Cryptography Meets Worst-case Complexity: Optimal Security and More From iO and Worst-case Assumptions

Rahul Ilango
Massachusetts Institute of Technology
Cambridge, MA, USA
rilango@mit.edu

Alex Lombardi
Princeton University
Princeton, NJ, USA
alex.lombardi@princeton.edu

***Abstract*—We study several problems in the intersection of cryptography and complexity theory based on the following high-level thesis.**

1) **Obfuscation can serve as a general-purpose *worst-case to average-case reduction*, reducing the existence of various forms of cryptography to corresponding worst-case assumptions.**
2) **We can therefore hope to overcome barriers in cryptography and average-case complexity by (i) making worst-case hardness assumptions beyond $\mathsf{P} \neq \mathsf{NP}$, and (ii) leveraging *worst-case* hardness reductions, either proved by traditional complexity-theoretic methods or facilitated further by cryptography.**

Concretely, our results include:

- **Optimal Hardness. Assuming sub-exponential indistinguishability obfuscation, we give fine-grained worst-case to average case reductions for circuit-SAT. In particular, if finding an NP-witness requires nearly brute-force time in the worst case, then the same is true for some efficiently sampleable distribution. In fact, we show that under these assumptions, there exist families of one-way functions with optimal time-probability security tradeoffs.**
  Under an additional, stronger assumption – the optimal *non-deterministic* hardness of *refuting* circuit-SAT – we construct additional cryptographic primitives such as PRGs and public-key encryption that have such optimal time-advantage security tradeoffs.
- **Direct Product Hardness. Again assuming iO and optimal non-deterministic hardness of SAT refutation, we show that the "(search) k-fold SAT problem" – the computational task of finding satisfying assignments to k circuit-SAT instances simultaneously – has (optimal) hardness roughly $(T/2^n)^k$ for time T algorithms. In fact, we build "optimally secure one-way product functions" (Holmgren-Lombardi, FOCS '18), demonstrating that optimal direct product theorems hold for some choice of one-way function family.**
- **Single-Input Correlation Intractability. Assuming either iO or LWE, we show a worst-case to average-case reduction for strong forms of single-input correlation intractability. That is, powerful forms of correlation-intractable hash functions exist provided that a collection of *worst-case* "correlation-finding" problems are hard.**
- **Non-interactive Proof of Quantumness. Assuming sub-exponential iO and OWFs, we give a non-interactive proof of quantumness based on the *worst-case* hardness of the white-box Simon problem. In particular, this proof of quantumness result does not explicitly assume quantum advantage for an average-case task.**

To help prove our first two results, we show along the way how to improve the Goldwasser-Sipser "set lower bound" protocol to have communication complexity quadratically smaller in the multiplicative approximation error ε.

***Index Terms*—Indistinguishability Obfuscation, Correlation Intractability, One-Way Product Functions, Non-deterministic Hardness, Worst-case to Average-case Reduction**

I. Introduction

Since its inception in the 1980s, the theory of cryptography has delivered remarkable conceptual and mathematical justifications for the security of cryptographic protocols. Its justifications come in at least two different forms.

1) **Constructions based on "standard assumptions."** Cryptographers have carefully accumulated computational hardness assumptions that we have some confidence in – such as the hardness of learning with errors [1], the hardness of factoring [2, 3], and the hardness of discrete logarithms [4] – and built a vast collection of cryptographic primitives and protocols under these assumptions. Over time, these assumptions have come to be labeled as "standard."
2) **Constructions based on generic primitives.** On the other hand, there is a wide web of cryptographic constructions and security reductions *between* cryptographic primitives. As a result, we know that several implications follow generically from the existence of one-way functions [5], oblivious transfer [6], or strong cryptographic primitives such as indistinguishability obfuscation [7, 8].

In recent years, indistinguishability obfuscation ($i\mathcal{O}$) [9, 10] has played an especially important role in "existential" theoretical cryptography. It is a powerful enough primitive that it, in combination with one-way functions, implies a large fraction of "standard cryptography" (see [7] and many subsequent works); on the other hand, as the result of a decade-long effort, it has a candidate construction based on "well-founded"

Rahul Ilango was supported by NSF CCF-2420092 and an NSF graduate research fellowship.

2575-8454/25/$31.00 ©2025 IEEE
DOI 10.1109/FOCS63196.2025.00032

cryptographic assumptions [8]. This suggests the possibility of viewing $i\mathcal{O}$ as a *central hub* for cryptography.

