Electronic Colloquium on Computational Complexity, Report No. 63 (2017)

Exponentially-Hard gap-CSP and local PRG via Local Hardcore Functions

Benny Applebaum*

April 10, 2017

Abstract

The gap-ETH assumption (Dinur 2016; Manurangsi and Raghavendra 2016) asserts that it is exponentially-hard to distinguish between a satisfiable 3-CNF formula and a 3-CNF formula which is at most 0.99-satisfiable. We show that this assumption follows from the exponential hardness of finding a satisfying assignment for *smooth* 3-CNFs. Here smoothness means that the number of satisfying assignments is not much smaller than the number of "almost-satisfying" assignments. We further show that the latter ("smooth-ETH") assumption follows from the exponential hardness of solving constraint satisfaction problems over well-studied distributions, and, more generally, from the existence of any exponentially-hard locally-computable one-way function. This confirms a conjecture of Dinur (ECCC 2016).

We also prove an analogous result in the cryptographic setting. Namely, we show that the existence of exponentially-hard locally-computable pseudorandom generator with linear stretch (el-PRG) follows from the existence of an exponentially-hard locally-computable "almost regular" one-way functions.

None of the above assumptions (gap-ETH and el-PRG) was previously known to follow from the hardness of a search problem. Our results are based on a new construction of general (GL-type) hardcore functions that, for any exponentially-hard one-way function, output linearly many hardcore bits, can be locally computed, and consume only a linear amount of random bits. We also show that such hardcore functions have several other useful applications in cryptography and complexity theory.

*School of Electrical Engineering, Tel-Aviv University, benny.applebaum@gmail.com. Supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, by an ICRC grant and by the Check Point Institute for Information Security.

ISSN 1433-8092

1 Introduction

A constraint satisfaction problem (CSP) consists of a list of m constraints over n formal variables $x = (x_1, \ldots, x_n)$ where each constraint depends on a constant k number of variables. The computational intractability of CSPs is a basic, well studied phenomena in computer science. The famous Cook-Levin theorem [Coo71, Lev73] shows that, assuming $\mathbf{P} \neq \mathbf{NP}$, no polynomial-time algorithm can solve the search problem (i.e., find a satisfying assignment) even when each constraint depends on $k = 3$ variables. A seemingly bolder conjecture asserts that the *gap* version of the problem is hard, namely, that one cannot distinguish between k-CSPs which are satisfiable to k-CSPs for which every assignment fail to satisfy at least γ-fraction of the constraints for some constant $\gamma > 0$. The celebrated PCP theorem [AS98, ALM+98] shows that the search variant reduces in polynomial-time to the gap variant, and therefore the two assumptions are actually equivalent. While we have a relatively clear understanding of polynomial hardness, the picture is less clear when it comes to exponential-time hardness.

It is widely believed that k-CSP (or specifically 3-SAT) over n variables cannot be solved in less than exponential time in n (i.e., $2^{\beta n}$ for some constant $\beta > 0$). This *Exponential-Time Hypothesis* (**ETH**) was introduced almost two decades ago by Impagliazzo and Paturi [IP99], and has gained a lot of attention lately due to its implications to the exact complexity of problems inside $\mathbf{P}$ ("fine-grained complexity"). Very recently, Dinur [Din16] and Manurangsi and Raghavendra [MR16] independently made a similar exponential-time conjecture regarding the hardness of Gap-CSPs.

Assumption 1.1 (gapETH). *For some constants $\beta, \gamma > 0$ and an integer k there is no $2^{\beta n}$-time probabilistic algorithm that, given a k-CSP φ over n variables, distinguishes, with probability better than $2/3$, between the case in which the CSP is satisfiable from the case in which every assignment violates a γ-fraction of constraints.*[1]

This new assumption have already found several exciting consequences including a weak form of the so called *sliding scale conjecture* of [BGLR94], tight results on the hardness of dense CSPs, strong inapproximability results for the Densest Subgraph problem, and parameterized inapproximability of some fundamental combinatorial optimization problems like Independent Set and Set Cover [Din16, MR16, Man16, CCK+17]. Clearly, **gapETH** implies **ETH**, however, the converse direction is currently unknown to hold. Indeed, known PCP reductions from search-CSPs to Gap-CSPs blow-up the number of variables by a super-constant factor (polylogarithmic at best [BS08, Din07]) and therefore fail to preserve exponential hardness. This raises the following natural question:

Question 1. *Can we base* **gapETH** *on* **ETH***, or at least on the exponential hardness of some search problem?*

It is worth mentioning that **ETH** easily implies that gap problems for which the constraints are *non-local* (e.g., circuit satisfiability) are exponentially hard. It is the combination of constant locality and exponential hardness that makes the reduction challenging. Put differently, the essence of the problem is to move from search problems to gap problems without introducing *too many auxiliary variables and while preserving the locality of the constraints*.

[1] By the PCP theorem, one can focus, without loss of generality, on the special case of 3-CNF problems (at the expense of decreasing the constant γ). Moreover, it is shown in [Din16] that, without loss of generality, the number of constraints can be assumed to be linear in the number of variables.

The cryptographic setting. A similar question also arises in the cryptographic setting. Let us first rephrase the (worst-case) intractability of CSPs in a functional form. Given a CSP φ over n variables and m constraints of arity k, we define a function $f : \{0,1\}^n \to \{0,1\}^m$ that takes an n-bit assignment x as an input, and outputs an m-bit string y whose i-th bit equals to 0 if and only if the i-th constraint is satisfied. The function is *k-local* in the sense that every output depends on at most k-inputs. Using this terminology, the intractability of the search problem asserts that it is hard to find a preimage of the all-zero string y, whereas the intractability of the gap problem says that it is hard to distinguish between the case where $y = 0^m$ is in the image of f to the case where y is γ-far (in relative Hamming distance) from the image.

The cryptographic setting addresses similar inversion/distinguishing problems but requires average-case hardness with respect to a natural distribution over the strings y. Specifically, we say that f is a *one-way function* if no efficient algorithm can invert f on an image $y = f(x)$ of a uniformly chosen input $x \xleftarrow{R} \{0,1\}^n$. We say that f is a *pseudorandom generator* (PRG) if (1) $m > n$ and (2) no efficient algorithm can distinguish between an image $y = f(x)$ of a uniformly chosen $x \xleftarrow{R} \{0,1\}^n$ (a satisfiable instance) to a randomly chosen m-bit string $y \xleftarrow{R} \{0,1\}^m$. When the PRG has a *linear stretch* $m - n = \Omega(n)$, a random $y \xleftarrow{R} \{0,1\}^m$ is likely to be $\Omega(1)$-far from the image of f, and so the resulting CSP instance is likely to be highly-unsatisfiable.

In the polynomial hardness regime, we have a relatively clear picture of locally-computable (aka $\mathbf{NC^0}$) cryptography. Locally-computable OWFs can be based on $\mathbf{NC}^1$ OWFs [AIK06], and there are generic local transformations from OWFs to PRGs with low (sublinear) stretch [AIK06, HRV13]. Local PRGs with linear stretch can be based on a concrete candidate OWF of Goldreich [Gol11] via a transformation of [App13].[2] However, all these reductions have a polynomial blow-up in the input length, and so they fail to preserve exponential hardness. As a result, although Goldreich's function is believed to be exponentially hard to invert [Gol11, CEMT14, BR13, BIO14], it is unknown how to (provably) turn it into an exponentially-strong $\mathbf{NC^0}$ PRG with linear-stretch. On the other hand, given known attacks, we have no reason to believe that such PRGs do not exist, and the literature contains several potential candidates (cf. [MST06, AIK08, ABR16, OW14] and [App16] for a survey.) We therefore ask:

Question 2. *Are there exponentially-strong PRGs with linear stretch in $\mathbf{NC^0}$? If so, can we base them on one-wayness assumptions?*

Exponentially-strong local PRGs provide an asymptotically optimal level of security together with an asymptotically optimal level of efficiency (since each output bit can be computed via a constant number of operations). Moreover, it is shown in [AIK08] that, under these efficiency and security requirements, one cannot hope for more than linear-stretch, i.e., $m - n = O(n)$. The existence of *exponentially-strong locally-computable PRGs with linear stretch* (hereafter referred to as the $\mathbf{elPRG}$ assumption) can be therefore viewed as a fundamental question regarding the best possible tradeoff between efficiency and security for a basic cryptographic object.

We further mention that if locality is not required then exponentially-strong PRGs with arbitrary stretch can be based on any exponentially-hard regular OWF [HILL99, HHR11].[3]

[2] This transformation also yields polynomial-stretch PRGs with either slightly super-constant locality or inverse polynomial distingsuishing advantage. We mention that large stretch is crucial for several important applications of local PRGs including cryptography with constant computational overhead [IKOS08] and general purpose program obfuscation [Lin16a, Lin16b, LV16].

[3] Currently, even in the non-local setting, it is unknown how to get an exponentially-strong PRG based on a gen-

1.1 Our results

We partially answer Questions 1 and 2 by showing that both, **gapETH** and **elPRG**, follow from the exponential hardness of *search* problems that satisfy some "smoothness" or "regularity" condition. We move on to a formal statement of our results starting with the worst-case setting.

1.1.1 Sufficient conditions for gapETH

A CSP φ over n variables is (γ, α)-*smooth* if the number of assignments that satisfy at least $1-\gamma$ of the constraints is at most $2^{\alpha n}$-times larger than the number of satisfying assignments. That is, the relaxation from full satisfiability to "almost satisfiability" does not increase the number of solutions by much. We introduce the *smooth exponential-time hypothesis* (**smETH**) which asserts that for some integer k and constants a, γ, β and $\alpha < \beta/5$, there is no $2^{\beta n}$-time algorithm for solving k-CSPs over n variables and an constraints which are (γ, α)-smooth. (Here and throughout the paper, we say that an algorithm A solves a promise search problem if given a Yes instance A outputs a solution with probability $\frac{1}{2}$.)

Theorem 1.2. **gapETH** *follows from* **smETH**.

We observe that typical candidates for hard (satisfiable) CSPs turn to be smooth. This is true for random k-CNFs whose clause-to-variable density m/n is just below the satisfiability threshold, and for CSPs arising from Goldreich's one-way function. More generally, we show that the existence of a locally-computable exponentially-strong OWF yields a distribution over CSPs which is concentrated over exponentially-hard smooth instances (Theorem 4.9). Combining with Theorem 1.2, we conclude the following theorem.

Theorem 1.3. **gapETH** *follows from the existence of any exponentially-strong locally-computable OWF* $f : \{0,1\}^n \to \{0,1\}^{O(n)}$.

The theorem holds even if the function is only one-way on a tiny (e.g., sub-exponential fraction of the inputs) and even if constant locality holds only after some, possibly non-local, public preprocessing. In addition to Goldreich's original assumption [Gol11], we show that such OWFs follow from the exponential hardness of random CNFs, and from a coding-related intractability assumption of Druk and Ishai [DI14]. We view these results as providing strong evidence towards the validity of **gapETH**. Getting rid of the smoothness condition and basing **gapETH** on **ETH** remains an interesting open problem.

1.1.2 Sufficient conditions for elPRG

A function $f : \{0,1\}^n \to \{0,1\}^m$ is α-*almost regular* if the number of preimages of $y \in \mathrm{Im}(f)$ can vary by a factor of at most $2^{\alpha n}$, i.e., $|f^{-1}(y)| \in [s, s \cdot 2^{\alpha n}]$, for some $s = s(n)$.

Theorem 1.4. *Suppose that there exists an* $\mathbf{NC^0}$ *function* $f : \{0,1\}^n \to \{0,1\}^{an}$ *which is* $2^{\beta n}$*-hard one-way function and is* α*-almost regular for constants* $\alpha < \beta/6$ *and* a*. Then* **elPRG** *holds. In particular, there exists an exponentially-strong linear-stretch PRG with locality 4.*

eral exponentially-hard OWF. The best known construction, due to [VZ12], blows up the number of variables by a polylogarithmic factor.

