constant-round public-coin interactive proof for $\Pi^{\mathsf{VerifyHist}}$ with completeness $1-2^{-n}$ and soundness 2^{-n}, where the verifier runs in time $\mathsf{poly}((T_1+T_2)/\tilde{\epsilon})$, where T_1 is the time, given (x,y) to verify that $x\in\mathcal{R}$ and $y=f(x)$, and T_2 is the time needed to sample from $U_{\mathcal{R}}$.

2.1 Statistical Lemma

We will be using the following extension of a Lemma from Akavia et al. [1]:

Lemma 2.18. *Let $\hat{S}\subseteq[t']$ be a set of size $\hat{t}\le t$. For $i\in\hat{S}$, let $n_i\in(M)$ be random variables of expectation $\mu\pm\frac{1}{t}$. Let $S\subseteq[t']$ be a set of size t such that $\hat{S}\subseteq S$. For $i\in S$, let $s_i\in(M)$ be integers. Further, $s_i\le n_i$ for all $i\in\hat{S}$. Then, $\forall\delta>0$ w.p. at least $1-\delta$, it holds that: $\mathsf{Avg}_{i\in\hat{S}}[n_i]\in[\mu-\frac{2M}{t\sqrt{\delta}}-\frac{1}{t},\mu+\frac{2M}{t\sqrt{\delta}}+\frac{1}{t}]$. If $\mathsf{Avg}_{i\in\hat{S}}[n_i]\le\mu+\frac{2M}{t\sqrt{\delta}}+\frac{1}{t}$ and $\mathsf{Avg}[s_i]\ge\mu-\frac{2M}{t\sqrt{\delta}}-\frac{1}{t}$, then the size of the set $|\{i\in\hat{S}:s_i\ne n_i\}|$ is at most $\frac{4M}{\sqrt{\delta}}+2+(t-\hat{t})\cdot M$ (where $\mathsf{Avg}[s_i]:=\frac{1}{t}\sum_{i\in S}s_i$).*

Proof. The first part of the Lemma follows immediately from Chebyshev's inequality and the fact that the variance of the n_i variables is at most M^2.

The second part of the Lemma is similar to Lemma 8 in [1], except for the additional sets $\hat{S},S$. We show that if $\mathsf{Avg}_{i\in\hat{S}}[n_i]\le\mu+\frac{2M}{t\sqrt{\delta}}+\frac{1}{t}$ and $\mathsf{Avg}[s_i]\ge\mu-\frac{2M}{t\sqrt{\delta}}-\frac{1}{t}$, then $s_i\ne n_i$ for at most $\frac{4M}{\sqrt{\delta}}+2+(t-\hat{t})\cdot M$ number of indeces $i\in\hat{S}$.

Assume towards contradiction that $s_i\ne n_i$ for more than $\frac{4M}{\sqrt{\delta}}+2+(t-\hat{t})\cdot M$ number of the indeces $i\in\hat{S}$. Since s_i is integer and $s_i\le n_i$, this means that $s_i\le n_i-1$ for all such i. Further, assume that $\mathsf{Avg}[s_i]\ge\mu-\frac{2M}{t\sqrt{\delta}}-\frac{2}{t}$. Then $\sum_{i\in S}s_i < (t-\hat{t})\cdot M+\sum_{i\in\hat{S}}n_i-\frac{4M}{\sqrt{\delta}}-2-(t-\hat{t})\cdot M$. Further, by assumption

$\sum_{i\in\hat{S}} n_i \le (\mu + \frac{1}{\hat{t}})\cdot\hat{t} + \frac{2M}{\sqrt{\delta}}$. This means that

$$\begin{aligned}\sum_{i\in S} s_i &< (t-\hat{t})\cdot M + \sum_{i\in\hat{S}} n_i - \frac{4M}{\sqrt{\delta}} - 2 - (t-\hat{t})\cdot M \\ &\le (t-\hat{t})\cdot M + (\mu + \frac{1}{\hat{t}})\cdot\hat{t} + \frac{2M}{\sqrt{\delta}} - \frac{4M}{\sqrt{\delta}} - 2 - (t-\hat{t})\cdot M \\ &\le \mu\cdot\hat{t} - \frac{2M}{\sqrt{\delta}} - 1\end{aligned}$$

Since $\hat{t} \le t$, his implies that $\mathsf{Avg}[s_i] < \mu - \frac{2M}{t\sqrt{\delta}} - \frac{1}{t}$, which is a contradiction. □

3 Set Size Bounds and a Hard Problem

We first prove size bounds on the sets $(H\circ f)^{-1}(y')$, where f is a hard-to-invert function and H is an ℓ-wise independent hash, $(H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2})$, where h is an ℓ-wise independent hash function, as well as a few other variants in Section 3.1. We then use these size bounds to prove the hardness of a particular search problem with respect to a uniformly random string in Section 3.2. We further consider a hardcore predicate with respect to a uniformly random string based on this search problem in Section 3.2.

3.1 Set Size Bounds

We begin by defining the following distribution which will be helpful for proving the statistical guarantees on the pre-image sizes, as well as proving the hardness of the search problem in Section 3.2.

Claim 3.1. Let $f:\{0,1\}^\kappa \to \{0,1\}^{\kappa'}$. There exists a distribution $\mathcal{D}'$ that is $3\kappa/p$-statistically close to $f(U_\kappa)$ and an integer k such that:

1. $\mathcal{D}'$ has min-entropy k.
2. At least $1/p$ of the weight of $\mathcal{D}'$ is on elements y such that $\Pr_{\mathcal{D}'}[y] \ge \frac{1}{2^{k+2}}$. Let $\mathcal{S}$ be the set of elements y in the support of $\mathcal{D}'$ such that $k \le -\log(\Pr_{\mathcal{D}'}[y]) \le k+2$
3. The support of $\mathcal{D}'$ (which we also denote by $\mathcal{D}'$) is a subset of the support of $f(U_\kappa)$.
4. For all $y\in\mathcal{D}'$, $\Pr_{\mathcal{D}'}[y] = \frac{|f^{-1}(y)|}{|\mathcal{D}'|}$.

Proof. We consider repeating the following procedure until items (1) and (2) above are satisfied:

1. Initialize $\mathcal{D}' := f(U_\kappa)$.
2. While less than $1/p$ of the weight of $\mathcal{D}'$ is on elements y such that $\Pr_{\mathcal{D}'}[y] \ge \frac{1}{2^{k+2}}$, where k is the min-entropy of $\mathcal{D}'$:
   (a) Remove all elements in $\mathcal{D}'$ such that $\Pr_{\mathcal{D}'}[y] \ge \frac{1}{2^{k+2}}$.
   (b) Re-normalize the distribution $\mathcal{D}'$.
   (c) Update $\mathcal{D}'$ to be this new distribution.

Note that the above loop can repeat at most κ times since $\frac{1}{2^{k+2}}\cdot\frac{1}{1-1/p} \le \frac{1}{2^{k+1}}$ when $1/p \le 1/2$. Thus, the min-entropy of $\mathcal{D}'$ must go up by at least 1 in each iteration, and $\mathcal{D}'$ can have min-entropy of at most κ at any point in the procedure, since the support of $\mathcal{D}' \subseteq \{0,1\}^\kappa$ is an invariant.

Clearly, (1) and (2) hold when the loop terminates.

We next analyze the statistical distance. First note that $\Pr_{f(U_\kappa)}[\{0,1\}^\kappa \backslash \mathcal{D}']$ is at most $1-(1-1/p)^\kappa \le \kappa/p$. Further, for every $y\in\mathcal{D}'$,

$$1 \le \frac{\Pr_{\mathcal{D}'}[y]}{\Pr_{f(U_\kappa)}[y]} \le \frac{1}{(1-1/p)^\kappa} \le 1 - \frac{2\kappa}{p},$$

when $\kappa/p \le 1/2$. Thus, the total statistical distance is at most $\frac{3\kappa}{p}$.

(3) and (4) also clearly hold by construction. □

Remark 2. Note that when f is a hard-to-invert function, $\mathcal{D}'$ from Claim 3.1 must have min-entropy $k \in \omega(\log(\kappa))$, since otherwise $1/p$ of the weight of the distribution is on elements y in the image of f that occur with $2^{-O(\log(\kappa))} = 1/\mathsf{poly}(\kappa)$ probability. Such y can be inverted by sampling random x and checking whether $f(x) = y$.

In the following, let $\mathcal{D}'$ be *any* distribution of min-entropy at least $k-1$, which has statistical distance at most $4\kappa/p$ from $f(U_\kappa)$, for which $1/(2p)$ of the weight is on elements y such that $\Pr_{\mathcal{D}'}[y] \geq \frac{1}{2^{k+3}}$, and satisfies the remaining properties of Theorem 3.1.