However, over time, we have also identified some *limitations* on our ability to carry out both (1) and (2):

a) Some problems lie beyond standard assumptions: Despite our best efforts, there are plenty of cryptographic primitives that still lie beyond the reach of "standard assumptions." In this paper, we focus on cryptographic primitives that unconditionally exist relative to a random oracle [11] but lack convincing standard-model constructions. Indeed, for some of these tasks – such as building cryptographic primitives with optimal security [12], correlation-intractable hashing [13, 14, 15], and extremely lossy functions [16], it is either known or believed to be impossible to solve them with standard cryptographic assumptions alone.

b) $i\mathcal{O}$ and OWFs do not always suffice: There are black-box impossibility results ruling out constructions of very simple forms of cryptographic hardness – collision-resistant hash functions [17] and hardness in SZK [18] – from $i\mathcal{O}$ and one-way functions. Thus, it is not even the case that "$i\mathcal{O}$ plus OWFs" is complete for "standard cryptography."

In this paper, we ask what can be done in the face of these obstacles.

Question I.1 (Informal)**.** *Can we find new, "reasonable" hardness assumptions that allow us to go beyond the reach of standard assumptions – in particular, to instantiate some of these random-oracle security properties?*

Question I.2 (Informal)**.** *Assuming the existence of $i\mathcal{O}$, what forms of computational hardness are sufficient to imply the existence of standard cryptography?*

A. This work: cryptography questions

We make progress on Question I.1 and Question I.2 in the context of the following problems.

a) Cryptography with Optimal Security: Typically, we say that a cryptographic primitive (such as an encryption scheme) is secure if polynomial-time algorithms cannot break its security property with better than negligible advantage. However, in practice, one needs *concrete security guarantees* about algorithms running in a *specific* running time achieving a *specific quantitative* advantage. For example, the best-known attack on AES-128 runs in time roughly $2^{126.1}$ [19], and its real-world security relies on it having strong quantitative hardness.

A natural idea for theoretically capturing these strong forms of hardness is the notion of *optimal security*. Informally, a cryptographic primitive is optimally secure if the naive brute-force attack (which, of course, depends on the cryptographic primitive) is optimal up to polynomial factors. In fact, we will be interested in *time-probability tradeoffs*, and therefore consider "brute-force attacks" with bounded running time. Naive brute-force attacks for common cryptographic primitives include:

- **One-way functions:** given a OWF f and output $y = f(x)$, sample T i.i.d. inputs x_i and check if $y = f(x_i)$.
- **Pseudorandom generators:** given a PRG G and output y (either pseudorandom or random), sample T i.i.d. inputs x_i and check if $y = G(x_i)$. If yes, output "pseudorandom." Otherwise, output a random bit.
- **Public-key encryption:** there are actually two incomparable brute-force attacks for a PKE scheme. Given a public key pk and ciphertext $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, m; r)$, one can either:
  - Search over the space of secret keys sk, checking whether $(\mathsf{pk}, \mathsf{sk})$ is a valid key pair.
  - Search over the space of encryption randomness r', checking if $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, m; r')$.

  Such attacks will allow for distinguishing between an encryption of a known message m and an encryption of the all zeroes string.
- **Indistinguishability obfuscation:** given an obfuscated program $\widetilde{C}$ and circuits C_0, C_1, search over the space of obfuscation randomness and check if $\widetilde{C} = i\mathcal{O}(C_b; r)$ for some b.

For example, a family[1] of one-way functions is optimally secure [12] if for any time T algorithm for OWF inversion, the success probability of the algorithm is at most $\frac{T}{2^n} \cdot \mathrm{poly}(n)$, where n denotes the OWF input length. One can similarly define optimal hardness families of pseudorandom generators and public-key encryption schemes (where one can ask for either one or both of the brute-force attacks to be optimal).

In this work, we ask:

Question I.3. *Under what assumptions can we build cryptography with optimal security?*

We observe that PRGs appear to be "complete" for a large class of primitives here: given any *sub-exponentially secure* cryptographic primitive (such as public-key encryption or indistinguishability obfuscation), one can combine a family of PRGs mapping $\{0,1\}^n \to \{0,1\}^{\mathrm{poly}(n)}$ with optimal security with an instantiation of the (sub-exponentially secure) primitive using randomness complexity (sufficiently large) $\mathrm{poly}(n)$ to obtain a hybrid scheme with optimal security.

b) One-Way Product Functions: Question I.3 concerns the quantitative security of standard cryptographic primitives. We now turn to a non-standard security property: *product one-wayness* [22, 12]. For an integer k (either constant or depending on the security parameter n), given a one-way function f as well as k outputs $y_i = f(x_i)$, is it hard to simultaneously invert all k instances? It is certainly as hard as the task of inverting a *single* instance. Unfortunately, [22] gives counterexamples showing that for *certain* OWF families, receiving k independent challenges does *not* amplify the quantitative hardness of inversion.