As special cases, **elPRG** follows from any exponentially-hard local OWF which is either regular, (i.e., $\alpha = 0$) or is "at most $2^{\alpha n}$-to-1" in the sense that no output has more than $2^{\alpha n}$ preimages. (In fact, in the latter case, the relation between α and β can be slightly improved.) Baron et al. [BIO14] presented, under similar conditions, a linear-time computable transformation from OWF to linear-stretch PRG. Our construction (which heavily relies on their result) has the additional advantage of being local.

Since (a variant of) Goldreich's OWF satisfy the almost-regularity condition [BIO14], we can plug it into the theorem and get the first exponentially-strong locally-computable PRG with linear-stretch whose security can be reduced to a one-wayness assumption. The resulting locality (4) is almost optimal since 2-local functions can be inverted in polynomial time [Gol11]. In fact, using the aforementioned coding-based assumption of [DI14], we get an optimal locality of 3. Finally, let us mention that Theorem 1.4 crucially relies on exponential hardness and so it is incomparable to the reductions of [App13] which apply to the polynomial-hardness regime (and are tailored to Goldreich's concrete one-way function). On the other hand, Theorem 1.4 bypasses some of the limitations of [App13]; Specifically, it can be based on a length preserving function $f : \{0,1\}^n \to \{0,1\}^n$ (as opposed to random local functions with *large* output length in [App13]) and it yields a PRG G which can be computed locally with no preprocessing (as opposed to a collection of local PRGs in [App13]).

1.1.3 Other results

Our tools have several other applications both in cryptography and complexity theory. First we derive a new *isolation lemma* that reduces the satisfiability of general k-CSPs to the satisfiability of k-CSPs that are guaranteed to have at most a single satisfying assignment.

Theorem 1.5 (Local Isolation Lemma)**.** *There exists a randomized polynomial-time reduction that takes a k-CSP φ over n variables and m constraints and map it into a new $\max(k,3)$-CSP φ' over $n' = n + O(n)$ variables and $m' = m + O(n)$ constraints such that: (1) If φ is unsatisfiable so is φ'; and (2) If φ is satisfiable then, with probability $\Omega(1/n)$, the CSP φ' is uniquely satisfiable.*

In a classical work Valiant and Vazirani [VV86] presented a polynomial-time isolation lemma which reduces k-SAT to a Unique Circuit-SAT instance φ'. One can further reduce φ' to a k-CSP instance (via the standard transformation), however this introduces a polynomial blow-up in the number of variables. The resulting transformation therefore preserves polynomial hardness but fails to preserve super-polynomial hardness. This problem was observed by Calabro et al. [CIKP08], who described, for every $\varepsilon > 0$, an $\exp(\varepsilon n)$-time isolation lemma that maps k-CSP φ into k_ε-CSP φ' while preserving the number of variables. In contrast, our reduction runs in polynomial-time (in fact, almost linear) but introduces a linear blow-up in the number of variables. As an immediate corollary we conclude that a $T(n)$-time algorithm for Unique-k-CSP over n variables implies a $\mathrm{poly}(n) \cdot T(O(n))$-time algorithm for k-CSP. In particular, for any $f(n) = \omega(\log n)$ if k-CSP cannot be solved by $2^{f(n)}$-time algorithms then Unique-k-CSP cannot be solved in $2^{f(\Omega(n))}$ time.

Moving back to the cryptographic domain, our tools allow us to transform intractable coding-related problems (such as decoding noisy codewords) defined over highly-efficient codes (i.e., computable by a linear-size circuit) into locally-computable cryptographic primitives (like OWFs

and PRGs) while preserving exponential hardness. Examples for such intractable linear-time computable codes were presented by Druk and Ishai [DI14] who proved a "win-win" result: If the decoding problems turn out to be tractable this would lead to interesting progress in coding theory (i.e., linear-time encodable and efficiently decodable codes that meet the Gilbert-Varshamov bound.) Previously, exponentially-hard locally-computable primitives were mainly based on direct local assumptions (such as Goldreich's candidate). Our tools provide a new alternative approach for such constructions.

1.2 Techniques

We sketch the basic ideas underlying the proof of Theorem 1.2 (**smETH** $\Rightarrow$ **gapETH**). As a warm-up, we begin with an exponential-time Turing reduction from search k-CSP problem to gap Circuit-SAT problem where the constraints are *non-local*. Later, we will show that a variant of this reduction yields local CSPs.[4] Our reduction strongly relies on list-decodable codes (or equivalently general hardcore functions).

1.2.1 From search to approximate-decision via list-decoding

Our goal is find a satisfying assignment for a satisfiable k-CSP φ over n variables $x = (x_1, \ldots, x_n)$ and $m = O(n)$ constraints in time $2^{\beta n}$ based on a $2^{\beta' n}$-time algorithm A that distinguishes between satisfiable instances and instances whose value is at most $1 - \gamma$. Let x^* be a satisfying assignment for φ. The basic strategy is to use A in order to get a noisy version of a codeword of x^*, and then use a decoding procedure to recover x^*. For now, our code will be based on the Goldreich-Levin multi-output hardcore function [GL89, Gol01] (which can be viewed as a list-decodable code [STV01]).

Let $\mathsf{GL}_w : \mathbb{F}_2^n \to \mathbb{F}_2^s$ denote the Goldreich-Levin hardcore function that takes a vector $x \in \mathbb{F}_2^n$ and a random $s \times n$ binary matrix w, and outputs the matrix-vector product $w \cdot x$. We think of x as an information word and of $(\mathsf{GL}_w(x))_{w \in \mathbb{F}_2^{s \times n}}$ as a huge codeword of length 2^{sn} over a 2^s-size alphabet. Given a matrix w, we would like to guess $\mathsf{GL}_w(x^*)$ with advantage of at least $2^{-s} + \varepsilon$. For this, we define for each possible $z \in \mathbb{F}_2^s$ a new (non-local) CSP

$$\varphi_{z,w} := \varphi \wedge (\mathsf{GL}_w(x) = z),$$

and reject z if the algorithm A claims that $\varphi_{z,w}$ is "highly un-satisfiable". Our guess for the value $\mathsf{GL}_w(x^*)$ is chosen uniformly among all strings z that pass the test. Assuming that the above procedure B succeeds with probability $2^{-s} + \varepsilon$, we can use it as a sub-routine inside the Goldreich-Levin decoding algorithm and recover x^* by making $\mathrm{poly}(1/\varepsilon)2^s$ calls to A. Since we started with $2^{\beta n}$-time hardness assumption, we can take $\varepsilon = 2^{-\Theta(n)}$ and $s = \Theta(n)$ where the constants in Theta notation are properly chosen.

To analyze the success probability first observe that $z^* = \mathsf{GL}_w(x^*)$ always passes the test since $\varphi_{z^*,w}$ is satisfiable (by the assignments x^*). We should further argue that not too many fake solutions pass the test. Since we added a linear number of constraints, $\varphi_{z^*,w}$ passes the test only if there exists an assignment x' that violates only γ'-fraction of the constraints in φ and γ'-fraction

[4] There are alternative simpler (polynomial-time Karp) reductions from k-CSP to gap Circuit-SAT, e.g., based on error-correcting codes. However, we do not know how to "localize" them.

of the constraints in $\mathsf{GL}_w(x) = z$, for some constant γ'. This means that z is γ'-close to $\mathsf{GL}_w(x')$ for some assignment x' which almost-satisfy φ. It follows that when φ has "few almost-satisfying assignments" (here few stands for $2^{\beta'' n}$ for some sufficiently small constant β''), the number of "fake solutions" is exponentially sparse in $\mathbb{F}_2^s$ and the reduction succeeds (for a properly chosen γ).

We complete the reduction by showing that smooth CSPs can be efficiently reduced to CSPs with few almost-satisfying assignments using standard hashing techniques. Indeed, suppose that there are $T = 2^t$ satisfying assignments and at most $T \cdot 2^{\beta'' n}$ almost-satisfying assignments. Then, we can add t (non-local) constraints of the form $g_v(x) = 0^t$ where g_v is sampled from a family $\{g_v\}$ of pairwise independent hash functions. As a result, the set of almost satisfying assignments and the set of satisfying assignments are likely to decrease by a factor of about T, and so we are likely to get a satisfiable CSP with few almost-satisfying assignments.

1.2.2 A local reduction via randomized encoding

To make the reduction local we need locally-computable list-decoding codes and locally-computable hash functions. Unfortunately, locally-computable functions cannot compute such objects, and we are forced to compromise and use weaker tools. In particular, consider the *randomized code* $\hat{h}_w(x; r)$ whose input consists of a random string $r \in \{0,1\}^\rho$, in addition to w and x. For a random choice of r, the distribution $\hat{h}_w(x; r)$ encodes the value $z = \mathsf{GL}_w(x)$ in the sense that $\hat{h}_w(x; r)$ is distributed uniformly over a set of 2^ρ distinct strings $D_z \subset \{0,1\}^{\hat{s}}$. Each set D_z is associated with a single $z \in \{0,1\}^s$, and together the sets $\{D_z\}_{z \in \{0,1\}^s}$ form a partition of $\{0,1\}^{\hat{s}}$. The function $\hat{h}_w(x; r)$ can be therefore viewed as a *perfect randomized encoding* (RE) [IK00, AIK06] of the function $\mathsf{GL}_w(x)$.

Although the function $\hat{h}_w(x; r)$ is not a list-decodable code it can be used (with some quantitative loss) in the above reduction. Unlike the "index" w, the randomness $r = (r_1, \ldots, r_\rho)$ cannot be fixed and is therefore treated as a sequence of new formal variables. That is, the instance $\varphi_{z,w}$ is defined over the variables x and r. Consequently, the reduction preserves exponential hardness as long as the number of auxiliary variables, ρ, is at most linear in n. Similarly, we can replace the pairwise independent hash functions with their RE at the expense of introducing additional auxiliary variables. Overall the task of "localizing" the reduction boils down to constructing local REs with linear complexity.

1.2.3 Local REs with linear complexity

The literature [IK00, AIK06] contains several constructions of local REs for any $\mathbf{NC}^1$ or even log-space computable function $f : \{0,1\}^n \to \{0,1\}^s$ (cf. [Ish13] for a survey). However, the complexity of all known constructions grows at least linearly with ns, the product of the input length and the output length of the encoded function. (Typically, the complexity is also polynomial in the description length of f with respect to some computational model). As a result, even for simple functions, like GL, when the output length is linear in the input length, we do not have local REs with sub-quadratic complexity (let alone linear). We bypass this limitation by presenting a new construction of local REs for (slightly generalized) *parity circuits* whose gates compute only parity operations. The REs that we get are 3-local and their complexity equals to the size of the circuit. Hence we can efficiently handle any function that can be computed by a linear-size parity circuit.

Somewhat surprisingly, it turns out that this class of functions is more powerful than it seems.

In a sequence of works it was shown that linear-size parity-circuits can compute asymptotically-good error correcting codes [Spi96], pairwise independent hash functions [IKOS08], and, most relevant to us, GL-type list decodable codes [BIO14]. By plugging-in our construction we get the following theorem. (See Theorem 3.2 for a formal and slightly stronger statement.)