Lemma 3.2. *Let $H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}$ be an ℓ-wise independent hash with $m_1 = k - \log(p) - \log(1/\epsilon)$, where $\epsilon = \epsilon(\kappa) = 1/\mathsf{poly}(\kappa)$. Then the following hold:*

1. *With all but negligible probability over choice of $H \sim \mathcal{H}_1$, for all $y' \in \{0,1\}^{m_1}$, $|(H \circ f)^{-1}(y') \cap f^{-1}(\mathcal{D}')| \in [\frac{2^\kappa}{2^{m_1-1}}, \frac{2^\kappa}{2^{m_1+1}}] = [\frac{2^\kappa}{2^k}\frac{p}{2\epsilon}, \frac{2^\kappa}{2^k}\frac{2p}{\epsilon}]$.*
2. *With all but $6\kappa/p$ probability over choice of $H \sim \mathcal{H}_1$, $y' \sim U_{m_1}$ $|(H \circ f)^{-1}(y')| \in [\frac{2^\kappa}{2^{m_1+2}}, \frac{2^\kappa}{2^{m_1-2}}] = [\frac{2^\kappa}{2^k} \cdot \frac{p}{4\epsilon}, \frac{2^\kappa}{2^k} \cdot \frac{4p}{\epsilon}]$.*
3. *With all but negligible probability over choice of $H \sim \mathcal{H}_1$, for all $y' \in \{0,1\}^{m_1}$, $|H^{-1}(y') \cap \mathcal{S}| \in [\frac{2^{k-1}}{p'2^{m_1}}, \frac{2^{k+1}}{p'2^{m_1}}] = [\frac{p}{2\epsilon p'}, \frac{2p}{\epsilon p'}]$.*
4. *With all but negligible probability over choice of $H \sim \mathcal{H}_1$, for all $y' \in \{0,1\}^{m_1}$, $|(H \circ f)^{-1}(y') \cap f^{-1}(\mathcal{S})| \in [\frac{2^\kappa}{2^{k+2}} \cdot \frac{p}{2\epsilon p'}, \frac{2^\kappa}{2^k} \cdot \frac{2p}{\epsilon p'}]$.*

Proof. Item (1) follows from the fact that ℓ-wise independent hash functions H are relative error extractors (see Def. 2.5), see [17], proof of Proposition A.1, and the fact that $\mathcal{D}'$ has min-entropy k, while H has output length $m_1 = k - \log(p) - \log(1/\epsilon)$. Item (2) follows from the fact that $f(U_\kappa)$ and $\mathcal{D}'$ are statistically $4\kappa/p$-close. Item (3) follows from Chernoff bounds with ℓ-wise independent random variables, such as Theorem 5 in [4]. Item (4) follows from Chernoff bounds with ℓ-wise independent random variables, such as Theorem 5 in [4] (as above), and from the definition of $\mathcal{S}$, which is the set of elements y in the support of $\mathcal{D}'$ such that $k \leq -\log(\Pr_{\mathcal{D}'}[y]) \leq k+2$. □

Corollary 3.3. *Let $H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}$ be an ℓ-wise independent hash with $m_1 = k - \log(p) - \log(1/\epsilon)$, where $\epsilon = \epsilon(\kappa) = 1/\mathsf{poly}(\kappa)$. Let $h : \{0,1\}^{\kappa} \to \{0,1\}^{m_2}$ be an ℓ-wise independent hash, where $m_2 = \kappa - k$. Then the following hold:*

1. *With all but negligible probability over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, for all $y' \in \{0,1\}^{m_1}$, $|(H \circ f)^{-1}(y') \cap f^{-1}(\mathcal{D}') \cap h^{-1}(0^{m_2})| \in [\frac{p}{4\epsilon}, \frac{4p}{\epsilon}]$.*
2. *With all but $4\kappa/p$ probability over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, $y' \sim U_{m_1}$ $|(H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| \in [\frac{p}{8\epsilon}, \frac{8p}{\epsilon}]$. Let $P := \frac{8p}{\epsilon}$.*
3. *With all but negligible probability over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, for all $y' \in \{0,1\}^{m_1}$, $|(H \circ f)^{-1}(y') \cap f^{-1}(\mathcal{S}) \cap h^{-1}(0^{m_2})| \in [\frac{p}{8\epsilon p'}, \frac{2p}{\epsilon p'}]$.*

Proof. Each of Items (1), (2), (3) above follows from Items (1), (2), (3) of Lemma 3.2, respectively, and Chernoff bounds with ℓ-wise independent random variables (e.g. Theorem 5 in [4]). □

Remark 3 (Parameters). Parameters k, P are defined above. m_1, m_2 are determined by k and the setting of parameters p, ϵ. Let $\mu' \leq 4\kappa/p$ be the probability over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, $y' \sim U_{m_1}$ that $|(H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| > P$. Let μ be the expected size of $|(H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2})|$ conditioned on $|(H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| \leq P$, where the expectation is taken over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, $y' \sim U_{m_1}$.

Given a set $\mathcal{Q} \subseteq \{0,1\}^\kappa$. Let $\mathsf{Truncate}_z(\mathcal{Q})$ denote the set that, for every y such that $f^{-1}(y) \cap \mathcal{Q} \neq \emptyset$, contains the lexicographically first z elements of $f^{-1}(y) \cap \mathcal{Q}$.

Claim 3.4. With all but negligible probability over H, h, y',

$$|\mathsf{Truncate}_{P+1}((H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2}))| \leq B = P/2 + (P+1) \cdot \kappa \cdot \frac{8}{\epsilon}.$$

Proof. Note that

$$|\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))| \le |(H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))\cap f^{-1}(\mathcal{D}')| + |\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))\cap f^{-1}(Im(f)\setminus\mathcal{D}')|.$$

By Property (1) of Corollary 3.3, $|(H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))\cap f^{-1}(\mathcal{D}')| \le P/2$ with all but negligible probability. Further,

$$|\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))\cap f^{-1}(Im(f)\setminus\mathcal{D}')| \le (P+1)|H^{-1}(y')\cap(Im(f)\setminus\mathcal{D}')|.$$

Note that by the definition of $\mathcal{D}'$, the loop can repeat at most κ times, and each time at most $2^{k+2}/p$ elements are removed. Thus, $|(Im(f)\setminus\mathcal{D}')| \le \kappa\cdot 2^{k+2}/p$. Thus, by the properties of the ℓ-wise independent hash H, with all but negligible probability over choice of H, for all y',

$$|H^{-1}(y')\cap(Im(f)\setminus\mathcal{D}')| \le 2\cdot\frac{\kappa\cdot 2^{k+2}}{p\cdot 2^{m_1}} = \kappa\cdot\frac{8}{\epsilon}.$$

□

The above implies that with all but negligible probability over H, h, y', the size of $\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))$ is polynomial in κ.

Claim 3.5. Let y^* be an arbitrary value such that $y^*\in H^{-1}(y')\cap\mathcal{S}$. Then with all but negligible probability over choice of H, $\Pr_{y\sim\mathcal{D}'|H(\mathcal{D}')=y'}[y=y^*] \ge \frac{2^\kappa}{2^{k+2}}\cdot\frac{2^{m_1}}{2\cdot 2^\kappa} = \frac{\epsilon}{8p}$.

Proof. By the fact that the ℓ-wise independent hash H is a relative error extractor, with all but negligible probability over choice of H, for all $y'\in\{0,1\}^{m_1}$, $\Pr_{y\sim\mathcal{D}'}[H(y)=y'] \le \frac{2}{2^{m_1}}$.

Further, since $y^*\in\mathcal{S}$, we have that $\Pr_{\mathcal{D}'}[y^*] \ge \frac{1}{2^{k+2}}$

Thus, we have that

$$\begin{aligned}\Pr_{y\sim\mathcal{D}'|H(\mathcal{D}')=y'}[y=y^*] &= \frac{\Pr_{\mathcal{D}'}[y^*]}{\Pr_{y\sim\mathcal{D}'}[H(y)=y']}\\ &\ge \frac{2^{m_1}}{8\cdot 2^k}\\ &= \frac{\epsilon}{8p}.\end{aligned}$$

□

3.2 The Hard Problem relative to the Uniform Distribution

Lemma 3.6. *Let $f:\{0,1\}^\kappa\to\{0,1\}^{\kappa'}$ be a hard-to-invert function and let $H:\{0,1\}^{\kappa'}\to\{0,1\}^{m_1}\sim\mathcal{H}, h:\{0,1\}^\kappa\to\{0,1\}^{m_2}\sim\mathcal{H}_2$ be ℓ-wise indpendent hash functions defined as above. For all non-uniform ppt adversaries A*

$$\Pr_{H\sim\mathcal{H},h\sim\mathcal{H}_2,y'\sim U_{m_1}}[A(y',H,h)=\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))]\in\mathsf{negl}(\kappa).$$

Proof. Assume towards contradiction that

$$\Pr_{H\sim\mathcal{H},h\sim\mathcal{H}_2,y'\sim U_{m_1}}[A(y',H,h)=\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))]=\rho(\kappa),$$

where $\rho(\kappa)$ is non-negligible.