Nevertheless, we say that a OWF family is (T, ε)*-product one-way* if given a function f as well as k outputs $(y_1, \ldots, y_k)$, the probability of simultaneous inversion is at most ε. The time

[1] In the non-uniform security model, it is known that any *fixed* OWF can be broken by a better-than-brute-force algorithms [20, 21]. In this work, we focus on (keyed) families of OWFs, where no such attack is known.

T brute-force attack for this problem achieves $\varepsilon \approx (\frac{T}{2^n})^k$ for OWF inputs of length n. In general, we are interested in the regime where $\varepsilon < \frac{T}{2^n}$, so that hardness cannot follow from (even optimal) hardness of inverting a single OWF challenge. Indeed, an informal intuition dating back to Rudich suggests that it will be difficult for "black-box reductions" to prove that any candidate OWPF has this property under standard assumptions. As a result, we ask:

Question I.4. *Under what assumptions can we build OWPFs?*

Intriguingly, a work of Holmgren and Lombardi [12] shows that if sufficiently quantitatively strong OWPFs exist, they can be used to build families of hash functions satisfying forms of *multi-input correlation intractability* that (1) we do not know how to build from standard assumptions and (2) overcome the Asharov-Segev barrier for constructing CRHFs from $i\mathcal{O}$ and OWFs. We do not discuss this implication further here, but our final question concerns related questions on *single-input correlation intractability*.

c) Correlation-Intractable Hash Functions: Given a binary relation $R(x, y)$, we say that a hash function family $\mathcal{H}$ is R-correlation intractable [23] if it is computationally hard, given a hash key k, to find an input x such that $R(x, H_k(x)) = 1$. Such a security property is plausible whenever (1) the size of the hash key is allowed to grow with the input length (which we will assume throughout this paper), and (2) the relation is *sparse* meaning that for every x, the probability that a random y satisfies $R(x, y) = 1$ is $\mathrm{negl}(n)$.

In recent years, correlation intractability has received a large amount of attention due to its connection to the security of the Fiat-Shamir heuristic [13, 14, 24, 12, 15, 25, 26]. In particular, a wide variety of cryptographic protocols have been constructed by proving the existence of CI hash functions for *specific* kinds of relations based on standard assumptions, and then making use of these hash functions in a protocol.

However, it has remained wide open to construct hash functions that are correlation-intractable for *all* sparse relations.

Question I.5. *Under what assumptions can we construct hash functions that are CI for all sparse relations?*

Two prior works [14, 24] build such hash functions under extremely strong and poorly understood assumptions, such as the existence of encryption schemes that are "optimally unbounded KDM-secure."

Subsequent works avoided the difficulties of this question by considering (sub-classes of) relations that are decidable in polynomial time. However, even this easier question is poorly understood.

Question I.6. *Under what assumptions can we construct hash functions that are CI for all sparse,* efficiently decidable *relations?*

The biggest difficulty in resolving this question concerns what is referred to as "compactness:" Question I.6 is asking for a single family of hash functions – in particular, whose evaluation time is some fixed polynomial in n – that is CI for relations that can be decided in polynomial time, but time longer than it takes to evaluate the hash function. Very little is known in this setting: besides the two mentioned works, [15] gives a construction based on the optimal *circular* security of a fully homomorphic encryption scheme.

B. This work: relationship to complexity theory

Our main thesis is that (worst-case) *complexity theory* offers us powerful tools to help answer these cryptographic questions. Even better, we find that a combination of cryptographic and complexity theoretic reasoning helps us to overcome barriers and improve our understanding of both fields.

a) A new perspective on obfuscation: At the center of our new perspective is the following basic phenomenon.

$i\mathcal{O}$ reduces the existence of cryptographic objects to their worst-case hardness.

Indeed, this has featured prominently in a prior work [27] showing that $i\mathcal{O}$ reduces the existence of one-way functions to the worst-case hardness of solving SAT. It turns out that for other cryptographic objects, while they may not reduce to the hardness of SAT (at least directly), they do reduce to other worst-case problems.