Theorem 1.6. *For any $s = O(n)$, there exists $t = O(n)$ and a function $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ with the following properties:*

1. *The collection $\{h(\cdot, w)\}_{w \in \{0,1\}^t}$ forms a pairwise-independent hash function.*
2. *The code $x \mapsto (h(x,w))_{w \in \{0,1\}^t}$ satisfies a Goldreich-Levin type list decoding properties: For any $\varepsilon > 0$, the code is list-decodable for radius $1 - 2^{-s} - \varepsilon$ with list of size $L = O(n/\varepsilon^2)$ and decoding time of $\mathrm{poly}(n)L$.*
3. *For any $w \in \{0,1\}^t$ the function $h_w(\cdot) := h(\cdot, w)$ has a perfect randomized encoding $\hat{h}_w$ with locality 3 and complexity of $O(n)$.*

The theorem provides an unusual example for an RE which *collectively* encodes all the outputs of a multi-output function (without encoding each output separately). As a complementary result we prove that the bilinear function $h(x,w)$ defined above (in which the value of w is not fixed and does not appear as part of the output) cannot be locally encoded with complexity smaller than $\Omega(ns)$ which is quadratic in the circuit size when $s = \Theta(n)$. To the best of knowledge, this is the first super-linear lower-bound for the complexity of RE.

1.2.4 Putting everything together

Theorem 1.6 allows us to prove Theorem 1.2 via the above list-decoding framework. To prove Theorem 1.4 we start with the well-known HILL transformation [HILL99] from OWF to PRG. For the case of almost-regular exponentially-hard OWFs $f : \{0,1\}^n \to \{0,1\}^{O(n)}$, one can use a simplified version of the transformation which essentially: (1) extracts (via hashing) randomness from the distribution of $x \xleftarrow{R} \{0,1\}^n$ conditioned on $f(x)$; (2) generates $\Omega(n)$ pseudorandom bits via the use of hardcore functions; and (3) applies a final hash function to get a pseudorandom distribution.[5] We can instantiate the reduction with (several copies of) the function h from Theorem 1.6, and then encode the *reduction* locally with the RE. Using the fact that an RE of a PRG is a PRG [AIK06], we get a new local PRG whose seed is only linearly larger than the original PRG (since the RE has linear complexity). As a result, exponential hardness is preserved.

1.3 Conclusion

We showed that, for CSPs with some regularity properties, one can locally reduce searching to gap-distinguishing both in the worst-case and in the average-case. Our results are based on three main insights: (1) Hardcore-functions can be used for establishing worst-case Gap-hardness in the exponential regime; (2) Parity circuits can be encoded locally and with an overhead proportional to their circuit size; and (3) Linear-size parity circuits can compute non-trivial combinatorial objects.

[5] Observe that the first two steps resemble the reduction from search-CSP to gap CSP. Indeed, our worst-case reduction was inspired by the analog cryptographic reduction.

Our work leaves several interesting open problems. First, can we base **gapETH** and **elPRG** on minimal assumptions (i.e., **ETH** and the existence of an exponentially-hard OWF in $\mathbf{NC^0}$)? One natural way to address this question is to reduce general CSPs to smooth-CSPs. More generally, for a CSP φ let us denote by w_i the number of assignments that violate exactly i the constraints. How do restrictions on the weight profile $w = (w_0, w_1, \ldots, w_m)$ affect the computational hardness of φ? Note that different promise problems (e.g., unique-CSP, gap-CSP, and smooth-CSP) can be all presented as putting some simple restrictions on w.

Moving to the domain of REs, it will be interesting to understand which functions can be encoded locally with linear complexity. Additionaly, we believe that REs of list-decodable codes (or hash functions) should be further explored. Currently, we do not have a clean abstraction of such objects and it is not fully clear under which circumstances list-decodable codes (or hash functions) can be replaced by their REs. Concretely, our proof of Theorem 1.2 adopts the outline sketched in Section 1.2 to work with REs in a somewhat ad-hoc way. This is very different from the cryptographic setting where one can prove that REs preserves the security of the encoded functions (e.g., "RE of PRG is a PRG"). Formulating similar "transference theorems" for other (worst-case) uses of REs remains an interesting challenge.

Organization. The rest of the paper is organized as follows. Following some preliminaries (Section 2), we describe the new RE construction and prove Theorem 1.6. We continue with sufficient conditions for **gapETH** and a proof of the new isolation lemma (Section 4), and end-up with sufficient conditions for **elPRG** (Section 5).

Acknowledgement. I am grateful to Irit Dinur for suggesting the question of basing **gapETH** on Goldreich's OWF. I also thank Oded Goldreich and Uri Feige for helpful discussions. Thanks are also due to Pasin Manurangsi for sharing the results of [CCK+17].

2 Preliminaries

The Hamming distance (resp., relative Hamming distance) between a pair of equal-length strings x, x' is the number (resp., fraction) of coordinates in which x and x' differ. We say that x is α-close (resp., α-far) from x' if their relative Hamming distance is at most α (resp., at least α). By default, logarithms are always taken to base 2. We let $H_2(\alpha) := -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ denote the binary entropy function, and often use the inequality $\binom{n}{\alpha n} \leq 2^{H_2(\alpha)n}$ to upper-bound the volume of an n-dimensional Hamming ball of radius αn. We use the following standard CSP terminology. The *value* of an assignment $x \in \{0,1\}^n$ for a CSP φ is defined to be the fraction of constraints that x satisfies. The *value* of the instance φ is the maximum, over all assignments x, of the fraction of satisfied constraints. We say that φ is γ-*unsatisfiable* if its value is at most $1-\gamma$.

2.1 Cryptographic Definitions

Let U_n denote the uniform distribution over $\{0,1\}^n$. An efficiently computable function $f : \{0,1\}^n \to \{0,1\}^{m(n)}$ is (T, ε) *one-way* if for every $T(n)$-time probabilistic algorithm A the *inversion probability*

$$\Pr_{y \xleftarrow{R} f(U_n)} [A(y) \in f^{-1}(y)]$$

is at most $\varepsilon(n)$. (Statements like this mean that we consider a family $f = \{f_n\}$ for growing input lengths, and we think of m, T and ε as functions of n. To simplify notation, we keep this convention implicit throughout the paper.) We say that f is exponentially-hard OWF if it is $(2^{\beta n}, 2^{-\beta n})$-OWF for some constant β. This definition naturally generalize to *collections of functions* $\mathcal{F} = \left\{f_w : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{w \in \{0,1\}^{t(n)}}$ by defining the inversion probability of an algorithm A to be

$$\Pr_{w \stackrel{R}{\leftarrow} U_{t(n)}, y \stackrel{R}{\leftarrow} f_w(U_n)}[A(w,y) \in f_w^{-1}(y)].$$

A distribution D_0 is (T,ε)-*indistinguishable* from a distribution D_1, if for every $T(n)$-time probabilistic algorithm A the *distinguishing advantage*

$$|\Pr[A(D_0) = 1] - \Pr[A(D_1) = 1]|$$

is at most $\varepsilon(n)$. The *statistical distance* between D_0 and D_1 is at most ε, if for every T, the distributions are (T,ε)-indistinguishable. (In this case, we say that the distribution are ε-statistically close). We say that $f : \{0,1\}^n \to \{0,1\}^{m(n)}$ is (T,ε) *pseudorandom* if (1) $m(n) > n$; and (2) the distributions $f(U_n)$ and $U_{m(n)}$ are (T,ε)-indistinguishable. We say that f is exponentially-secure PRG if it is $(2^{\beta n}, 2^{-\beta n})$-PRG for some constant β. A collection $\mathcal{F}$ is (T,ε)-PRG if the distributions $(w, f_w(U_n))$ and $(w, U_{m(n)})$, where $w \stackrel{R}{\leftarrow} U_{t(n)}$, are (T,ε)-indistinguishable.

We say that f is k-*local* if each of its outputs depends on at most k inputs. A function is *locally computable* (equivalently in $\mathbf{NC^0}$) if it is k-local for some constant k that does not grow with the input length. Similarly, a (possibly infinite) collection $\mathcal{F}$ of functions is local if all the functions in the collection are k-local for some universal constant.

2.2 Circuits

Let $\mathbb{F}$ be a finite field (by default the binary field). An $\mathbb{F}$-arithmetic circuit C over a set of input variables X, and a set of output variables Y is a directed acyclic graph as follows: Every vertex v in C is either of in-degree 0 (input gate) or of in-degree 2 (computation gate). Every vertex v of in-degree 0 is labelled by either a variable in X or a field element in $\mathbb{F}$. Every vertex v of in-degree 2 is labelled by either $\times$ (product gate) or $+$ (sum gate). A subset of the computation nodes are also labeled by output variables (each output variable appears in exactly one output node). Gates which are neither input gates nor output gates are called *internal gates*. For two gates u and v, if (u,v) is an edge in C, then u is called a *child* of v. The size of C, denoted $|C|$, is the number of gates in C. A circuit C is *skew* with respect to a subset $X' \subset X$ of the input variables, if every product gate in C have at least one child which is labeled by a constant or by a variable in X'. (This, in particular, implies that the computed function is affine in the variables $X \setminus X'$.) An arithmetic circuit over n input variables and m output variables defines a function $f : \mathbb{F}^n \to \mathbb{F}^m$ in the natural way.

2.3 Randomized Encoding of Functions

Roughly speaking, a *randomized encoding* [IK00, AIK06] of a function $f(x)$ is a randomized mapping $\hat{f}(x;r)$ such that for every input x the output distribution $\hat{f}(x;r)$ (induced by a random choice of r) depends only on the output of $f(x)$. Throughout the paper we employ *perfect randomized encoding* as defined below.

Definition 2.1 (Perfect Randomized Encoding)**.** *Let $f : \{0,1\}^n \to \{0,1\}^s$ be a function. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^{\hat{s}}$ is a* perfect randomized encoding *(PRE) of f if there exists a deterministic decoding algorithm* Dec *and a randomized simulator* Sim *which satisfy the following:*

- *(Perfect correctness) For every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $\mathsf{Dec}(\hat{f}(x;r)) = f(x)$.*
- *(Perfect privacy) For every $x \in \{0,1\}^n$, the distribution $\hat{f}(x;r)$, induced by a uniform choice of $r \xleftarrow{R} \{0,1\}^\rho$, is identical to the distribution $\mathsf{Sim}(f(x))$.*
- *(Balanced simulation) The distribution $\mathsf{Sim}(y)$ induced by choosing $y \xleftarrow{R} \{0,1\}^s$ is identical to the uniform distribution over $\{0,1\}^{\hat{s}}$.*
- *(Length preserving) The difference between the output length and the total input length of the encoding, $\hat{s} - (n + \rho)$, is equal to the difference, $s - n$, between the output length and the input length of f. Equivalently, the* randomness complexity *ρ equals to the difference between the* output complexity *$\hat{s}$ of $\hat{f}$ to the output length s of f.*

We refer to the second input of $\hat{f}$ as its random input *and define the complexity of the RE to be its randomness complexity ρ. (In the case of local perfect REs, this also measures the computational overhead of computing the RE compared to computing f.)*

Encoding collections. The definition naturally extends to the case where $\mathcal{F}$ is a collection of functions $\left\{f_z : \{0,1\}^{n(z)} \to \{0,1\}^{s(z)}\right\}_{z \in \{0,1\}^*}$. In particular, we say that the collection $\hat{\mathcal{F}}$, defined by $\left\{\hat{f}_z : \{0,1\}^{n(z)} \times \{0,1\}^{\rho(z)} \to \{0,1\}^{\hat{s}(z)}\right\}_{z \in \{0,1\}^*}$, perfectly encodes $\mathcal{F}$ if for every z, $\hat{f}_z$ perfectly encodes f_z. Furthermore, we always assume that the encoding is uniform in the sense that there exists a polynomial-time algorithm which given z outputs a description (say as a boolean circuit) of the encoding $\hat{f}_z$, its decoder Dec_z and its simulator Sim_z.

Combinatorial view. It is not hard to show (see [AIK06, Section 4]) that perfect REs satisfy the combinatorial structure defined in the introduction: The space of encodings $\{0,1\}^{\hat{s}}$ can be partitioned to 2^ρ size sets $\{D_y\}_{y \in \{0,1\}^s}$ such that for every x the mapping $g(r) = \hat{f}(x;r)$ forms a bijection from the randomness space $\{0,1\}^\rho$ to the set D_y. (The injectivity part is sometimes referred to as the unique randomness property). As a result, we get the following simple but useful claim.