We first note that item (1) of Lemma 3.2 implies that

$$\Pr_{\substack{H\sim\mathcal{H},h\sim\mathcal{H}_2,\\ y'\sim H(\mathcal{D}')}}[A(y',H,h)=\mathsf{Truncate}_{P+1}((H\circ f)^{-1}(y')\cap h^{-1}(0^{m_2}))]\ge\frac{\rho(\kappa)}{2}-\mathsf{negl}(\kappa). \tag{1}$$

By item (3) of Corollary 3.3, with all but negligible probability over choice of $H \sim \mathcal{H}_1, h \sim \mathcal{H}_2$, for all $y' \in \{0,1\}^{m_1}$, $|(H \circ f)^{-1}(y') \cap f^{-1}(\mathcal{S}) \cap h^{-1}(0^{m_2})| \in [\frac{p^2}{8\epsilon^2 p'}, \frac{8p^2}{\epsilon^2 p'}]$.

This implies that

$$\Pr_{\substack{H\sim\mathcal{H},h\sim\mathcal{H}_2,\\ y'\sim H(\mathcal{D}')}}[A(y',H,h) = \mathsf{Truncate}_{\frac{8p}{\epsilon}}((H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})) \\ \wedge \mathsf{Truncate}_{\frac{8p}{\epsilon}}((H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})) \cap f^{-1}(\mathcal{S}) \neq \emptyset] \geq \frac{\rho(\kappa)}{2} - \mathsf{negl}(\kappa). \quad (2)$$

Let x^* be the lexicographically first element contained in $\mathsf{Truncate}_{\frac{8p}{\epsilon}}((H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})) \cap f^{-1}(\mathcal{S})$, if the intersection is non-empty. Otherwise, x^* is equal to $\perp$. Let $y^* = f(x^*)$, if $x^* \neq \perp$ and let $y^* = \perp$ otherwise. Note that $y^* \in H^{-1}(y') \cap \mathcal{S}$, when $x^* \neq \perp$.

The above along with Claim 3.5 imply that

$$\Pr_{\substack{H\sim\mathcal{H},h\sim\mathcal{H}_2,\\ y'\sim H(\mathcal{D}'),y\sim\mathcal{D}'|H(\mathcal{D}')=y'}}[A(y',H,h) = \mathsf{Truncate}_{\frac{8p}{\epsilon}}((H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})) \wedge x^* \neq \perp \wedge y = y^*] \geq (\frac{\rho(\kappa)}{2} - \mathsf{negl}(\kappa)) \cdot \frac{\epsilon}{p}. \quad (3)$$

The above implies a hard-to-invert function adversary as follows: A' receives a $y \sim f(U_\kappa)$. A' guesses that y is in the support of $\mathcal{D}'$ (correct with probability at least $1 - \kappa/p$). In this case, y is distributed as a draw from $\mathcal{D}'$. A' chooses H, h, computes $y' = H(y)$, and runs $A(y', H, h)$. A outputs a set $\mathcal{S}'$ of polynomial size, which is equal to $\mathsf{Truncate}_{P+1}((H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2}))$ with probability at least $\rho(\kappa)/2 - \mathsf{negl}(\kappa)$. A' checks if $y \in f(\mathcal{S}')$. By (3), this occurs with probability at least $(\frac{\rho(\kappa)}{2} - \mathsf{negl}(\kappa)) \cdot \frac{\epsilon}{p}$. If yes, then one of the elements of $\mathcal{S}'$ is a pre-image of y. A' finds this element and outputs it.

The overall success probability of A' is

$$(1 - \kappa/p) \cdot (\frac{\rho(\kappa)}{2} - \mathsf{negl}(\kappa)) \cdot \frac{\epsilon}{p},$$

which is non-negligible. □

For a function $\psi(\cdot) : \mathbb{N} \to \mathbb{R}$, let $S_{\psi(\kappa)}$ be the set of $z \in [P]$ such that $\Pr_{H\sim\mathcal{H},h\sim\mathcal{H}_2,y'\sim U_{m_1}}[|(H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| = z] \geq \psi(\kappa)$.

Corollary 3.7. *Let $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$ be a hard-to-invert function and let $H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1} \sim \mathcal{H}, h : \{0,1\}^\kappa \to \{0,1\}^{m_2} \sim \mathcal{H}_2$ be ℓ-wise independent hash functions defined as above. For functions $\psi(\cdot)$, $\psi'(\cdot)$, every $z \in S_{\psi(\kappa)}$ and for all ppt adversaries A, if*

$$\Pr_{H\sim\mathcal{H},h\sim\mathcal{H}_2,y'\sim U_{m_1}}[A(y',H,h) = (H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2}) \mid |(H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| = z] \geq \psi'(\kappa),$$

then $\psi \cdot \psi'(\kappa)$ is negligible.

Given this, the following is an immediate corollary of the Goldreich Levin Algorithm [11]. This result is typically presented as a hardcore predicate for a one-way function, but the core algorithm can also be viewed as an efficient local list decoder for the Walsh-Hadamard Code. Because this list-decoder will output a polynomial-size list containing the correct input with non-negligible probability, simply guessing the correct item in the list will successfully output the correct input with non-negligible probability overall.

Corollary 3.8. *Let $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$ be a hard-to-invert function and let $H : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1} \sim \mathcal{H}, h : \{0,1\}^\kappa \to \{0,1\}^{m_2} \sim \mathcal{H}_2$ be ℓ-wise indpendent hash functions defined as above. For every $\psi(\cdot)$, $\hat{\psi}(\cdot)$, every $z \in S_{\psi(\kappa)}$, and for all ppt adversaries A, if*

$$\Pr_{H\sim\mathcal{H},h\sim\mathcal{H}_2,y'\sim U_{m_1},u\sim U_{\kappa\cdot P}}[A(y',H,h) = \langle (H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2}), u\rangle \mid |(H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2})| = z] \geq \frac{1}{2} + \hat{\psi}(\kappa),$$

then $\psi \cdot \hat{\psi}(\kappa)$ is negligible, whereby $\langle (H\circ f)^{-1}(y') \cap h^{-1}(0^{m_2}), u\rangle$ we denote the inner product of a lexicographic ordering of the set $(H \circ f)^{-1}(y') \cap h^{-1}(0^{m_2})$ and u.

4 Verifiable Hard Core of a Random String

We introduce a new primitive we call a verifiable hard core of a random string (VHCR). This can be thought of as a statistically binding commitment scheme for random values where most strings are valid commitments. While random strings must correspond to valid commitments, we conversely do not require that commitments to a particular value can be produced efficiently. Opening a random commitment obviously should not be efficient, but we simply require that it can be done not too inefficiently. Verifying an opening is still required to be in polynomial time.

Definition 4.1 (Verifiable Hard Core of a Random String)**.** *Let $\delta(\cdot) \in 1/\mathsf{Poly}$. A tuple of algorithms $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$, allow $\delta(\kappa)$-confidence verification of a hardcore predicate $\mathsf{DecRand}$ with efficiency $\mathsf{poly}(1/\delta(\kappa)) \cdot 2^{\kappa}$ if there exist polynomials $m(\cdot), m'(\cdot), m''(\cdot)$ such that $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$ and $\mathsf{DecRand}$ have the following syntax:*

- $\mathsf{Gen} : 1^{\kappa} \to \{0,1\}^{m''(\kappa)}$ *is a deterministic algorithm that, for each security parameter κ, outputs public parameters PP.*
- $\mathsf{Open} : \{0,1\}^{m''(\kappa)} \times \{0,1\}^{m(\kappa)} \to \{0,1\}^{m'(\kappa)}$ *takes as input the public parameters PP and x, and outputs π, and runs in time $\mathsf{poly}(1/\delta(\kappa)) \cdot 2^{\kappa}$.*
- $\mathsf{Verify} : \{0,1\}^{m''(\kappa)} \times \{0,1\}^{m(\kappa)} \times \{0,1\}^{m'(\kappa)} \to \{0,1,\perp\}$ *takes as input PP, x, π and outputs $0, 1$ or $\perp$, and runs in polynomial time in security parameter κ.*
- $\mathsf{DecRand} : \{0,1\}^{m''(\kappa)} \times \{0,1\}^{m(\kappa)} \to \{0,1\} \cup \{?\}$ *is a partial predicate that takes as input the public parameters PP and input x and outputs $0, 1$ or $?$.*

The following security properties are required for $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$ and $\mathsf{DecRand}$:

Completeness. *Completeness has two parts:*

- *Most strings are "openable" to a bit. We say a string $x \in \{0,1\}^m$ is openable if $\mathsf{DecRand}(\mathsf{PP}, x) \in \{0,1\}$. Otherwise, the string is said to be unopenable. A random string is openable with probability at least $1 - \delta$,*
$$\Pr_{x \sim \mathcal{U}_m}[\mathsf{DecRand}(\mathsf{PP}, x) \in \{0,1\}] \geq 1 - \delta.$$
- *"Openable" strings can be efficiently opened. For any openable $x \in \{0,1\}^m(n)$, $\mathsf{Verify}(\mathsf{PP}, x, \mathsf{Open}(\mathsf{PP}, x)) = \mathsf{DecRand}(\mathsf{PP}, x)$. If x is unopenable then $\mathsf{Verify}(\mathsf{PP}, x, \mathsf{Open}(\mathsf{PP}, x)) = \perp$.*