We carefully examine how to use $i\mathcal{O}$ to reduce cryptographic primitives to worst-case problems. This includes both sharpening connections to SAT and also reducing to other worst-case problems. By doing so, we can then make progress on cryptographic questions by applying complexity-theoretic tools. Indeed, our work suggests the following general framework for using $i\mathcal{O}$:

1) Reduce a cryptographic object to a worst-case hardness assumption.
2) Use complexity theory tools and/or assumptions (perhaps beyond P versus NP) to show that worst-case problem is hard.

Intriguingly, in the process of doing (2) for cryptographic purposes, we make progress on independently interesting questions in complexity theory, as described below.

b) How hard is NP*:* The P versus NP question asks whether every NP problem can be solved in polynomial time [28, 29]. Over fifty years since P versus NP was first asked, the best known algorithm for solving a generic NP problem is still the trivial one [30]: $2^w \cdot \mathrm{poly}(t)$ time, where w and t are the bits of non-determinism and time the NP algorithm uses respectively. Moreover, this trivial $2^w \cdot \mathrm{poly}(t)$ is the best known even for algorithms that are equipped with non-uniformity[2] and/or co-non-determinism and even if one just considers average-case problems. By the Cook-Levin theorem, these questions are completely characterized (up to a $\mathrm{poly}(t)$ factor) by the corresponding complexity of Circuit SAT (henceforth, SAT).[3]

[2]For non-uniform circuits, there is also a trivial $O(2^n/n)$ bound, by essentially memorizing the function.

[3]Specifically, given an instance of any such NP problem, one can reduce it in $\mathrm{poly}(t)$ time to an instance of SAT with w-inputs and $\mathrm{poly}(t)$ size.

A fundamental question is whether the bounds in these models are optimal and understanding how they are related. Perhaps the most prominent of these problems is understanding the relationship between worst-case and average-case complexity.

Question I.7 (Connecting the Worst-case and Average-case Hardness of NP)**.** *Are the worst-case and average-case complexities of* NP *tightly related?*

For example, it is consistent with current state-of-knowledge that NP is easy on average, but exponentially hard in the worst case [31]. Indeed, positive progress on this question immediately implies progress on eliminating Heuristica [32], a longstanding open problem in complexity theory. We also mention that for other NP-complete problems such as k-SAT, the best known worst-case and average-case (for natural "hard" distributions) algorithms do *not* have tightly related complexity [33].

Question I.7 is even unclear under powerful assumptions like the existence of indistinguishability obfuscation. While we know that $i\mathcal{O}$ and the worst-case hardness of SAT implies one-way functions (and hence average-case hardness of SAT) [27], the proof of this only gives average-case hardness that is upper bounded by the security S of the obfuscation scheme (and in particular, not the worst-case hardness of SAT). Moreover, all known constructions of obfuscation with security S require assumptions that already imply that NP is hard on average with security S, so this argument seems circular.

Besides Question I.7, we mention two other interesting open problems in worst-case complexity theory.

c) Time-probability tradeoffs for decisional SAT*:* As stated above, we do not know a better algorithm for SAT than the $2^w \cdot \mathrm{poly}(t)$ brute force algorithm. What if we allow the algorithm to be probabilistic? If a probabilistic algorithm runs in time T, what is the largest probability with which it can solve SAT on worst-case instances?

Again, the brute force algorithm is the best we know, which achieves:

- Success probability roughly $\frac{T}{2^w}$ for the *search* problem.
- Success probability roughly $\frac{1}{2} + O(\frac{T}{2^w})$ for the *decision* problem (by guessing uniformly if it fails to find a witness).

For the search problem, it is clear that (T, ε)-hardness follows from the time T/ε-hardness of SAT by naive repetition. However, the tradeoff for the decisional version is harder to reason about: essentially, this is because both known success probability amplification and known search-to-decision reductions for SAT do not have good dependence on the advantage ε.

d) k-fold Search-SAT*:* Even less is known about another type of hardness, the direct product hardness of Search-SAT. Specifically, given k distinct satisfiable circuits, what is best possible success probability ε a time T algorithm can achieve at simultaneously producing satisfying assignments for all of them? This is the complexity of what we call the "k-fold SAT problem."