Claim 2.2. *Let $h : \{0,1\}^n \to \{0,1\}^s$ be a function and let $\hat{h}(x,r) : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^{\hat{s}}$ be a perfect randomized encoding of h with decoder* Dec*. Then for any $\hat{y} \in \{0,1\}^{\hat{s}}$ and any subset $X \subset \{0,1\}^n$ the size of the set*

$$\left\{(x,r) : (x \in X) \wedge (x,r) \in \hat{h}^{-1}(\hat{y})\right\}$$

equals to the size of the set

$$X \cap h^{-1}(y),$$

where $y = \mathsf{Dec}(\hat{y})$.

2.4 Pairwise Independent Hashing and List-Decodable Codes

Let $h : X \times W \to Z$ be a two argument function. In the following we think of the first input (denoted by x) as the main input and on the second input (denoted by w) as a "key" or an "index". Correspondingly, we write $h_w(x) := h(x, w)$, and sometimes view h as a collection of functions $\{h_w : X \to Z\}_{w \in W}$. We say that h is *pairwise independent* hash function if for any $x \neq x' \in X$, for a random choice of $w \stackrel{R}{\leftarrow} W$, the joint distribution of the random variables $(h_w(x), h_w(x'))$ is uniform over Z^2.

We can also think of h as a code which maps an information word x into a codeword of length $|W|$ over the alphabet Z defined by $(h_w(x))_{w \in W}$. That is, h provides a direct access to every coordinate of the codeword. We say that a (possibly randomized) codeword oracle $\mathcal{O} : W \to Z \cup \{\perp\}$ is (δ, ε)-*correlated* with a codeword of $x \in X$ (or, in short, correlated with x) if

$$\forall w \in W, \Pr[\mathcal{O}(w) \neq \perp] \geq \delta \quad \text{and} \quad \Pr_{w \stackrel{R}{\leftarrow} W}[\mathcal{O}(w) = h(x, w) | \mathcal{O}(w) \neq \perp] \geq \frac{1}{|Z|} + \varepsilon.$$

We say that h is (δ, ε)-*list-decodable* with κ oracle calls and list size λ if there exists an oracle algorithm A with running time $\lambda \cdot \mathrm{poly}(\log(|X|))$ which, after at most κ oracle calls to an oracle $\mathcal{O}$ which is (δ, ε)-correlated with some $x \in X$, generates a set Λ of size at most λ, such that $\Pr[x \in \Lambda] \geq \frac{1}{2}$.

Linear-time constructions Ishai, Kushilevitz, Ostrovsky and Sahai [IKOS08] proved that there exists a pairwise hash function $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ which can be computed by a linear-size circuit which is skew with respect to the index argument. Baron, Ishai and Ostrovsky [BIO14] (building on Holenstein, Maurer, Sjödin [HMS04]) showed that a variant of the IKOS construction yields a locally-decodable code. In particular, the following theorem holds.

Theorem 2.3 ([IKOS08, BIO14])**.** *For any $s = O(n)$, there exists $t = O(n)$ and a function $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ with the following properties:*

1. *h is pairwise-independent hash function.*
2. *For any $\varepsilon, \delta > 0$ the code induced by h is (δ, ε)-list-decodable with complexity and list size of $O(n/\delta\varepsilon^2)$.*
3. *The function h can be computed by a linear-size circuit which is skew with respect to the second argument.*

3 Efficient Randomized Encoding for List-Decodable Codes

The following lemma (whose proof is deferred to Section 3.1) shows that linear circuits (or more generally skew circuits whose skew variables are public) can be encoded by a linear size circuits. For simplicity, the lemma is stated with respect to the binary field though the proof readily generalizes to arithmetic circuits over arbitrary finite fields.

Lemma 3.1. *Let $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ be a function which can be computed by a circuit of size S which is skew with respect to the second argument of h. Then, for every $w \in \{0,1\}^k$ the function $h_w(\cdot) = h(x, w)$ admits a 3-local PRE of complexity $\Theta(S)$. Moreover, the function $H : \{0,1\}^{n+t} \to \{0,1\}^{s+t}$ also admits such an encoding.*

Combining the above with Theorem 2.3, we derive Theorem 1.6 (re-stated below in a a more detailed form).

Theorem 3.2 (Theorem 1.6 restated)**.** *For any $s = O(n)$, there exists $t = O(n)$ and a function $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ with the following properties:*

1. *h forms a collection of pairwise-independent hash function.*
2. *For any $\varepsilon, \delta > 0$ the code induced by h is (δ, ε)-list-decodable with complexity and list size of $O(n/\delta\varepsilon^2)$.*
3. *For any $w \in \{0,1\}^t$ the function $h_w(\cdot) := h(\cdot, w)$ has a perfect randomized encoding $\hat{h}_w$ with locality 3 and complexity of $O(n)$. Moreover, this also holds for the function $H(x, w) = (h(x, w), w)$.*[6]

Remark 3.3. *We will sometimes employ Theorem 3.2 with varying output lengths s (e.g., s is chosen at random from $\{1, \ldots, n\}$). Note that as long as we have some linear upper-bound on the largest possible value of s (e.g., n), we get an upper-bound of cn, for some absolute constant $c > 0$, on the maximal complexity of the RE.*

It is natural to ask whether more general linear-size circuits admit local encodings with linear complexity. By adopting an argument from [AIK09], we show that there is a degree-2 function computable by a linear-size circuit that cannot be encoded locally with less than $\Omega(n^2)$ complexity. In fact, this holds for the IKOS pair-wise independent hash function $h(x, w)$ (for the case where the index w is not outputted and remains hidden).

Lemma 3.4. *If $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ is pairwise independent then any perfect encoding $\hat{h}(x, w; r)$ of h with locality k must have complexity of at least sn/k. In particular, the function h defined in Theorem 2.3, which is a degree-2 function with $O(n)$ circuit size, cannot be encoded with constant locality with complexity smaller than $\Omega(n^2)$.*

Lemma 3.4 forms the first quadratic separation between the circuit size and the complexity of its local encoding.

Proof. Let $\hat{h}(x, w; r)$ be a k-local encoding of $h(x, w)$ with randomness complexity of ρ and output complexity of $\hat{s} = s + \rho$. Consider the bipartite dependency graph of the function $\hat{h}$ in which there are $n + t + \rho$ input nodes and $\hat{s}$ output nodes and each output is connected to the inputs on which it depends. The graph has at most $\hat{s}k$ edges and therefore, by an averaging argument, there exists an input x_i which influences at most $\ell = \hat{s}k/n$ outputs. Fix some input x (say to 0^n). We claim that given $h(x, w)$ where $w \xleftarrow{R} \{0,1\}^t$, we can guess with probability $2^{-\ell}$ the value of $h(z, w)$ where z denotes the string x with its i-th bit flipped. Indeed, we can apply the simulator to $y = h(x, w)$ and get a string $\hat{y}$ which, for some r, equals to $\hat{y} = \hat{h}(x, w; r)$. Then, replace the bits which are influenced by the i-th input with a random ℓ-bit string and get, with probability at least $2^{-\ell}$, a string $\hat{z} = \hat{h}(z, w; r)$. Finally, apply the decoder and get $h(z, r)$. Since h is 2-wise independent $\ell \geq s$ and so the theorem follows. □

[6] The "Moreover part" will be useful for our cryptographic applications and will allow us to get a single local PRG (as opposed to collection of local PRGs).

3.1 Proof of Lemma 3.1

We prove the second (Moreover) part of the theorem by constructing a 3-local PRE $\hat{H}(x, w; r)$ of the form $(\hat{h}(x, w; r), w)$ whose randomness complexity equals to the number of internal gates in the skew circuit that computes H. The first part then follows by considering the encoding $\hat{h}_w(x; r) = \hat{h}(x, w; r)$.

The encoding: For every internal gate (which is not an input or an output gate) we allocate a random input r_i. For ease of notation, we also define "dummy" variables for input and output gates. Specifically, for an input gate i, we set r_i to the value of the corresponding label (an x variable, a w variable or a constant), and for an output gate i we let $r_i = 0$. For every non-input gate i whose children are the gates j and ℓ, we output the three local function $\hat{y}_i = r_j \diamond r_\ell - r_i$ where $\diamond$ stands for addition if i is an addition gate and for multiplication if i is a multiplication gate. We also output the values $w = (w_1, \ldots, w_t)$ of the "key" variables. Observe that the number of random inputs, ρ, is exactly the number of internal gates, and the number of outputs is exactly $\rho + s + t$, as required.

Correctness: Fix some input (x, w) and randomness r and let $(\hat{y}, w)$ denote the output of the encoding. Let $v_i = v_i(x, w)$ be the value induced on the i-th gate of H by the input (x, w). Our goal is to decode the value v_i of the i-th output wire of H given the encoding w and $\hat{y}$. For this it suffices to show that we can compute the value $g_i = v_i(x, w) - r_i$ for each gate i (since r_i is taken to be zero for output variables). We compute these values one-by-one by traversing the circuit in topological order as follows.

- If i is an input/constant gate then we observe that $g_i = 0$.
- If i is an addition gate whose children are the gates j and ℓ, we compute g_i by

  $$g_j + g_\ell + \hat{y}_i = v_j - r_j + v_\ell - r_\ell + r_j + r_\ell - r_i = v_j + v_\ell - r_i = v_i - r_i,$$

  where $\hat{y}_i$ is the output of the encoding which is associated with the i-th gate of H.
- If i is a multiplication gate whose children are the gates j and ℓ, then ℓ must be an input gate whose value v_ℓ is known as part of the encoding (i.e., it is one of the w variables or just a field constant). We therefore output the value $v_\ell \cdot g_j + \hat{y}_i$. Recalling that $v_\ell = r_\ell$, the output simplifies to $v_\ell(v_j - r_j) + r_j \cdot v_\ell - r_i = v_\ell v_j - r_i = v_i - r_i$.

Privacy: Let M denote the set of internal gates in H (which are neither input nor output gates) and let O denote the set of output gates. Given (y, w) the simulator samples an encoding as follows: (1) Sample $(\hat{y}_i)_{i \in M}$ uniformly at random; (2) Apply the decoder to $w, (\hat{y}_i)_{i \in M}$ and compute, for each internal gate i the value g_i, and for each output gate o set $\hat{y}_o = g_i + g_j - y_o$ where i and j are the children of the o-th output. (3) Output $(\hat{y}, w)$.

First observe that the simulator maps the uniform distribution over $\{0,1\}^{s+t}$ to the uniform distribution over $\{0,1\}^{\rho+s+t}$, and therefore it is balanced. Next, we prove that the simulator perfectly simulates the encoding. Fix some x and w. Consider the distribution $\hat{H}(x, w; r) = (\hat{y}, w)$ induced by a random choice of r. Observe that the joint distribution of $(\hat{y}_i)_{i \in M}$ is uniform since each $\hat{y}_i$ can be written as $f_i(w, x, r) - r_i$, where f_i does not depend on r_i. Hence, the simulator perfectly samples the prefix $(\hat{y}_i)_{i \in M}$. Moreover, both in the simulation and in the actual encoding the suffix $((\hat{y}_o)_{o \in O}, w)$ is uniquely determined by the prefix of the encoding $(\hat{y}_i)_{i \in M}$ and by the output value $(y, w) = H(x, w)$, and therefore the two distributions are identical. □

4 Sufficient conditions for gapETH

4.1 fasETH implies gapETH

In this section we base Gap-ETH on the assumption that it is exponentially hard to solve φ even under the promise that φ has only "few almost-satisfying" assignments.