Soundness. *The predicate is statistically binding:*

$$\Pr_{x \sim \mathcal{U}_m}[\exists \pi_0, \pi_1 : \mathsf{Verify}(\mathsf{PP}, x, \pi_0) = 0 \wedge \mathsf{Verify}(\mathsf{PP}, x, \pi_1) = 1] \leq \delta$$

Unbiased. *Openable strings are unbiased:*

$$\Pr_{x \sim \mathcal{U}_m}[\mathsf{DecRand}(\mathsf{PP}, x) = 1 | \mathsf{DecRand}(\mathsf{PP}, x) \in \{0,1\}] = 1/2$$

Indistinguishability. *Openable strings are computationally hiding; For $\mathsf{PP} = \mathsf{Gen}(1^{\kappa})$, it is infeasible to distinguish between encodings of 0 and 1:*

$$(\mathsf{PP}, \mathcal{U} | \mathsf{DecRand}(\mathsf{PP}, \mathcal{U}) = 0) \approx (\mathsf{PP}, \mathcal{U} | \mathsf{DecRand}(\mathsf{PP}, \mathcal{U}) = 1)$$

4.1 Our Construction of VHCR from HIF

In this section we prove the following theorem.

Theorem 4.2. *If hard-to-invert functions exist, then for any $\delta(\cdot) \in 1/\mathsf{poly}$ there exists a partial predicate $\mathsf{DecRand}$ and a tuple of algorithms $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$, that allow $\delta(\kappa)$-confidence verification of $\mathsf{DecRand}$ with efficiency $\mathsf{poly}(1/\delta(\kappa)) \cdot 2^{\kappa}$.*

We begin by defining $\mathsf{Gen}, \mathsf{DecRand}, \mathsf{Open}, \mathsf{Verify}$ in Figure 4.1, and then proving the completeness, soundness, unbiased, and indistinguishability properties. Throughout this section we assume existence of a hard-to-invert function $f : \{0,1\}^\kappa \to \{0,1\}^{\kappa'}$ with a deterministic verifier V_f (that takes in x, y and τ, where τ is polynomial in $|x| + |y|$, such that $f(x) = y \iff \exists \tau, V_f(x, y, \tau) = 1$.

Figure 4.1: VHCR Construction

$\mathsf{Gen}(1^\kappa)$ By Claim 3.1, $f(U_\kappa)$ is at most $3\kappa/p$-far from a distribution $\mathcal{D}'$ with min-entropy k. k is used to set the parameters m_1, m_2. The threshold size P and statistics μ, μ' are defined as in Remark 3. Gen outputs $\mathsf{PP} = (m_1, m_2, P, \mu, \mu')$.

$\mathsf{DecRand}(\mathsf{PP}, x)$**:** Given a uniform random string x, we parse x as t' strings of the form $(y'_1, H_1, h_1, u_1), \ldots (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$, where for $i \in [t']$, $H_i : \{0,1\}^{\kappa'} \to \{0,1\}^{m_1}$ and $h_i : \{0,1\}^\kappa \to \{0,1\}^{m_2}$ are interpreted as the descriptions of ℓ-wise independent hash function, and u_i is a random string of length $P \cdot \kappa$.

- For all $i \in [t']$, define $S_{(y'_i, H_i, h_i)} := (H_i \circ f)^{-1}(y') \cap h_i^{-1}(0^{m_2})$.
- For all $i \in [t']$, $n'_i := 0$ if $|S_{(y'_i, H_i, h_i)}| > P$ and $n'_i := 0$ otherwise.
- Let $\hat{S}$ of size $\hat{t}$ be the set of indeces i such that $n'_i = 1$
- For all $j \in \hat{S}$, let $n_j := |S_{(y'_j, H_j, h_j)}|$.
- If $\mathsf{Avg}_{i \in [t']}[n'_i] \in [\mu' - \frac{2}{t'\sqrt{\delta}} - \frac{1}{t'}, \mu' + \frac{2}{t'\sqrt{\delta}} + \frac{1}{t'}$ and $\mathsf{Avg}_{j \in \hat{S}}[n_j] \in [\mu - \frac{2P}{t\sqrt{\delta}} - \frac{1}{t}, \mu - \frac{2P}{t\sqrt{\delta}} - \frac{1}{t}]$ then x is openable. Otherwise x is unopenable and $\mathsf{DecRand}(\mathsf{PP}, x)$ outputs ?.
- If x is openable let $b_j := \langle S_{(y'_i, H_i, h_i)}, u_j \rangle$, where the above denotes taking the dot product of u_j with the vector obtained from a lexicographic ordering of $S_{(y'_i, H_i, h_i)}$. $\mathsf{DecRand}(\mathsf{PP}, x)$ outputs $\mathsf{Maj}([b_j]_{j \in \hat{S}})$.

$\mathsf{Open}(\mathsf{PP}, x)$**:** We parse x as t' strings of the form $(y'_1, H_1, h_1, u_1), \ldots (y'_{t'}, H_{t'}, h_{t'}, u_{t'})$. If x is unopenable, the opening consists of $\perp$. Otherwise, the opening consists of the following:

- $[n'_i]_{i \in [t']}$, $[n_i]_{i \in \hat{S}}$, defined as above.
- At least $P + 1$ elements of the set $S_{(y'_i, H_i, h_i)}$ for each i such that $n'_i = 0$, and for all $j \in \hat{S}$, all elements of the set $S_{(y'_j, H_j, h_j)}$ of size n_j. For each such element $x \in S_{(y'_i, H_i, h_i)}$ sent, additionally include y, τ such that $f(x) = y$ and $V(x, y, \tau) = 1$

$\mathsf{Verify}(\mathsf{PP}, x, \pi)$**:**

- Parse π to learn $[s'_i]_{i \in [t']}$. Let S' be the set of i such that $s'_i = 1$.
- Parses π to learn $[s_i]_{i \in S'}$.
- Use the values μ', μ contained in PP to check that $\mathsf{Avg}[s'_i] \geq \mu' - \frac{2}{t'\sqrt{\delta}} - \frac{1}{t'}$ and $\mathsf{Avg}[s_i] \geq \mu - \frac{2P}{t\sqrt{\delta}} - \frac{1}{t}$.
- Use the value of P contained in PP to check that π contains lists of size at least $P + 1$ elements of the set $S_{(y'_i, H_i, h_i)}$ for each i such that $s'_i = 0$. π contains the n_j elements of the set $S_{(y'_j, H_j, h_j)}$ for each $j \in S'$.

4.2 Completeness and Soundness of the Construction

Lemma 4.3. $(\mathsf{Gen}, \mathsf{DecRand}, \mathsf{Open}, \mathsf{Verify})$ *as defined in Figure 4.1 has the following completeness property:*

- ***Completeness:*** *The probability that a uniformly random string x is openable is at least $1 - 2\tilde{\delta} \geq 1 - \delta$.*

The claim follows immediately from Lemma 2.18 and by setting $\tilde{\delta} \leq \delta/2$.

Towards proving soundness, we begin with the following claim:

Claim 4.4. If x is openable then For any π such that $\mathsf{Verify}(\mathsf{PP}, x, \pi) \in \{0,1\}$, we have that $|\{i \in [t'] : s'_i \neq n'_i\}| \leq \frac{4}{\sqrt{\delta}} + 2$ AND $|\{i \in \hat{S} : s_i \neq n_i\}| \leq \frac{8P}{\sqrt{\delta}} + 2P + 2$

Proof. We apply Lemma 2.18 twice. The first time, we have $n'_1, \ldots, n'_{t'} \in \{0,1\}$ indicating whether a block has more than P number of pre-images or at most P number of pre-images. In this case, we apply the lemma

with sets $\hat{S} = S = [t']$ and $M = 1$. Since we assume x is openable, applying Lemma 2.18, this means that $s'_i \neq n'_i$ for at most $\frac{4}{\sqrt{\tilde{\delta}}} + 2$ of the indeces $i \in [t']$.