The trivial brute-force algorithm for this problem achieves $\varepsilon \approx (\frac{T}{2^w})^k$. To the best of our knowledge, there are no results in the literature in the regime where $\varepsilon \ll 2^{-w}$; see discussion by Drucker [34] for more details.

Question I.8. *Does solving multiple instances of* Search-SAT *obey a direct product theorem?*

C. Our Results

We now describe our results on the questions stated above.

a) Optimally Secure Cryptography: We give new constructions of OWF and PRG families with optimal security. Our constructions are based on sub-exponentially secure $i\mathcal{O}$, sub-exponentially secure OWFs, and worst-case assumptions about the quantitative hardness of SAT. Specifically, we introduce the following worst-case hardness assumptions.

Assumption I.9 (Optimal Hardness of NP)**.** There exists a polynomial $p(n)$ such that SAT on circuits of size $p(n)$ and input length n requires circuits of size at least $2^n \cdot \frac{1}{p(n)}$ for all n.

Assumption I.10 (Optimal Non-Deterministic Hardness of coNP)**.** There exists a polynomial $p(n)$ such that UNSAT (the complement of SAT) on circuits of size $p(n)$ and input length n requires *non-deterministic circuits* of size at least $2^n \cdot \frac{1}{p(n)}$ for all n.

We remark that these assumptions make no reference to time-probability tradeoffs; they are about the (non-uniform or non-deterministic) time complexity of solving SAT and UNSAT on all inputs. The assumptions state that there are no worst-case algorithms for these problems beating naive brute-force search by a superpolynomial factor.

Given that there have been no non-trivial algorithms for these problems discovered so far, we find Assumption IV.1 and Assumption IV.2 to be simple and plausible. We briefly mention that the recent non-uniform algorithms of [20, 21] seem to be specific to the setting of "compression problems" and do not have implications for general circuit-SAT.

With these assumptions in hand, we are ready to state our main results.

Theorem I.11 (Informal, see Theorem V.1)**.** *Suppose that sub-exponentially secure $i\mathcal{O}$ and OWFs exist. If, in addition, Circuit-SAT is optimally hard (Assumption IV.1), then there exist optimally secure OWF families.*

Theorem I.12 (Informal, see Theorem V.1)**.** *Suppose that sub-exponentially secure $i\mathcal{O}$ and OWFs exist. If, in addition,* refuting *Circuit-SAT is optimally hard (Assumption IV.2), there exist optimally secure PRG families.*

We make a few remarks on these theorems:

- By [27], the assumption that sub-exponential OWFs exist in Theorems I.11 and I.12 is redundant and is included only for convenience.
- Assumption IV.1 is *necessary* for the existence of optimally secure OWFs. This is a complexity theoretic barrier

to building optimally secure OWFs that likely cannot be overcome with standard assumptions alone. What we show is that, under sub-exponential standard assumptions, there is a worst-case to average-case reduction taking us from the complexity-theoretic barrier all the way to (optimally secure) OWFs.
- Even if Assumption IV.1 is false, our cryptographic construction preserves the fine-grained hardness of circuit SAT up to polynomial factors.
- By the "PRG-completeness" mentioned earlier, Assumption IV.2 (along with sub-exponential standard assumptions) implies the existence of optimally secure cryptographic primitives such as public-key encryption and $i\mathcal{O}$.
- As stated, these results pertain to *classical security*. In Section V, we extend Theorem I.11 to the post-quantum setting, but extending Theorem I.12 remains open.

In order to prove Theorem I.12, we prove the following purely complexity-theoretic result.

Theorem I.13 (Informal, see Theorem IV.9)**.** *If decisional* Unique-SAT *is solvable in time T with $(1/2+\varepsilon)$-success, then* UNSAT *can be solved in non-deterministic time $T/\varepsilon \cdot \mathrm{poly}(n)$.*

Here, Unique-SAT denotes SAT when promised there is at most one satisfying assignment. This is a non-deterministic *refutation* to (unique) decision reduction with optimal parameters. Even without uniqueness, such a reduction was not known (to our knowledge), and it seems unlikely to be able to give a search-to-decision reduction with similar parameters.

We prove Theorem IV.9 by giving an improved Goldwasser-Sipser "set lower bound" protocol with communication complexity quadratically smaller in the multiplicative approximation error ε (see Theorem III.2). Essentially, this gives a "non-deterministic" solution to the coin problem (distinguishing an unbiased coin from a ε-biased coin) with only $1/\varepsilon$ "samples." This result is also crucial for our next main results on OWPFs.

b) Direct Product Hardness: Fascinatingly, it turns out that Assumption IV.2 is also sufficient to construct optimal OWPFs, resolving the hardness amplification problem!