Assumption 4.1 (fasETH). *The* $\mathbf{fasETH}(k, a, \beta, \gamma, \alpha)$ *assumption asserts that there is no* $2^{\beta n}$*-time algorithm that solves* k*-CSPs over* n *variables and* $m \leq an$ *constraints with at most* $2^{\alpha n}$ *assignments of value larger than* $1-\gamma$*. The* **fasETH** *assumption asserts that* $\mathbf{fasETH}(k, a, \beta, \gamma, \beta/5)$ *holds for some integer* k *and some constants* a, β, γ*.*

Lemma 4.2. **fasETH** *implies* **gapETH***. In particular, if* $\mathbf{fasETH}(k, a, \beta, \gamma, \beta/5)$ *holds then, for some* $\gamma' > 0$*, there is no* $2^{\beta n/5}$*-time algorithm that distinguishes between satisfiable* k*-CSPs and* γ'*-unsatisfiable* k*-CSPs.*

Proof. Let $b = \beta/5$ and let $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{bn}$ be the list decodable code promised by Theorem 3.2. For any given $w \in \{0,1\}^t$, let $h_w(x) := h(x, w)$ and let $\hat{h}_w(x; r)$ be a perfect randomized encoding of h_w with randomness complexity of cn and output complexity of $(b + c)n$ for a constant $c = c(b)$. (In fact, by Remark 3.3, c can be treated as a universal constant independent of b.)

Assume, towards a contradiction, that there exists a $2^{\beta n/5}$-time algorithm A which distinguishes between satisfiable k-CSPs and γ'-unsatisfiable k-CSPs for some constant $\gamma' < \frac{\gamma}{a+b+c}$ (additional restrictions on γ' will be added later). We use A to find a satisfying assignment for a k-CNF φ over n variables and $m \leq an$ constraints which has no more than $2^{\beta n/5}$ assignments of value larger than $1-\gamma$. Specifically, we show how to convert A to an oracle $\mathcal{O}$ which is $(\delta \geq 2^{-bn}, \varepsilon \geq 2^{-\beta n/4})$-correlated with the h-codeword of a satisfying assignment $x \in \{0,1\}^n$ of φ. Each query to $\mathcal{O}$ will be emulated in time $\mathrm{poly}(n)$ and a single call to the algorithm A on a k-CSP over $n' = (1+b)n$ variables. Therefore, we can then recover x in time $\mathrm{poly}(n) \cdot 2^{\frac{\beta n'}{5}} \cdot \frac{1}{\delta \varepsilon^2} = \mathrm{poly}(n) 2^{(0.9\beta + \beta^2/25)n} < 2^{\beta n}$, in contradiction to our hypothesis.

The oracle $\mathcal{O}$:

- Given $w \in \{0,1\}^t$, sample $\hat{z} \overset{R}{\leftarrow} \{0,1\}^{(b+c)n}$ and define a k-CSP $\psi_{w,\hat{z}}$ over $n' = (1+b)n$ variables $x = (x_1, \ldots, x_n), r = (r_1, \ldots, r_{cn})$ and $m' = m + (b+c)n \leq (a+b+c)n$ constraints by $\psi_{w,\hat{z}} := \varphi \wedge (\hat{h}_w(x; r) = \hat{z})$.
- If A claims that the value of $\psi_{w,\hat{z}}$ is smaller than $1-\gamma'$ output $\perp$; Otherwise, output $z = \mathsf{Dec}_w(\hat{z})$ where Dec_w is the decoder of the encoding $\hat{h}_w$.

Analysis. The running time of $\mathcal{O}$ is evident from its description. Fix some assignment $x \in \{0,1\}^n$ that satisfies φ. We claim that the oracle is (δ, ε)-correlated with x. Fix w and let $z = h_w(x)$. First, observe that if $\hat{z}$ hits the set $G = \left\{ \hat{h}_w(x; r) : r \in \{0,1\}^{cn} \right\}$ then, by perfect correctness, the algorithm outputs the correct answer. Also, recall that, since the encoding is perfect, the size of the set G is exactly 2^{cn}. We conclude that $\delta \geq |G|/2^{(b+c)n} \geq 2^{-bn}$, as required.

We move on to analyze ε. Let $\hat{Z} \subset \{0,1\}^{(b+c)n}$ be the set of strings $\hat{z}$ for which $\psi_{w,\hat{z}}$ is $(1-\gamma')m'$ satisfiable. Observe that $\hat{z} \in \hat{Z}$ only if there exists an assignment (x', r') such that (1) x' satisfies at least $m - \gamma' m'$ of the constraints of φ and (2) the string $\hat{z}$ differs from the string $\hat{h}_w(x'; r')$ in at most

$\gamma' m'$ coordinates. Recall that $\gamma' < \frac{\gamma}{a+b+c}$, and therefore, by assumption, (1) holds for at most $2^{\beta n/5}$ strings x'. Denoting this set by X, it holds, by the unique randomness property of the perfect RE, that the set $\hat{Z}_0 = \left\{\hat{h}_w(x;r) : x \in X, r \in \{0,1\}^{cn}\right\}$ is of size at most $2^{(c+\beta/5)n}$. Since $\hat{Z}$ consists of all the strings of length $(b+c)n$ which differ in at most $\gamma' m' \leq \gamma'(a+b+c)n$ locations from some string in $\hat{Z}_0$, it follows that

$$|\hat{Z}| \leq |\hat{Z}_0| \cdot 2^{H_2(\frac{\gamma'(a+b+c)}{b+c})(b+c)n} \leq 2^{n(c+\beta/5+H_2(\frac{\gamma'(a+b+c)}{b+c})(b+c))},$$

which is upper-bounded by $2^{\beta n/4+cn}$ for sufficiently small γ'. We conclude that $\varepsilon = |G|/|\hat{Z}| \geq 2^{-\beta n/4}$, as required. □

4.2 Smooth-ETH implies Gap-ETH

In this section we base Gap-ETH on the assumption that it is exponentially hard to solve φ under the promise that the number of "almost-satisfying" assignments of φ is not "much larger" than the number of satisfying assignments.

For a CSP φ and a real number $\varepsilon \in [0,1]$, we let $\mathsf{sat}_\varepsilon(\varphi)$ denote the number of assignments that satisfy at least ε-fraction of the constraints of φ. In particular, $\mathsf{sat}_1(\varphi)$ is simply the number of satisfying assignments. We say that φ is (γ,α)-*smooth* if

$$\frac{\mathsf{sat}_{1-\gamma}(\varphi)}{\mathsf{sat}_1(\varphi)} \leq 2^{\alpha n}.$$

Assumption 4.3 (smETH). *The* $\mathbf{smETH}(k,a,\beta,\gamma,\alpha)$ *asserts that there is no* $2^{\beta n}$*-time algorithm for solving* k*-CSPs over* n *variables and* an *constraints which are* (γ,α)*-smooth. The* **smETH** *asserts that* $\mathbf{smETH}(k,a,\beta,\gamma,\alpha)$ *holds for some integer* k *and constants* a,γ,β *and* $\alpha < \beta/5$*.*

Our goal is to show that **smETH** implies **gapETH**. By Lemma 4.2, it suffices to derive **fasETH** from **smETH**. We achieve this goal via the following procedure which sparsifies the set of almost-satisfying assignments of a CSP φ by a factor of roughly $\mathsf{sat}_1(\varphi)$.

Construction 4.4 (The transformation A)**.** *Given a* k*-CSP* φ *over* n *variables and* m *constraints output a* $\max(k,3)$*-CSP* φ' *over* $n+O(n)$ *variables and* $m+O(n)$ *constraints defined as follows:*

1. *Sample* $s \in \{2,\ldots,n+1\}$*, and let* $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^s$ *be the pair-wise independent hash function promised by Theorem 3.2. Sample a string* $y \xleftarrow{R} \{0,1\}^s$ *and a key* $w \xleftarrow{R} \{0,1\}^k$ *for* h*.*
2. *Output*

$$\varphi' = \left(\varphi \wedge (\hat{h}_w(x;r) = \hat{y})\right),$$

where $\hat{h}_w$ *is the 3-local perfect RE of* $\hat{h}_w$ *and* $\hat{y}$ *is sampled by applying the encoding's simulator to* y*.*

Since φ' contains φ as a sub-formula, the transformation preserves unsatisfiability.

Lemma 4.5. *For every choice of randomness, if* φ *is unsatisfiable so is* φ'*. Moreover, a satisfying assignment* x' *for* φ' *can be efficiently transformed (by projection) to a satisfying assignment* x *for* φ*.*

We also prove the following isolation property.

Lemma 4.6. *If φ is satisfiable and $s = \lceil \log(\mathsf{sat}_1(\varphi)) \rceil + 1$ then, with probability of at least 1/8 over the choice of $(w, \hat{y})$, the CSP φ' has a unique satisfying assignment.*

Proof. Let S be the set of assignments that satisfy φ. Fix an arbitrary[7] string $y \in \{0,1\}^s$. Valiant and Vazirani [VV86] proved that if $|S| \in [2^{s-2}, 2^{s-1}]$ then, with probability at least $1/8$ over the choice of $w \stackrel{R}{\leftarrow} \{0,1\}^t$, it holds that

$$\exists \text{ unique } x \in S \text{ such that } h_w(x) = y. \tag{1}$$

By Claim 2.2, for every w which satisfies (1), there exists a unique $x \in S$ and a unique r for which $\hat{h}_w(x, r) = \hat{y}$. The lemma follows. □

Theorem 1.5 follows by noting that, for a satisfiable CSP φ, a random s equals to $\lceil \log(\mathsf{sat}_1(\varphi)) \rceil + 1$ with probability $\Omega(1/n)$.

In the following lemma, we show that A sparsifies the set of almost-satisfying assignments.

Lemma 4.7. *For every constants $a > 0$, $\gamma \in [0,1]$ and $\varepsilon > 0$ there exists a constant $\gamma' > 0$ for which the following holds. Suppose that φ has $m = an$ constraints, and that $s = \lceil \log(\mathsf{sat}_1(\varphi)) \rceil + 1$. Then, with probability $1 - o(1)$ over the choice of w and $\hat{y}$,*

$$\mathsf{sat}_{1-\gamma'}(\varphi') \leq 2^{\varepsilon n} \frac{\mathsf{sat}_{1-\gamma}(\varphi)}{\mathsf{sat}_1(\varphi)}.$$

Proof. Fix a, γ and ε. Let S be the set of satisfying assignments of φ and let B be the set of assignments that satisfy at least $1 - \gamma$ fraction of the constraints in φ. Recall that $s = \lceil \log(|S|) \rceil + 1$. Let $\hat{s}$ denote the output length of the encoding $\hat{h}_w$ and let ρ denote its randomness complexity. Recall that $\rho = \Theta(n)$ and that $\hat{s} = \rho + s = \Theta(n)$, and note that φ' is a CSP over $n + \rho$ variables and over $an + \hat{s}$ constraints. Let $\delta > 0$ be a constant for which $\binom{\hat{s}}{\delta\hat{s}} \leq 2^{\varepsilon n/2}$ and let $\gamma' > 0$ be a constant smaller than both $(\gamma a)/(a + \hat{s}/n)$ and $(\delta\hat{s}/n)/(a + \hat{s}/n)$.

By the choice of γ', it follows that if an assignment (x, r) violates at most γ'-fraction of the constraints in φ' then

$$x \in B \quad \text{and} \quad \hat{h}_w(x, r) \text{ is } \delta\text{-close to } \hat{y}. \tag{2}$$

To prove the lemma, we show that, with high probability over the choice of w and $\hat{y}$, the number of such assignments is at most $|B| 2^{\varepsilon n - s} \leq 2^{\varepsilon n} |B|/|S|$. For $z \in \{0,1\}^s$ let

$$B_z = \{x \in \{0,1\}^n : x \in B \wedge h_w(x) = z\}.$$

Observe that, for each z, the expected size of B_z (over the choice of w) is $\frac{|B|}{2^s}$. Call z *heavy* if the size of B_z is larger than its expectation by a multiplicative factor of $2^{\varepsilon n/2}$. By Markov's inequality, each z is likely to be heavy with probability of at most $2^{-\varepsilon n/2}$, and so the expected number of heavy z's is at most $2^{s - \varepsilon n/2}$. Call w *good* if the total number of heavy z's is at most $2^{s - \varepsilon n}$. By another application of Markov's inequality, almost all w's (except for a negligible fraction) are good.