Next, we will apply Lemma 2.18 with the sets $\hat{S}, S' \subseteq [t']$ and $M = P$. Recall that $\hat{S}$ is the set of indeces i such that $n'_i = 1$ (with size $\hat{t}$) and that S' is the set of indeces i such that $s'_i = 1$ (with size t). Further, note that by the above $(t - \hat{t}) \leq \frac{4}{\sqrt{\tilde{\delta}}} + 2$. Since we assume x is openable, applying Lemma 2.18, this means that $|\{i \in \hat{S} : s_i \neq n_i\}|$ is at most $\frac{4P}{\sqrt{\tilde{\delta}}} + 2 + (t - \hat{t}) \cdot P \leq \frac{8P}{\sqrt{\tilde{\delta}}} + 2P + 2$ □

Lemma 4.5. $(\mathsf{Gen}, \mathsf{DecRand}, \mathsf{Open}, \mathsf{Verify})$ *as defined in Figure 4.1 has the following soundness property:*

- ***Soundness:*** *The probability over random choice of x that there exist a π, π' such that $\mathsf{Verify}(\mathsf{PP}, x, \pi) = 0$ and $\mathsf{Verify}(\mathsf{PP}, x, \pi') = 1$ is at most $2\tilde{\delta} + (20P + 10)/(\sqrt{\tilde{\delta}} \cdot \sqrt{t}) \leq \delta$.*

Proof. The probability is upperbounded by the probability that either x is unopenable or the probability that x is openable and there exists π such that $\mathsf{Verify}(\mathsf{PP}, x, \pi) = 1 - \mathsf{DecRand}(\mathsf{PP}, x)$. Since the probability x is unopenable is at most $2\tilde{\delta}$, it suffices to bound the probability that x is openable and there exists π such that $\mathsf{Verify}(\mathsf{PP}, x, \pi) = 1 - \mathsf{DecRand}(\mathsf{PP}, x)$. by $(20P + 10)/(\sqrt{\tilde{\delta}} \cdot \sqrt{t})$.

Let $\hat{S}$ of size $\hat{t}$ be the set of indeces i for which $n'_i = 1$. Let $[b_i]_{i \in \hat{S}}$ be the corresponding hardcore bits. Let S of size t be the set of indeces i for which $s'_i = 1$. Let $[\tilde{b}_i]_{i \in S'}$ be the corresponding hardcore bits. We want to show that with high probability $\mathsf{Maj}([b_i]_{i \in \hat{S}}) = \mathsf{Maj}([\tilde{b}_i]_{i \in S'})$.

First, assume that $\mathsf{Maj}([b_i]_{i \in \hat{S}}) = 0$ and that $\mathsf{Maj}([\tilde{b}_i]_{i \in S'}) = 1$. This implies that $\sum_{i \in \hat{S}} b_i \leq \hat{t}/2$ and that $\sum_{i \in S'} \tilde{b}_i \geq t/2$.

Further, when x is openable, the following holds:

$$\begin{aligned}
\hat{t}/2 &\leq t/2 \\
&\leq \sum_{i \in S'} \tilde{b}_i \\
&= \sum_{i \in \hat{S}} \tilde{b}_i + \sum_{i \in S' \setminus \hat{S}} \tilde{b}_i \\
&\leq \sum_{i \in \hat{S}} b_i + \frac{8P}{\sqrt{\tilde{\delta}}} + 2P + 2 + \frac{4}{\sqrt{\tilde{\delta}}} + 2 \\
&= \sum_{i \in \hat{S}} b_i + (8P + 4)/\sqrt{\tilde{\delta}} + 2P + 4 \\
&\leq \sum_{i \in \hat{S}} b_i + (10P + 5)/\sqrt{\tilde{\delta}},
\end{aligned}$$

where the third inequality holds due to Claim 4.4.

This implies that $\sum_{i \in \hat{S}} b_i \geq \hat{t}/2 - (10P + 5)/\sqrt{\tilde{\delta}}$.

Symmetrically, to flip from a 1 to a 0, we require that $\sum_{i \in \hat{S}} b_i \leq \hat{t}/2 + (10P + 5)/\sqrt{\tilde{\delta}}$.

Thus, a block can only flip from majority 0 to majority 1 (or vice-versa) if the number of hardcore bits in the set S of size $\hat{t}$ originally had between $\hat{t}/2 - (10P + 5)/\sqrt{\tilde{\delta}}$ and $\hat{t}/2 + (10P + 5)/\sqrt{\tilde{\delta}}$ number of 1's. Using the fact that $\binom{\hat{t}}{\hat{t}/2} \leq \frac{2^{\hat{t}}}{\sqrt{\pi \cdot \hat{t}/2}} \leq \frac{2^{\hat{t}}}{\sqrt{\hat{t}}}$, this occurs with at most $(20P + 10)/(\sqrt{\tilde{\delta}} \cdot \sqrt{\hat{t}})$ probability. Further, by Corollary 3.3 item (2) and Chernoff bounds, we have that with all but negligible probability $\hat{t} \geq (1 - 6\kappa/p)t'$. We therefore set parameters $\tilde{\delta}, t'$ such that $2\tilde{\delta} + (20P + 10)/(\sqrt{\tilde{\delta}} \cdot \sqrt{(1 - 6\kappa/p)t'}) \leq \delta$. □

4.3 Unbiased Property

Recall that for openable x, the encoded bit is the majority of b_j. Thus to show the construction is unbiased it suffices to show each b_j is unbiased. Recall that each b_j is computed from the inner product of uniformly random u_j and the ordered encoding of the set $S_{(y'_i,H_i,h_i)}$ (mod 2). This is unbiased so long as the ordered encoding $S_{(y'_i,H_i,h_i)}$ is not all 0. To achieve this we simply assume encoding always appends a 1.

4.4 Indistinguishability Property

We next state the indistinguishability property of the decoding.

Lemma 4.6. *Let* $(\mathsf{Gen}, \mathsf{DecRand}, \mathsf{Open}, \mathsf{Verify})$ *be as defined in Figure 4.1. Consider the following distributions*

$$\mathcal{D}_0 := (\mathsf{PP} = \mathsf{Gen}(1^\kappa), x \sim U_{t'(m_1+2\tilde{\kappa}+P\cdot\kappa)} \mid \mathsf{DecRand}(\mathsf{PP}, x) = 0),$$
$$\mathcal{D}_0 := (\mathsf{PP} = \mathsf{Gen}(1^\kappa), x \sim U_{t'(m_1+2\tilde{\kappa}+P\cdot\kappa)} \mid \mathsf{DecRand}(\mathsf{PP}, x) = 1),$$

Then $\mathcal{D}_0$ *and* $\mathcal{D}_1$ *are computationally indistinguishable.*

Proof. We show that an adversary contradicting the computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ can be used to construct an adversary contradicting Corollary 3.8.

Note that each of $\mathcal{D}_0$ and $\mathcal{D}_1$ is a distribution over strings x, conditioned on x being openable. Each distribution can be viewed as a convex combination over distributions $\mathcal{D}_{0,\hat{t}}$, $\mathcal{D}_{1,\hat{t}}$ where each distribution corresponds to a choice of $\hat{t}$ (the number of positions in $[t']$ that have pre-image size at most P) that is consistent with x being openable, and then sampling the $\hat{t}$ blocks consistently with x being openable to either 0 or 1.

Note that there must exist some particular $\hat{t}$ such that the distributions $\mathcal{D}_{0,\hat{t}}$ and $\mathcal{D}_{1,\hat{t}}$ are distinguishable.

Further, for $m \in [\lfloor \hat{t}/2 \rfloor]$, let $D(\mathcal{D}_{\hat{t},m})$ be the distribution in which m of the $\hat{t}$ blocks whose pre-image size is at most P have hardcore bit of 1. For $m' \in \{\lfloor \hat{t}/2 \rfloor + 1, \ldots, \hat{t}\}$, let $D(\mathcal{D}_{\hat{t},m'})$ be the distribution in which m' of the $\hat{t}$ blocks whose pre-image size is at most P have hardcore bit of 1. Then there must exist two values $m \in [\lfloor \hat{t}/2 \rfloor], m' \in \{\lfloor \hat{t}/2 \rfloor + 1, \ldots, \hat{t}\}$ such that $|\Pr[D(\mathcal{D}_{\hat{t},m}) = 1] - \Pr[D(\mathcal{D}_{\hat{t},m'}) = 1]| \geq \chi(\kappa)$, where $\chi(\cdot)$ is non-negligible.

By a standard hybrid argument, there must exist some $i \in \{m, \ldots, m'-1\}$ such that $|\Pr[D(\mathcal{D}_{\hat{t},i}) = 1] - \Pr[D(\mathcal{D}_{\hat{t},i+1}) = 1]| \geq \chi'(\kappa)$, where $\chi'(\cdot)$ is non-negligible. Note that the distribution over *sizes* of pre-images $n_1, \ldots, n_{\hat{t}}$ for x drawn from $D(\mathcal{D}_{\hat{t},i})$ and x drawn from $D(\mathcal{D}_{\hat{t},i+1})$ are identical.

Thus, there must exist a particular vector of pre-image sizes $[n_j]_{j\in[\hat{t}]}$, where each $n_j \in S_{\chi''(\kappa)}$ (where $\chi''(\cdot) = \frac{\chi'(\kappa)}{2P\hat{t}}$ is non-negligible), and a position $j^* \in [\hat{t}]$ for which $|\Pr[D(\mathcal{D}_{\hat{t},i,j^*,[n_j]_{j\in\hat{t}}}) = 1] - \Pr[D(\mathcal{D}_{\hat{t},i+1,j^*,[n_j]_{j\in[\hat{t}]}}) = 1]| \geq \chi'(\kappa)/2$. Here $D(\mathcal{D}_{\hat{t},i,j^*,[n_j]_{j\in\hat{t}}})$ is the same as the distribution $D(\mathcal{D}_{\hat{t},i})$, except the pre-image sizes are fixed to $[n_j]_{j\in\hat{t}}$ and the hardcore of the j^*-th block is fixed to 0; $D(\mathcal{D}_{\hat{t},i+1,j^*,[n_j]_{j\in\hat{t}}})$ is the same as the distribution $D(\mathcal{D}_{\hat{t},i+1})$, except the pre-image sizes are fixed to $[n_j]_{j\in\hat{t}}$ and the hardcore of the j^*-th block is fixed to 1.

We now build our reduction by considering a distribution over circuits which hardwire PP, $t-\hat{t}$ samples that have pre-image size greater than P, $\hat{t}-1$ samples with preimage sizes $[n_j]_{j\in[\hat{t}]\setminus j^*}$, where i samples have hardcore bit 0 and $\hat{t}-i-1$ samples have hardcore bit 1, and the values of j^*, n_{j^*}. The reduction will receive as input a block with n_{j^*} number of pre-images and hardcore bit of either 0 or 1 with probability 1/2. The reduction will place the input block in position j^* and to obtain x and forwards PP, x to the adversary. The reduction outputs whatever the adversary does. Note that this distribution over circuits succeeds in guessing the hardcore bit with probability at least $1/2 + \chi'(\kappa)$. Since $\chi'(\cdot)$ is non-negligible, so is $\chi' \cdot \chi''(\cdot) = \frac{(\chi'(\kappa))^2}{2P\hat{t}}$. This implies that there exists a particular circuit in the support of this distribution which breaks the hardcore bit guarantee given in Corollary 3.8, thus leading to contradiction. □

5 Our oNIZK construction with non-uniform Verifier

In this section, we prove the following theorem:

Theorem 5.1. *Assume hard-to-invert exists. For sufficiently large κ, and for all polynomials $\psi(\cdot)$, there exists an oNIZK proof system in the URS model with* non-uniform *verifiers and prover complexity* $\mathsf{poly}(\kappa)\cdot 2^{\kappa}$ *for the language* $\mathsf{HamPath}_{\psi(\kappa)}$*, which is Hamiltonian path for input instances of length $\psi(\kappa)$.*

5.1 Description of Protocol

Our protocol makes use of an underlying hidden bits proof system $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$ for the NP-complete language L of Hamiltonian cycle (See Definition 2.6 and Theorem 2.7). We assume the $(P_{\mathsf{hb}}, V_{\mathsf{hb}})$ proof system requires $n(\kappa)$ hidden bits (i.e. the length of random variable R in Definition2.6 is $n = n(\kappa)$) to prove statements st of length $\ell(\kappa)$ are contained in the language L.

Using Theorem 4.2, we have that for $\delta(\cdot) = \frac{n(\cdot)}{8}$ there exists a partial predicate $\mathsf{DecRand}$ and a tuple of algorithms $(\mathsf{Gen}, \mathsf{Open}, \mathsf{Verify})$, that allow $\delta(\kappa) = \frac{1}{8n(\kappa)}$-confidence verification of $\mathsf{DecRand}$ with efficiency $\mathsf{poly}(1/\delta(\kappa)) \cdot 2^{\kappa}$. We present our protocol in Figure 5.1 and then argue its completeness, soundness, and offline zero knowledge properties.

Figure 5.1: oNIZK Protocol

Setup: the uniform random string $\rho = \rho_1, \ldots, \rho_{n'}$ is divided into n' sections, where each section has length n blocks. Thus $\rho_{i,j}$ refers to the j-th block of the i-th section. $\mathsf{PP} \leftarrow \mathsf{Gen}(1^{\kappa})$ is run and the Verifier receives PP as non-uniform advice.

Prover: The prover receives URS ρ, statement $\mathsf{st} \in L$, and witness w as input. For $i \in [n']$, the prover does the following:

- Parse ρ_i as n blocks $\rho_{i,1}, \ldots, \rho_{i,n}$. $\rho_{i,j}$. Set $b^i_j := \mathsf{Verify}(\mathsf{PP}, \rho_{i,j}, \mathsf{Open}(\mathsf{PP}, \rho_{i,j}))$.
- If $\forall j \in [n], b^i_j \neq \perp$, run the hidden bits prover P_{hb} on the sequence $R := b^i_1, \ldots, b^i_n$, input statement st and witness w. P_{hb} outputs proof π_i and set I_i of opened bits.
- If $\forall j \in [n], b^i_j \neq \perp$, add i to the list L_{good} and output the following for the i-th subproof:
  - The set I_i,
  - For $j \in I_i$, the pair $(b^i_j, \pi^{\mathsf{op}}_{i,j} = \mathsf{Open}(\mathsf{PP}, \rho_{i,j}))$
  - The proof π_i.
- Output the list L_{good}

Verifier: The verifier receives URS ρ, statement st, and proof π as input, as well as the non-uniform advice PP. For each $i \in L_{\mathsf{good}}$ the Verifier does the following:

- Check that for all $j \in I_i$, $\mathsf{Verify}(\mathsf{PP}, \rho_{i,j}, \pi^{\mathsf{op}}_{i,j}) = b^i_j$. We assume that these are provided to the verifier as non-uniform advice.
- Run the hidden bits verifier V_{hb} to verify the proof π_i relative to statement st, the set I_i and hidden bits $R_{I_i} := [b^i_j]_{j \in I_i}$.
- If all checks pass output 1. Otherwise output 0.

If at least $n'/2$ number of 1's were outputted in the previous stage then ACCEPT. Otherwise REJECT.

Completeness. Fix a statement $\mathsf{st} \in L$. The probability over choice of random string ρ that the proof is accepting is the probability that at least half of the substrings $\rho_i, i \in [n']$ have accepting proofs. Since the underlying hidden bits proof has perfect completeness, this is the same as the probability that for at least half of $i \in [n']$ all of $\rho_{i,1}, \ldots, \rho_{i,n}$ satisfy completeness. For every $i \in [n']$, we have by Lemma 4.3, that the probability that all of $\rho_{i,1}, \ldots, \rho_{i,n}$ satisfy completeness is at least $1 - n \cdot \frac{1}{4n} \geq 3/4$. Since the sections of ρ are independent, we can use Hoeffding bounds to lower bound the probability of half the proofs accepting by: $1 - e^{-\frac{n'}{16}}$.

Soundness. Fix a statement $x \notin L$. The probability over choice of random string ρ that there exists an accepting proof is the probability that at least half of the substrings $\rho_i, i \in [n']$ have accepting proofs. For

each $i \in [n']$, the probability that ρ_i has an accepting proof is upperbounded by the probability that one of the following events occurs:

- There exists a valid proof of $x \in L$ in the hidden bits model with respect to bits $R = b_1^i, \ldots, b_n^i$.
- For some $j \in [n]$, $\rho_{i,j}$ is unopenable.
- For some $j \in [n]$, there exist $\pi_{i,j}^{\mathsf{op}}, \pi'^{\mathsf{op}}_{i,j}$ such that $\mathsf{Verify}(\mathsf{PP}, \rho_{i,j}, \pi_{i,j}^{\mathsf{op}}) = 0$ and $\mathsf{Verify}(\mathsf{PP}, \rho_{i,j}, \pi'^{\mathsf{op}}_{i,j}) = 1$.

The first event occurs with at most negligible probability by the soundness of the underlying hidden bits proof. The second event occurs with probability at most $n \cdot \frac{1}{8n} \leq 1/8$ by Lemma 4.3 and the third event occurs with probability at most $n \cdot \frac{1}{8n} \leq 1/8$ by Lemma 4.5. Since the sections of ρ are independent, we can use Hoeffding bounds to upper bound the probability of half the proofs accepting by: $e^{-\frac{n'}{16}}$.

Offline Zero Knowledge (oZK) The simulator Sim is a poly-time algorithm with non-uniform advice which is sampled from the following distribution: Let $\alpha \leq \delta(\kappa)$ be the probability that $x \sim U_{\kappa'+2\tilde{\kappa}+P\cdot\kappa}$ is unopenable. For $i \in [n']$, $j \in [n]$, the non-uniform advice consists of one of the following:

- With probability α, it consists of a single block $\rho_{i,j}$ that is unopenable (i.e. $\rho_{i,j}^0 \sim U_{m_1+2\tilde{\kappa}+P\cdot\kappa} \mid \mathsf{DecRand}(\mathsf{PP}, U_{m_1+2\tilde{\kappa}+P\cdot\kappa}) = ?$)
- With probability $1-\alpha$ it consists of two blocks $\rho_{i,j}^0, \rho_{i,j}^1$ sampled as follows: $\rho_{i,j}^0 \sim U_{m_1+2\tilde{\kappa}+P\cdot\kappa} \mid \mathsf{DecRand}(\mathsf{PP}, U_{m_1+2\tilde{\kappa}+P\cdot\kappa}) = 0$, and $\rho_{i,j}^1 \sim U_{m_1+2\tilde{\kappa}+P\cdot\kappa} \mid \mathsf{DecRand}(\mathsf{PP}, U_{m_1+2\tilde{\kappa}+P\cdot\kappa}) = 1$, along with their corresponding honest openings $\mathsf{Open}(\mathsf{PP}, \rho_{i,j}^0)$, $\mathsf{Open}(\mathsf{PP}, \rho_{i,j}^1)$.