Fix some good w. Call $\hat{y}$ *good* if it lands within a distance of at least $\delta\hat{s}$ from any string $\hat{z}$ which decodes to a heavy string z. Recall that $\hat{y}$ is sampled independently from w by applying

[7] Indeed, for this claim, we could use a simpler variant of A which sets y to be the fixed zero string and sets $\hat{y}$ to be some fixed encoding of it.

the simulator on a uniformly chosen $y \xleftarrow{R} \{0,1\}^s$. Since the encoding is perfect, this means that $\hat{y}$ is uniformly distributed over $\{0,1\}^{\hat{s}}$. Therefore, the probability that $\hat{y}$ lands δ-close to some string $\hat{z}$ which decodes to a heavy string z is at most

$$\frac{2^{s-\varepsilon n} \cdot 2^{\rho} \cdot \binom{\hat{s}}{\delta\hat{s}}}{2^{\hat{s}}} \leq 2^{s-\varepsilon n+\rho+\varepsilon n/2-\hat{s}} = 2^{-\varepsilon n/2}.$$

It follows that that, except with negligible probability, $\hat{y}$ is good.

We can now complete the argument. Fix some good w and $\hat{y}$. The number of assignments (x, r) that satisfy (2) is at most

$$\begin{aligned}\sum_{\hat{z}\in\{0,1\}^{\hat{s}}:\Delta(\hat{z},\hat{y})\leq\delta} \left|\left\{(x,r) : x \in B \wedge \hat{h}_w(x,r) = \hat{z}\right\}\right| &\leq \sum_{\hat{z}\in\{0,1\}^{\hat{s}}:\Delta(\hat{z},\hat{y})\leq\delta} |B_{\mathsf{Dec}(\hat{z})}| \\ &\leq \binom{\hat{s}}{\delta\hat{s}} |B| 2^{(\varepsilon n/2)-s} \\ &\leq |B| 2^{\varepsilon n - s},\end{aligned}$$

where Dec denotes the decoder of the RE, the first inequality follows from Claim 2.2, the second inequality follows from the goodness of $\hat{y}$ and the last inequality follows from our choice of δ. □

We can now prove that **smETH** implies **fasETH**.

Theorem 4.8. **smETH** *implies* **fasETH**.

Combined with Lemma 4.2, Theorem 4.8 implies Theorem 1.2.

Proof. Suppose that $\mathbf{smETH}(k, a, \beta, \gamma, \alpha)$ holds for some integer k and constants a, γ, β and $\alpha < \beta/5$. Let $\varepsilon = \frac{1}{2}(\beta/5 - \alpha)$ and let cn (resp., σn) be an upper-bound on the number of variables (resp., constraints) added by the transformation A (defined in Construction 4.4) when applied to a CSP with n variables. We will prove that $\mathbf{fasETH}(k, a', \beta', \gamma', \beta'/5)$ holds for the constants $a' = a + \sigma$, $\beta' = (\beta - \varepsilon)/(1 + c)$, some constant $\gamma' > 0$.

Assume, towards a contradiction, that $\mathbf{fasETH}(k, a', \beta', \gamma', \beta'/5)$ does not hold. We show that, in time $2^{\beta n}$, it is possible to find a satisfying assignment for any (γ, α)-smooth satisfiable k-CSP φ over n variables and $m = an$ constraints. First apply A to φ and get a k-CSP φ' over $n' = (1 + c')n$ variables and $m' \leq m + \sigma n \leq a'n'$ constraints where $0 \leq c' \leq c$. We condition on the event that the parameter s is chosen to be $\lceil \log(\mathsf{sat}_1(\varphi)) \rceil + 1$, which happens with probability $1/n$. By Lemmas 4.6 and 4.7, with constant probability over the choice of $(w, \hat{y})$, the resulting CSP φ' is satisfiable and has at most

$$2^{\varepsilon n} \frac{\mathsf{sat}_{1-\gamma}(\varphi)}{\mathsf{sat}_1(\varphi)} \leq 2^{\varepsilon n + \alpha n} = 2^{n'(\alpha+\varepsilon)/(1+c')} \leq 2^{n'\beta'/5}$$

assignments that satisfy more than $1 - \gamma'$ of the constraints for some $\gamma'(\varepsilon, a) > 0$. Assuming that $\mathbf{fasETH}(k, a', \beta', \gamma', \beta'/5)$ does not hold, we can find, with probability $\frac{1}{2}$, a satisfiable assignment x' for φ' in time $2^{\beta' n'}$ and, by Lemma 4.5, project it into a satisfying assignment x to φ. Overall, the algorithm has a success probability of $\Omega(1/n)$, and so by standard repetition, we can increase the success probability to $\frac{1}{2}$. The total running time is $\mathrm{poly}(n) \cdot 2^{\beta' n'} \leq \mathrm{poly}(n) \cdot 2^{(\beta-\varepsilon)n} < 2^{\beta n}$, contradicting $\mathbf{smETH}(k, a, \beta, \gamma, \alpha)$. □

4.3 Exponentially-hard Local One-way Functions imply Gap-ETH

We prove that **smETH** follows from the existence of locally-computable function which is *weakly hard* to invert for $2^{\Omega(n)}$-time algorithms. In fact, the theorem holds even for the case of $\mathbf{NC^0}$ collections.

Theorem 4.9. *If there exists a collection of $\mathbf{NC^0}$ functions $\mathcal{F} = \left\{f_w : \{0,1\}^n \to \{0,1\}^{O(n)}\right\}$ which cannot be inverted by $2^{\Omega(n)}$-time algorithms with inversion probability better than $1-2^{-o(n)}$, then **smETH** holds.*

To prove the theorem we show that, for any function f, the CSP distribution induced by inverting f on a random image $y \xleftarrow{R} f(U_n)$, is almost surely smooth.

Claim 4.10. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function, $\gamma, \alpha > 0$ be positive reals and $\varepsilon = \alpha - (m/n)H_2(\gamma)$. Then, with all but $2^{-\varepsilon n}$ probability over $y \xleftarrow{R} f(U_n)$, the constraint satisfaction problem $f(x) = y$ is (γ, α)-smooth.*

Proof. Let $w(y) = \Pr[y = f(U_n)]$ denote the weight of $y \in \{0,1\}^m$ under the distribution $f(U_n)$. We define a directed graph over the image of f where $y' \to y$ is an edge if y is γ-close to y' and $w(y) > w(y')2^{\alpha n}$. Note that the latter condition makes the graph acyclic. Also, by definition, the CSP $f(x) = y$ is (γ, α)-smooth if and only if y is a sink in the graph. We would like to show that all but $r = 2^{-\varepsilon n}$ fraction of the weight is assigned to sinks. For this, it suffices to show that for each sink y,

$$\sum_{y' : \exists \text{ path from } y' \text{ to } y} w(y') \leq w(y) \cdot \frac{r}{1-r}.$$

Decomposing the sum according to the length i of the path from y' to y, and noting that the in-degree of each node is upper-bounded by $2^{mH_2(\gamma)}$, we can upper-bound the LHS by

$$\sum_{i \geq 1} (2^{mH_2(\gamma)})^i w(y) (2^{-\alpha n})^i = w(y) \sum_{i \geq 1} (2^{mH_2(\gamma) - \alpha n})^i,$$

using the standard formula for the sum of infinite geometric series, we get an upper-bound of $w(y)\frac{r}{1-r}$, as required. □

We can now prove the theorem.

Proof of Theorem 4.9. Let $\mathcal{F} = \{f_w : \{0,1\}^n \to \{0,1\}^{an}\}_{w \in \{0,1\}^{t(n)}}$ be a collection of k-local which cannot be inverted by $2^{\beta n}$-time algorithms with inversion probability better than $1 - 2^{-o(n)}$. Fix some $\beta' < \beta$ and some $\alpha < \beta'/5$ and let $\gamma > 0$ be a constant for which $H_2(\gamma) < \alpha/a$. Set $\varepsilon = \alpha - aH_2(\gamma) > 0$. We will show that if $\mathbf{smETH}(k, a, \beta', \gamma, \alpha)$ does not hold then, for every $w \in \{0,1\}^t$, the function f_w can be inverted in time $2^{\beta n}$ with probability $1 - 2^{-\Omega(n)}$. Fix some w and let $f = f_w$. By Claim 4.10, with probability $1 - 2^{-\varepsilon n}$ over $y \xleftarrow{R} f(U_n)$, the k-CSP $f(x) = y$ is (γ, α)-smooth. Hence, an algorithm which violates $\mathbf{smETH}(k, a, \beta', \gamma, \alpha)$ immediately yields a $2^{\beta' n}$-time algorithm A that, with probability $1 - 2^{-\varepsilon n}$ over $y \xleftarrow{R} f(U_n)$, finds a preimage $x' \in f^{-1}(y)$ with probability $1/2$ (over its internal coin tosses). A standard repetition of $\mathrm{poly}(n)$ times yields an algorithm that inverts the function with all but $2^{-\Omega(n)}$ probability in time $\mathrm{poly}(n)2^{\beta' n} < 2^{\beta n}$. The theorem follows. □

4.4 Random CNFs are smooth

As mentioned in the introduction, Goldreich's function is conjectured to be exponentially hard to invert and thus implies **gapETH**. (Another coding-based candidate appears in Section 5). In this section, we present other sufficient conditions based on the conjectured hardness of random CNFs.

Let $U_{n,m,k}$ denote the *uniform distribution* over k-CNFs with n variables and m constraints where the clauses are chosen uniformly, independently and without replacement among all $\binom{n}{k}2^k$ non-trivial clauses of length k, i.e., clauses with k distinct, non-complementary literals. It is believed that $U_{n,m,k}$ is hard-to-satisfy when the clause density $r = m/n$ is slightly below the satisfiability threshold which is about $2^k \ln 2$. (See [AC08, ACR11] and the survey [Ach09]). We show that such instances are likely to be smooth, and therefore the exponential hardness of this distribution implies **gapETH**.

Lemma 4.11. *There is a sequence of $\varepsilon_k \to 0$ for which the following holds for every $k \geq 8$ and every $r < (1-\varepsilon_k)2^k \ln 2$. For every $\gamma > 0$, a randomly chosen k-CNF $\varphi \xleftarrow{R} U_{n,m=\lceil rn \rceil,k}$ is, with high probability, (α, γ)-smooth where $\alpha = k2^{3-k} \ln 2 + r(3H_2(\gamma) - \gamma \log(1 - 2^{-k}))$.*

Proof. Let $\varphi \xleftarrow{R} U_{n,m=\lceil rn \rceil,k}$ and define $\mu_m = 2^n(1-2^{-k})^m$ to be the expected number of satisfying assignment of φ. First we upper-bound the number of $(1-\gamma)$-satisfying assignments. Fix some subset $S \subset [m]$ of size $m' = (1-\gamma)m$ and consider the restriction φ_S of φ to the clauses in S. By Markov's inequality, the probability that φ_S has more than $\mu_{m'}B$ satisfying assignments is at most $1/B$. Taking $B = 2^{2H_2(\gamma)m}$, and applying a union-bound over all S's, we conclude that, with high probability,

$$\mathsf{sat}_{1-\gamma}(\varphi) < \binom{m}{(1-\gamma)m} B\mu_{m'} \leq 2^{3H_2(\gamma)m}\mu_{m'}.$$

On the other hand, Lemma 22 of [AC08] shows that (when r satisfies the above conditions), with high probability,

$$\mathsf{sat}_1(\varphi) \geq \mu_m 2^{-(\ln 2)k2^{3-k}n}.$$

By a union bound, we conclude that, with high probability,

$$\frac{\mathsf{sat}_{1-\gamma}(\varphi)}{\mathsf{sat}_1(\varphi)} \leq 2^{\alpha n}$$

where

$$\alpha = k2^{3-k} \ln 2 + r(3H_2(\gamma) - \gamma \log(1 - 2^{-k})),$$

as required. □

Noting that α goes to zero when k increases and γ decreases, we derive the following theorem.

Theorem 4.12. *Suppose that there exist constants $\beta > 0$ and $\varepsilon > 0$ such that for all sufficiently large k's, there is no $2^{\beta n}$-time algorithm that with constant probability finds a satisfying assignment for $\varphi \xleftarrow{R} U_{n,m,k}$ where $m = n(1-\varepsilon)2^k \ln 2$. Then,* **gapETH** *holds.*

The hypothesis assumes that k-CNFs do not become "easier" when k grow. This looks conservative since the complexity of known exponential-time algorithm actually grows with k.

Proof. By Theorem 4.8 it suffices to show that **smETH** holds. Fix some constant $\alpha < \beta/5$. By Lemma 4.11, for sufficiently large k and sufficiently small γ, it holds that $1 - \varepsilon < (1 - \varepsilon_k)$, and therefore $\varphi \xleftarrow{R} U_{n,m=\lceil rn \rceil,k}$ is, with high probability, (γ, α)-smooth. If $\mathbf{smETH}(k, a = (1 - \varepsilon)2^k \ln 2, \beta, \gamma, \alpha)$ does not hold then we can solve, with high probability, a random k-CNF sampled from $U_{n,m,k}$ in time $2^{\beta n}$, in contradicting the theorem's hypothesis. □

4.5 Local OWF from Planted CNFs

Another candidate hard distribution over k-CNFs is the *planted* distribution (again with clause-to-variables density slightly below the satisfiability threshold). We show that under this hardness assumption, one can get a local OWF. (The results of Achlioptas, Coja-Oghlan [AC08] establish some relation between planted CNFs and random CNFs, however, it is not clear to us whether the relation is strong enough to carry over exponential hardness.)

The planted distribution. Let $P_{n,m,k}$ denote the *planted distribution* over k-CNFs with n variables and m constraints where a k-CNF is chosen by first selecting a random assignment $x \xleftarrow{R} \{0,1\}^n$ and then selecting each clause uniformly, independently and without replacement among all $\binom{n}{k}(2^k - 1)$ non-trivial clauses of length k that are satisfied by x.

We note that the intractability of $P_{n,m,k}$ easily yields a locally-computable collection of one-way function $\left\{ f_w : \{0,1\}^{n+m'k} \to \{0,1\}^{m'(k+1)} \right\}$ where $m' = m/(1 - 2^{-k})$. The collection is defined as follows:

- Public index: m' random distinct k-subsets of $[n] := \{1, \ldots, n\}$, denoted by $w = (S_1, \ldots, S_{m'})$. We assume that each set S_i is ordered and let $S_{i,j}$ denote the j-th entry of S_i.
- Private input: $x \in \{0,1\}^n$ and m' random k-bit strings $z_1, \ldots, z_{m'}$. We let $z_{i,j}$ denote the j-th bit of z_i.
- The output of the function is parsed a m' blocks of length $k + 1$ where the i-block equals to $(1 z_i)$ if the boolean expression $(z_{i,1} - x_{S_{i,1}}) \vee \ldots \vee (z_{i,k} - x_{S_{i,k}})$ evaluates to true; and the output is 0^{k+1} otherwise.

Clearly, for every fixing of w, the function f_w is $2k$-local. Note that for uniformly chosen w, x and z, the distribution $(w, y = f_w(x, z))$, conditioned on seeing b blocks of zeroes, is just an encoding of a sample $\varphi_{w,y}$ from $P_{n,m-b,k}$. Moreover, if (x, z) is a preimage of y under f_w then the assignment x satisfies $\varphi_{w,y}$. Note that b takes the value $2^{-k}m'$ with $\Omega(1/\sqrt{n})$ probability. We conclude that if T-time algorithms cannot solve a random instance $\varphi \xleftarrow{R} P_{n,m,k}$ with probability better than ε, then the collection cannot be inverted in time T with probability better than $\varepsilon/\sqrt{n}$. If ε is sub-exponential in n, then so is ε'. By Theorem 4.9, we conclude that the exponential intractability of the planted distribution implies **gapETH**.

5 Cryptographic Applications

5.1 Exponentially-strong local PRGs with linear stretch

In this section we show that an exponentially-strong PRG with linear stretch and constant locality can be based on an "almost"-regular $2^{\beta n}$-hard one-way function $f : \{0,1\}^n \to \{0,1\}^{O(n)}$ in $\mathbf{NC^0}$. We begin with a *non-local* construction which can be viewed as a simplified version of the HILL construction [HILL99] (see also [Gol01, Chapter 3.5.2]).[8] The idea is to append to $y = f(x)$ about $s_1 = \sigma n$-bits of entropy y_1 from the input via hashing (where each image of f is assumed to have at least $2^{\sigma n}$ preimages), to append $\Omega(\beta n)$ hardcore bits y_2, and to apply a randomness extractor to y and y_1 in order to recover almost all n bits of entropy from x. Formally, we employ the following construction.

Construction 5.1. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function and s_1, s_2, s_3 be parameters. Instantiate Theorem 3.2 with input n and output lengths s_1, s_2, s_3 and denote the resulting functions by $h : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{s_1}$, $C : \{0,1\}^n \times \{0,1\}^{t_2} \to \{0,1\}^{s_2}$ and $E : \{0,1\}^{n+s_1} \times \{0,1\}^{t_3} \to \{0,1\}^{s_3}$. Define the function*

$$G^f : \{0,1\}^{n+t_1+t_2+t_3} \to \{0,1\}^{t_1+t_2+t_3+s_2+s_3}$$

as follows:

$$G^f(x, w_1, w_2, w_3) = (w_1, w_2, w_3, E_{w_3}(f(x), h_{w_1}(x)), C_{w_2}(x)).$$

We say that $f : \{0,1\}^n \to \{0,1\}^{an}$ is (σ_0, σ_1)-*regular* if for every y in the image of f,

$$\sigma_0 \leq \frac{1}{n} |\log |f^{-1}(y)| \leq \sigma_1.$$

We say that f is α-*almost regular* if it is $(\sigma, \sigma + \alpha)$-regular for some $\sigma \geq 0$.

Theorem 5.2. *Let $f : \{0,1\}^n \to \{0,1\}^{an}$ be a $2^{\beta n}$-hard one-way function which is $(\sigma, \sigma + \beta/6 - \gamma)$-regular for some positive constants σ and $\gamma > 0$. Then, the function G^f (defined in Construction 5.1) yields an exponentially-strong PRG with linear stretch whenever*

$$\begin{aligned}
s_1 &\in [n(\sigma - \gamma/4), n(\sigma - \gamma/8)] \\
s_2 &\in [n(\beta/6 - \gamma/4), n(\beta/6 - \gamma/8)] \\
s_3 &\in [s_1 + n(1 - \sigma - \beta/6 + \gamma - \gamma/4), s_1 + n(1 - \sigma - \beta/6 + \gamma - \gamma/8)].
\end{aligned}$$

The constant 8 is arbitrary and can be replaced by any constant larger than 4. The theorem (whose proof is postponed to Section 5.3) yields the following result.

Corollary 5.3 (Theorem 1.4 restated)**.** *If there exists an $\mathbf{NC^0}$ function $f : \{0,1\}^n \to \{0,1\}^{an}$ which is $2^{\beta n}$-hard one-way function and is α-almost regular for $\alpha < \beta/6$ then there exists an exponentially-strong PRG in $\mathbf{NC^0}$ with linear stretch. In fact, such a PRG exists even with locality 4.*

We mention that 4-local PRGs cannot have super-linear stretch even if one requires only polynomial hardness [MST06]. Also, let us mention that given an exponentially-secure PRG with constant locality and linear stretch of $m = (1 + \varepsilon)n$ for some constant $\varepsilon > 0$, one can get (e.g., via a tree-like construction) a PRG with linear stretch of $m = Cn$ for arbitrary constant $C > 0$, while preserving exponential hardness, and at the expense of increasing the locality to a larger constant.

[8] The almost-regularity and the exponential hardness assumptions makes the OWF-to-PRG transformation significantly easier.

Proof. Assume that f is $(\sigma, \sigma + \beta/6 - \gamma)$-regular and for now assume that σ is known. Then, by Theorem 5.2, there exists a $2^{\beta' n}$-secure PRG, $G(x, w)$, with linear stretch. We will show that G can be encoded locally with linear complexity. First, observe that $G(x, w)$ can be written as

$$H(x, z, w) = (w, g(x, z, w)), \qquad \text{where } z = f(x)$$

and g is computable by an $O(n)$-size circuit which is skew with respect to the argument w. By Lemma 3.1, we can perfectly encode $H(x, z, w)$ by a 3-local encoding $\hat{H}(x, z, w; r)$ with complexity of $O(n)$. Plugging in $z = f(x)$, we derive (by the so-called substitution Lemma, cf. [AIK14]) a perfect encoding $\hat{G}(x, w; r) = \hat{H}(x, f(x), w; r)$ for $G(x, w)$ with constant locality (at most 3 times larger than the locality of G) and linear complexity. By Lemma 6.3 of [AIK06], the function $\hat{G}$, viewed as a single-input function, is a $2^{\beta' n} - \mathrm{poly}(n)$-strong PRG with additive stretch of $\Omega(n)$. Since the input length n' of $\hat{G}$ is linear in n, both exponential hardness and linear stretch are preserved, and we get an exponentially-strong PRG in $\mathbf{NC^0}$ with linear stretch.

We move on to the case where the parameter σ is unknown. The main observation is that the "allowed window" for s_1, s_2, s_3 is of width $\gamma/8n$. As a result, the parameters can be chosen based on a $\gamma/8n$-additive approximation of σ. In particular, by trying all values of the form $\sigma_i = i\gamma/8n$ for $i \in \{1, \ldots, 8/\gamma\}$, we get a constant number of $\mathbf{NC^0}$ functions $\hat{G}_1, \ldots, \hat{G}_{8/\gamma}$ out of which at least one is an exponentially-strong PRG with linear-stretch. Using a constant number of self-compositions, we can increase the stretch from n'-bits to Cn' for arbitrary large constant C while preserving (larger) constant locality. Now we can combine the candidate PRGs by applying them on independent seeds and XOR-ing their output. The overall input length is $n'' = 8n'/\gamma = O(n')$. By taking $C > 8/\gamma'$ the output is still linearly larger than the input. Moreover, a standard argument shows that exponential security in n' (and therefore also in n'') is preserved. Overall, the resulting mapping still has (large) constant locality. We can re-encode this PRG via the 4-local perfect RE of [AIK06] which has only linear complexity when applied to an $\mathbf{NC^0}$ function. This yields a 4-local linear-stretch PRG with exponential hardness. □

Remark 5.4. *We note that Corollary 5.3 yields a black-box transformation from any OWF that satisfies the stated hardness and regularity conditions to a linear-stretch PRG. The reduction is both local and preserves exponential hardness. Similarly, it can be shown that the RE $\hat{H}(x, w; r)$ of the function $H(x, w)$ from Theorem 3.2 forms a general local hard-core function that, for any exponentially-hard OWFs, outputs linearly many exponentially-secure pseudorandom bits while consuming only a linear number of random bits.*

5.2 Reducing the locality to 3

The locality achieved in Corollary 5.3 is almost optimal. We can get optimal locality of 3 by combining the techniques of [AIK08] together with an assumption of Druk and Ishai [DI14]. PRGs with locality 3, sublinear stretch, and polynomial security were previously constructed based on the intractability of decoding random linear code [AIK09].