Let $L_{\mathsf{unopenable}}$ be the set of pairs (i,j) such that $\rho_{i,j}$ is unopenable. Let L_{good} be the set of $i \in [n']$ such that $\forall j \in [n]$, $(i,j) \notin L_{\mathsf{unopenable}}$.

In the online phase, the simulator is given the statement st and runs the hidden bits simulator n' times on input st to obtain proofs $\pi_1, \ldots, \pi_{n'}$, sets of opened bits $I_1, \ldots, I_{n'}$, and openings $[b_j^i]_{j\in[I_i], i\in[n']}$. The simulator outputs:

- A URS whose (i,j)-th block is equal to one of the following:
  - $\rho_{i,j}$ if $(i,j) \in L_{\mathsf{unopenable}}$.
  - $\rho_{i,j}^{b_j^i}$ if $i \in L_{\mathsf{good}}$ and $j \in I_i$.
  - $\rho_{i,j}^0$ otherwise.
- A proof that consists of the following:
  - π_i for $i \in L_{\mathsf{good}}$
  - $\pi_{i,j}^{\mathsf{op}} = \mathsf{Open}(\mathsf{PP}, \rho_{i,j}^{b_j^i})$ for $i \in L_{\mathsf{good}}$, $j \in I_i$ (note $\mathsf{Open}(\mathsf{PP}, \rho_{i,j}^{b_j^i})$ is provided as non-uniform advice to the simulator)
  - The list L_{good}.

Indistinguishability of real and simulated views follows immediately from a standard hybrid argument and Lemma 4.6.

6 Getting rid of non-uniform advice

In this section, we prove the following theorem:

Theorem 6.1. *Assume hard-to-invert exists. For sufficiently large κ, and for all polynomials $\psi(\cdot)$, there exists an oNIZK proof system in the URS model with* uniform *verifiers and prover complexity 2^κ for the language $\mathsf{HamPath}_{\psi(\kappa)}$, which is Hamiltonian path for input instances of length $\psi(\kappa)$.*

6.1 AM protocol for Smooth Min-Entropy

We first present an AM protocol for proving that the $1/\tilde{p}$-smooth min-entropy of a distribution $f(U_\kappa)$, where f is efficiently evaluatable, is at least k.

- The Prover provides a $(\tilde{\epsilon}, \tilde{t})$ histogram h for $f(U_\kappa)$, as well as statistics $1/\tilde{p} \leq 3\kappa/p, k$. The parties run the public coin, 2-message AM protocol $\mathsf{VerifyHist}$ to verify that the provided histogram has Wasserstein distance at most $\frac{20}{\tilde{t}}$ from the true histogram h^*. If $\mathsf{VerifyHist}$ rejects, the Verifier rejects.

– The Verifier checks that $\sum_{i \le k/\tilde{\epsilon}} h[i] \le 1/\tilde{p}$ and that $\sum_{k/\tilde{\epsilon} \le i \le (k+2)/\tilde{\epsilon}} h[i] \ge 1/p$. If both checks pass, the verifier accepts. Otherwise it rejects.

The following can be verified by inspection of the VerifyHist protocol.

Claim 6.2. The prover complexity of the above protocol is at most $\mathsf{poly}(\tilde{t}, \kappa) \cdot T_{p,1}$, where $T_{p,1}$ is the maximum time (over all $y \in 2^{\kappa'}$) it takes to output the set $f^{-1}(y)$.

Claim 6.3. We have the following completeness and soundness for the above protocol:

– **Completeness:** If the Prover is honest and $f(U_\kappa)$ is $3\kappa/p$-far from a distribution with min-entropy k and has at least $1/p$ weight on elements with pre-image sizes between $\frac{2^\kappa}{2^{k+2}}, \frac{2^\kappa}{2^k}$, then the Verifier accepts with probability $1 - 2^{-\kappa}$.
– **Soundness:** For any cheating prover, if the Verifier does not reject, then with probability $1 - 2^{-\kappa}$, $f(U_\kappa)$ is at most $1/p + 1/\Delta$-far from a distribution with min-entropy $k - \tilde{\epsilon} \cdot \Delta$ and has at least $1/p - 1/\Delta$ weight on elements with pre-image sizes between $\frac{2^\kappa}{2^{k+2+\tilde{\epsilon}\cdot\Delta}}, \frac{2^\kappa}{2^{k-\tilde{\epsilon}\cdot\Delta}}$. Setting $\Delta > 2p$ and $\tilde{\epsilon}$ sufficiently large (but still 1/poly), we obtain that with all but negligible probability $f(U_\kappa)$ is at most $4\kappa/p$-far from a distribution with min-entropy $k - 1$ and has at least $1/(2p)$ weight on elements with pre-image sizes between $\frac{2^\kappa}{2^{k+3}}, \frac{2^\kappa}{2^{k-1}}$.

The above claim follows immediately from Claim 2.16.

6.2 AM protocol for learning μ, μ'

Now that the values of m_1, m_2 have been determined, we will have a preamble phase to estimate μ, μ' which proceeds as follows: Let $\tilde{\kappa}$ be the number of random bits needed to specify hashes H, h.

– Let $\mathcal{R}$ be the set of $(x, H, h) \in \{0,1\}^{\kappa + 2\tilde{\kappa}}$ such that $h(x) = 0^{m_2}$. Note that this set is efficiently samplable.
– The Prover provides a $(\tilde{\epsilon}, \tilde{t})$ histogram h for the distribution $\tilde{f}(U_{\mathcal{R}})$, where $\tilde{f}(x, H, h) = (H \circ f(x), H, h)$, and where $\tilde{t} = \frac{\kappa + 2\tilde{\kappa}}{\tilde{\epsilon}}$. The parties run the public coin, 2-message AM protocol VerifyHist to verify that the provided histogram has Wasserstein distance at most $\frac{20}{\tilde{t}}$ from the true histogram h^*. If VerifyHist rejects, the Verifier rejects.
– The Verifier does the following:
  • For $j \in (P+1)$, let $k(j) = \lfloor \frac{-\log(j) + \kappa + 2\tilde{\kappa} - m_2}{\tilde{\epsilon}} \rfloor$. The Verifier checks that $h[i] = 0$ for all $i \in [k(P+1)] \setminus \{k(j)\}_{j \in (P+1)}$. If not, the Verifier rejects.
  • Computes $\overline{\mu'} = \sum_{i \ge (-\log(P) + \kappa + 2\tilde{\kappa} - m_2)/\tilde{\epsilon}} h[i] \cdot \frac{2^{i \cdot \tilde{\epsilon}}}{2^{m_1 + 2\tilde{\kappa}}}$ and $\mu = \sum_{i > (-\log(P) + \kappa + 2\tilde{\kappa} - m_2)/\tilde{\epsilon}} h[i] \cdot \frac{2^\kappa}{2^{m_1 + m_2}}$. The Verifier outputs $\mu' = (1 - \overline{\mu'}), \mu$.

The following can be verified by inspection of the VerifyHist protocol.

Claim 6.4. The prover complexity of the above protocol is at most $\mathsf{poly}(\tilde{t}, \kappa) \cdot T_{p,2}$, where $T_{p,2}$ is the maximum time (over all $(y', H, h) \in 2^{m_1 + 2\tilde{\kappa}}$) it takes to output the set $\tilde{f}^{-1}((y', H, h)) \cap S$.

Claim 6.5. Let $(mu')^* := \Pr_{(y', H, h) \sim U_{m_1 + 2\tilde{\kappa}}}[|\tilde{f}^{-1}(y', H, h) \cap S| > P]$ and let $\mu^* := \mathbf{E}_{(y', H, h) \sim U_{m_1 + 2\tilde{\kappa}} || \tilde{f}^{-1}(y', H, h) \cap S| \le P}[|\tilde{f}^{-1}(y', H, h) \cap S|]$. We have the following completeness and soundness for the above protocol:

– **Completeness:** If the Prover is honest, then the Verifier outputs reject with probability at most $2^{-\kappa - 2\tilde{\kappa}}$. If the Verifier does not reject then:
  • μ' outputted by the Verifier is within $2^{\pm\tilde{\epsilon}} \cdot (\mu')^* \subseteq [(\mu')^* - \frac{1}{t'}, (\mu')^* + \frac{1}{t'}]$, by our choice of sufficiently small $\tilde{\epsilon}$.
  • μ outputted by the Verifier is within $2^{\pm 2\tilde{\epsilon}} \cdot \mu^* \subseteq [\mu^* - \frac{1}{t}, \mu^* + \frac{1}{t}]$, by our choice of sufficiently small $\tilde{\epsilon}$.
– **Soundness:** For any cheating prover, if the Verifier does not reject then with probability $1 - 2^{-\kappa - 2\tilde{\kappa}}$:
  • μ' outputted by the Verifier is within $(\mu')^* \pm \frac{20 \cdot P \cdot 2^{\kappa + 2\tilde{\kappa} - m_2}}{\Delta \cdot 2^{m_1 + 2\tilde{\kappa}}} + \frac{1}{2t'} \subseteq [(\mu')^* - \frac{1}{t'}, \hat{\mu}' + (\mu')^*]$, by our choice of sufficiently small (1/poly) $\tilde{\epsilon}$ and sufficiently large Δ.