In the following, we say that an ensemble $\mathcal{C}$ of $[m, n]$ linear codes is (T, ε)-pseudorandom for noise rate μ if the distribution $(C, Cx + e)$, is (T, ε)-indistinguishable from (C, y), where $C \stackrel{R}{\leftarrow} \mathcal{C}$, $x \stackrel{R}{\leftarrow} \{0,1\}^n, y \stackrel{R}{\leftarrow} \{0,1\}^m$ and e is a noise vector that each of its entries is chosen to be 1 independently with probability μ.

Druk and Ishai [DI14] presented an ensemble of codes with the following properties. For every constant noise rate μ and every linear codeword length $m = O(n)$, the ensemble is conjectured to

be exponentially hard-to-decode just like a random linear code.[9] Moreover, based on this assumption, there exists a related ensemble which is exponentially pseudorandom for similar parameters. Finally, each code C_w in the ensemble can be represented by $O(n)$-bit string w and the mapping $C(x, w) = C_w x$ can be evaluated by an $O(n)$-size circuit which is skew with respect to the index w.

The construction. In [AIK08] it is suggested to turn a code which is pseudorandom for noise rate 2^{-b} into a PRG via the following template.

- Input: a code description w, a random information word $x \in \{0,1\}^n$, a seed for the noise distribution $y = (y_{i,j})_{i\in[m],j\in[b]}$ and a seed z for a seeded randomness extractor $h_z(\cdot)$.
- Output: $(w, z, C_w x + e, h_z(y))$ where $e_i = \prod_{j\in[b]} y_{i,j}$ and multiplication is over $\mathbb{F}_2$.

Note that each entry of e takes the value 1 independently with probability 2^{-b} and therefore the noise distribution is sampled properly. Since the noise sampling procedure is simple but wasteful, a randomness extractor is applied to compensate the entropy loss. It is proved in [AIK08, Lemma 5.6] that except with probability of $\exp(-m2^{-b}/3)$, the min-entropy of e conditioned on $C_w x + e$, is $\ell = (1 - \exp(-b))bm$. Let $b = 2$ (i.e., use a noise rate of $1/4$), and take h to be the pairwise independent hash function from Theorem 3.2 with output length of, say, $\ell/2$. Then, if we take $m = cn$ for sufficiently large constant c, we get an exponentially-strong PRG with linear stretch. (The proof is essentially the same as in [AIK08].)

We construct a 3-local encoding with linear complexity for the PRG $G(w, x, y, z)$ as follows. First, use Lemma 3.1 to encode the function $g_1(x, e) = (w, C_w x+e)$ by a local RE $\hat{g}_1(x, e; r_1)$ and the function $g_2(z, y) = (z, h_z(y))$ by a local RE $\hat{g}_2(z, y; r_2)$. Substituting $e_i = y_{i,1} \cdot y_{i,2}$ and concatenating the encodings, yields a 4-local encoding $G'(w, x, y, z; r_1, r_2) = (\hat{g}_1(x, e; r_1), \hat{g}_2(z, y; r_2))$ with linear complexity and degree $d = 2$. Finally, we re-encode G' via the locality lemma of [AIK06]. When applied to a degree-d local function the lemma yields an RE $\hat{G}'(w, x, y, z, r_1, r_2; r_3)$ with locality $d + 1$ (3 in our case) and complexity which is linear in the output length of G' (which is linear in n). By using standard concatenation, substitution and composition lemmas of REs ([AIK06]), we conclude that the final encoding $\hat{G}(w, x, y, z; r_1, r_2, r_3)$ encodes the PRG G and is therefore a PRG. Since we added only $O(n)$ additional inputs the PRG $\hat{G}(w, x, y, z, r_1, r_2, r_3)$ is exponentially secure.

5.3 Proof of Theorem 5.2

First observe that $s_2 + s_3$ is at least $n(\beta/6 - \gamma/4 + \sigma - \gamma/4 + 1 - \sigma - \beta/6 + \gamma - \gamma/4) > (\gamma/4)n$ and therefore the PRG has $\Omega(n)$ stretch. (Note that the total input length n' of G is linear in n, the input length of f, and so the overall stretch is linear in the input length n').

We prove the pseudoranodmness of G in several steps.

Claim 5.5. *If f is (t, ε) one-way then the function*

$$f_1(x, w_1) = (w_1, f(x), h_{w_1}(x))$$

is $(t_1 = t - \mathrm{poly}(n), \varepsilon_1 \leq \varepsilon + \varepsilon^{1/2} 2^{-L/2})$ one-way where $L = n\sigma - s_2$.

[9] Known sub-exponential algorithms for decoding random linear code (e.g.,[BKW03, Lyu05]) apply only to the case where the codeword is of super-linear length, i.e., the rate is sub-constant.

Proof. Clearly, given $(W_1, f(X), R)$, for $X \xleftarrow{R} \{0,1\}^n, W_1 \xleftarrow{R} \{0,1\}^{t_1}$ and $R \xleftarrow{R} \{0,1\}^{s_1}$, a t_1-time adversary cannot find a preimage of $f(X)$ with probability better than ε. Observe that for every fixing of $Y = y$ the conditional distribution of X is uniform over a set of size at least 2^σ and therefore has min-entropy of σ bits. Theorem 3.4 of [BDK+11] shows that in such a case no t_1-time algorithm can find a preimage of $f(X)$ given $(W_1, f(X), h_{W_1}(X))$ with probability better than ε_1. □

We conclude that f_1 is a $2^{\beta_1 n}$-hard one-way function for some constant $\beta_1 > \beta/2$.

Claim 5.6. *Let $(X, W_1, W_2) \xleftarrow{R} \{0,1\}^{n+t_1+t_2}$ and $Z \xleftarrow{R} \{0,1\}^{s_2}$. Then the distributions*

$$(W_1, W_2, f(X), h_{W_1}(X), C_{W_2}(X)) \quad \textit{and} \quad (W_1, W_2, f(X), h_{W_1}(X), Z),$$

are $2^{-\Omega(n)}$-indistinguishable for $2^{\Omega(n)}$-time algorithms.

Proof. Recall that f_1 is $2^{\beta_1 n}$-hard OWF and s_2/n is upper-bounded by a constant which is strictly smaller than $\beta_1/3$. It is shown in [BIO14, Corollary 23] that under these conditions the claim follows. □

We complete the proof via the following claim.

Claim 5.7. *The distribution*

$$(W_1, W_2, W_3, E_{W_3}(f(X), h_{W_1}(X)), Z) \tag{3}$$

is $2^{\Omega(-n)}$-statistically close to the uniform distribution over $(t_1 + t_2 + t_3 + s_2 + s_3)$-bit strings.

Proof. To see this, first observe that the distribution

$$(W_1, W_2, f(X), h_{W_1}(X), Z)$$

is $2^{\Omega(-n)}$-statistically close to

$$(W_1, W_2, f(X), R, Z) \quad \text{where } R \xleftarrow{R} \{0,1\}^{s_1}.$$

This follows from the leftover hashing lemma [HILL99] and due to the fact that, for every fixed y, the conditional distribution $[X|f(X) = y]$ is uniform over a set of size $\geq 2^{\sigma n}$ and therefore has min-entropy of σn bits. We conclude that the "target distribution" (3) is $2^{\Omega(-n)}$-statistically close to

$$(W_1, W_2, W_3, E_{W_3}(f(X), R), Z).$$

The claim now follows by noting that the last distribution is $2^{\Omega(-n)}$-statistically close to uniform. Indeed, this follows from the leftover hashing lemma together with the observation that $(f(X), R)$ has min-entropy of $s_1 + n(1 - (\sigma + \beta/6 - \gamma)) > s_3 + \Omega(n)$ (due to the upper-bound on preimage size of f). □

This completes the proof of Theorem 5.2. □

References

[ABR16] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. A dichotomy for local small-bias generators. *J. Cryptology*, 29(3):577–596, 2016.

[AC08] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 793–802. IEEE Computer Society, 2008.

[Ach09] Dimitris Achlioptas. Random satisfiability. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 245–270. IOS Press, 2009.

[ACR11] Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Random Struct. Algorithms*, 38(3):251–268, 2011.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in nc^0. *Computational Complexity*, 17(1):38–69, 2008.

[AIK09] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. *J. Cryptology*, 22(4):429–469, 2009.

[AIK14] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. *SIAM J. Comput.*, 43(2):905–929, 2014.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[App13] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM J. Comput.*, 42(5):2008–2037, 2013.

[App16] Benny Applebaum. Cryptographic hardness of random local functions - survey. *Computational Complexity*, 25(3):667–722, 2016.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[BDK+11] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.

[BGLR94] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistic checkable proofs and applications to approximation. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, page 820. ACM, 1994.

[BIO14] Joshua Baron, Yuval Ishai, and Rafail Ostrovsky. On linear-size pseudorandom generators and hardcore functions. *Theor. Comput. Sci.*, 554:50–63, 2014.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BR13] Andrej Bogdanov and Alon Rosen. Input locality and hardness amplification. *J. Cryptology*, 26(1):144–171, 2013.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short pcps with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[CCK+17] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. Unpublished manuscript, 2017.

[CEMT14] James Cook, Omid Etesami, Rachel Miller, and Luca Trevisan. On the one-way function candidate proposed by goldreich. *TOCT*, 6(3):14:1–14:35, 2014.

[CIKP08] Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-sat: An isolation lemma for k-CNFs. *J. Comput. Syst. Sci.*, 74(3):386–393, 2008.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971.

[DI14] Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the gilbert-varshamov bound and their cryptographic applications. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 169–182. ACM, 2014.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[Din16] Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32. ACM, 1989.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[Gol11] Oded Goldreich. Candidate one-way functions based on expander graphs. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, volume 6650 of *Lecture Notes in Computer Science*, pages 76–87. Springer, 2011.

[HHR11] Iftach Haitner, Danny Harnik, and Omer Reingold. On the power of the randomized iterate. *SIAM J. Comput.*, 40(6):1486–1528, 2011.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HMS04] Thomas Holenstein, Ueli M. Maurer, and Johan Sjödin. Complete classification of bilinear hard-core functions. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 73–91. Springer, 2004.

[HRV13] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304. IEEE Computer Society, 2000.

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 433–442. ACM, 2008.

[IP99] Russell Impagliazzo and Ramamohan Paturi. Complexity of k-sat. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240. IEEE Computer Society, 1999.

[Ish13] Yuval Ishai. Randomization techniques for secure computation. In Manoj Prabhakaran and Amit Sahai, editors, *Secure Multi-Party Computation*, volume 10 of *Cryptology and Information Security Series*, pages 222–248. IOS Press, 2013.

[Lev73] Levin. Universal sequential search problems. *PINFTRANS: Problems of Information Transmission (translated from Problemy Peredachi Informatsii (Russian))*, 9, 1973.

[Lin16a] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 28–57. Springer, 2016.

[Lin16b] Huijia Lin. Indistinguishability obfuscation from DDH on 5-linear maps and locality-5 prgs. *IACR Cryptology ePrint Archive*, 2016:1096, 2016.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 11–20. IEEE Computer Society, 2016.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.

[Man16] Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. *CoRR*, abs/1611.05991, 2016. To appear in STOC'17.

[MR16] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. *CoRR*, abs/1607.02986, 2016.

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On epsilon-biased generators in nc^0. *Random Struct. Algorithms*, 29(1):56–81, 2006.

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 1–12. IEEE Computer Society, 2014.

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Information Theory*, 42(6):1723–1731, 1996.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

[VZ12] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 817–836. ACM, 2012.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il