- μ outputted by the Verifier is within $\mu^* \pm \frac{20}{\Delta} + \frac{1}{2t} \subseteq [\hat{\mu} - \frac{1}{t}, \hat{\mu} + \frac{1}{t}]$, by our choice of sufficiently small (1/poly) $\tilde{\epsilon}$ and sufficiently large Δ.

Proof of Claim 6.5. We first bound $|(\mu')^* - \mu'|$. Assume h, h^* have Wasserstein distance at most $\frac{20}{t}$. Recall that for $j \in (P+1)$, $k(j) = \lfloor \frac{-\log(j) + \kappa + 2\tilde{\kappa} - m_2}{\tilde{\epsilon}} \rfloor$. Recall further that if the Verifier does not reject then $h[i] = 0$ for all $i \in [k(P+1)] \setminus \{k(j)\}_{j \in (P+1)}$. Thus, $\overline{\mu}' = \sum_{j \in (P)} h[k(j)] \cdot \frac{2^{k(j) \cdot \tilde{\epsilon}}}{2^{m_1 + 2\tilde{\kappa}}}$ Additionally, $h^*[i] = 0$ for all $i \in [k(P+1)] \setminus \{k(j)\}_{j \in (P+1)}$. Let $\tilde{\mu}' = \sum_{j \in (P)} h^*[k(j)] \cdot \frac{2^{k(j) \cdot \tilde{\epsilon}}}{2^{m_1 + 2\tilde{\kappa}}}$.

We claim that for $j \in (P)$ $|h^*[k(j)] - h[k(j)]| \leq \frac{20}{\Delta}$. This follows from the fact that for $j \in (P)$ $h^*[k(j)] = \sum_{i \in [(k(j+1)+k(j))/2, (k(j)+k(j-1))/2]} h^*[i]$, $h[k(j)] = \sum_{i \in [(k(j+1)+k(j))/2, (k(j)+k(j-1))/2]} h[i]$ and from Claim 2.16, with $\tilde{\epsilon}$ set sufficiently small so that $\Delta \leq \min_j (k(j) - k(j+1))/2 \geq \frac{1}{2P(\ln(2))\tilde{\epsilon}}$.

We have via Cauchy-Schwartz that

$$\begin{aligned}
|\tilde{\mu}' - \overline{\mu}'| &= \left| \sum_{j=0}^{P} (h^*[k(j)] - h[k(j)]) \cdot \frac{2^{i(j) \cdot \tilde{\epsilon}}}{2^{m_1 + 2\tilde{\kappa}}} \right| \\
&\leq \sqrt{\sum_{j=0}^{P} (h^*[k(j)] - h[k(j)])^2} \cdot \sqrt{\sum_{j=0}^{P} \left(\frac{2^{k(j) \cdot \tilde{\epsilon}}}{2^{m_1 + 2\tilde{\kappa}}} \right)^2} \\
&\leq \frac{20 \cdot P \cdot 2^{\kappa + 2\tilde{\kappa} - m_2}}{\Delta \cdot 2^{m_1 + 2\tilde{\kappa}}}.
\end{aligned}$$

Finally, $\tilde{\mu}'$ deviates from the correct expectation $(\overline{\mu}')^*$ by at most a multiplicative factor of $2^{\pm \tilde{\epsilon}}$. We set $\tilde{\epsilon}$ sufficiently small such that $\tilde{\mu}' \cdot 2^{\pm \tilde{\epsilon}} \subseteq \tilde{\mu}' \pm \frac{1}{2t'}$.

Thus, $|(\mu')^* - \mu'| = |(\overline{\mu}')^* - \overline{\mu}'| \leq \frac{20 \cdot P \cdot 2^{\kappa + 2\tilde{\kappa} - m_2}}{\Delta \cdot 2^{m_1 + 2\tilde{\kappa}}} + \frac{1}{2t'}$.

We next bound $|\mu^* - \mu|$. Similarly to above, we have that $\mu = \sum_{i \geq k(P)} h[i]$. Let $\tilde{\mu} = \sum_{i \geq k(P)} h^*[k(j)]$.

We claim that for $j \in (P)$ $|\tilde{\mu} - \mu]| \leq \frac{20}{\Delta}$. This follows from the fact that $\tilde{\mu} = \sum_{i \geq k(P) - (k(P) - k(P+1))/2]} h^*[i]$, $\mu = \sum_{i \geq k(P) - (k(P) - k(P+1))/2]} h[i]$ and from Claim 2.16, with $\tilde{\epsilon}$ set sufficiently small so that $\Delta \leq (k(P) - k(P+1))/2 \geq \frac{1}{2P(\ln(2))\tilde{\epsilon}}$.

Finally, $\tilde{\mu}$ deviates from the correct expectation μ^* by at most a multiplicative factor of $2^{\pm \tilde{\epsilon}}$. We set $\tilde{\epsilon}$ sufficiently small such that $\tilde{\mu} \cdot 2^{\pm \tilde{\epsilon}} \subseteq \tilde{\mu} \pm \frac{1}{2t}$.

Thus, $|\mu^* - \mu| \leq \frac{20}{\Delta} + \frac{1}{2t}$. □

6.3 Using the AM protocols to obtain a uniform verifier oNIZK in the URS model

We apply Theorem 2.13 to the protocols given in Sections 6.1 and 6.2 to reduce the constant round, public-coin protocols down to 2 message, public-coin protocols. Note that this also accounts for the fact that the Prover sees the Verifier's first message before committing to a histogram h. This transformation increases the Prover's complexity by at most a $\mathsf{poly}(\kappa)$ factor. Thus, we obtain 2-message AM protocols for obtaining the statistics k, μ', μ with a polynomial-time Verifier and $\mathsf{poly}(\kappa) \cdot 2^{\kappa}$-time Prover.

Given these protocols, we now use the same construction as the one presented in Section 5, except we add a preamble phase in which the two AM protocols from above are executed. Specifically, we pre-pend two uniformly random strings ρ_1', ρ_2' to the URS and these two strings will be used as the verifier's first message in each of the two AM protocols, respectively. The Prover will use the remainder of the uniform random string (URS) to complete the proof as before.

References

1. Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 701–710, Seattle, WA, USA, May 21–23, 2006. ACM Press.
2. Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Comput. Complex.*, 25(2):349–418, 2016.

3. László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.
4. Marshall Ball, Dana Dachman-Soled, Siyao Guo, Tal Malkin, and Li-Yang Tan. Non-malleable codes for small-depth circuits. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 826–837, Paris, France, October 7–9, 2018. IEEE Computer Society Press.
5. Marshall Ball, Dana Dachman-Soled, and Mukul Kulkarni. New techniques for zero-knowledge: Leveraging inefficient provers to reduce assumptions, interaction, and trust. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 674–703, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.
6. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.
7. Andrej Bogdanov, Kunal Talwar, and Andrew Wan. Hard instances for satisfiability and quasi-one-way functions. In Andrew Chi-Chih Yao, editor, *ICS 2010: 1st Innovations in Computer Science*, pages 290–300, Tsinghua University, Beijing, China, January 5–7, 2010. Tsinghua University Press.
8. Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 711–720, Seattle, WA, USA, May 21–23, 2006. ACM Press.
9. Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st Annual Symposium on Foundations of Computer Science*, pages 283–293, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.
10. Riddhi Ghosal, Yuval Ishai, Alexis Korb, Eyal Kushilevitz, Paul Lou, and Amit Sahai. Hard languages in NP ∩ conp and NIZK proofs from unstructured hardness. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1243–1256. ACM, 2023.
11. Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001.
12. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.
13. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.
14. Iftach Haitner, Mohammad Mahmoody, and David Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 76–87. IEEE Computer Society, 2010.
15. Thomas Holenstein and Robin Künzler. A new view on worst-case to average-case reductions for NP problems. In Zhipeng Cai, Alex Zelikovsky, and Anu G. Bourgeois, editors, *Computing and Combinatorics - 20th International Conference, COCOON 2014, Atlanta, GA, USA, August 4-6, 2014. Proceedings*, volume 8591 of *Lecture Notes in Computer Science*, pages 203–214. Springer, 2014.
16. Alon Rosen, Gil Segev, and Ido Shahaf. Can PPAD hardness be based on standard cryptographic assumptions? In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 747–776, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.
17. Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il