Theorem I.14 (Informal, see Corollary VI.3)**.** *If refuting Circuit-SAT is optimally hard (Assumption IV.2), then for every $k = k(n)$, there exists an optimally secure k-OWPF family.*

As before, we actually give a fine-grained security reduction relating the (T, ε)-hardness of OWPFs to the $\frac{T}{\varepsilon^{1/k}}$-hardness of refuting SAT. Thus, we obtain interesting results even assuming relaxations of Assumption IV.2. For example, taking $k = n$, we can obtain 2^{-n}-secure n-OWPFs assuming only that NP $\not\subseteq$ i.o.-coNP/poly (in addition to $i\mathcal{O}$). See Theorem VI.2 for details.

One major difference between the proofs of Theorem I.14 and of Theorems I.11 and I.12 are that the worst-case complexity-theoretic reductions involved in Theorem I.14 only hold under cryptographic assumptions! That is, we prove:

Theorem I.15 (Informal, see Theorem VI.1)**.** *Assume subexponentially secure $i\mathcal{O}$ and subexponentially secure puncturable PRFs exist. Then, under Assumption IV.2, the k-fold* Search-SAT *problem is optimally hard (for probabilistic algorithms) in the worst case.*

Thus, we have used cryptography ($i\mathcal{O}$) as a catalyst for a *worst-case* complexity-theoretic reduction.

c) Implications for Multi-Input Correlation Intractability and CRHFs: By combining Theorem I.14 with the results of [12], we obtain new constructions of several kinds of multi-input correlation intractable hash functions from sub-exponential $i\mathcal{O}$ and Assumption IV.2. While some of these forms of CI are "beyond standard assumptions," others are more familiar, including collision-resistant hash functions. Although CRHFs are known based on standard assumptions, they are known to be black-box separated from $i\mathcal{O}$ and (even optimally secure) OWFs [17, 12]. We circumvent this barrier by making an assumption about the non-deterministic hardness of coNP.

For this special case of CRHFs, it is reasonable to ask if one really needs the full power of Assumption IV.2. In fact, we show (with a different proof, but somewhat related ideas) that one can drastically weaken Assumption IV.2 and still construct a *distributional* collision-resistant hash function family ([35]).

Theorem I.16 (See Theorem VII.1)**.** *Assume subexponentially secure indistinguishably obfuscation exists, subexponentially secure OWFs exist, and that* NP $\not\subseteq$ *i.o.-*coNP/poly*. Then a distributional collision resistant hash function family exists.*

We leave it as an open problem to improve Theorem I.16 to construct a full-fledged CHRF.

1) More worst-case to average-case reductions: Several of our results above use $i\mathcal{O}$ to help give a worst-case to average-case reduction for computational problems. To conclude the paper, we give a variety of additional applications of this high-level randomization technique.

a) Single-input correlation intractability: We give a worst-case to average-case reduction for the problem of constructing (single-input) CI hash functions. To do this, we define the "worst-case correlation finding problem" $\mathsf{Search}_R[s]$ (Definition VIII.4) and prove the following two theorems:

Theorem I.17 (Informal, see Theorem VIII.9)**.** *Assume that sub-exponential $i\mathcal{O}$ and OWFs exist.*

Then, if $\mathsf{Search}_R[s]$ is worst-case hard for every sparse relation R, then there exists a hash function family that is correlation-intractable for all sparse relations.

Theorem I.18 (Informal, see Theorem VIII.8)**.** *Assume that sub-exponential $i\mathcal{O}$ and OWFs exist,* or *that* LWE *is hard.*

Then, if $\mathsf{Search}_R[s]$ is worst-case hard for every sparse and efficiently decidable *relation R, then there exists a hash function family that is correlation-intractable for all sparse, efficiently decidable relations.*

Thus, we have reduced the existence of CI hash functions to questions about the worst-case complexity of these computational problems. Moreover, we hope that in the future, using some combination of cryptographic and complexity-theoretic

techniques, it will be possible to *prove* these new worst-case hardness conjectures under simpler complexity-theoretic assumptions.

b) (Extremely) Lossy functions: A distribution $\mathcal{D}$ over circuits $C : \{0,1\}^n \to \{0,1\}^m$ is a *lossy function family* [36] if $C \leftarrow \mathcal{D}$ computes an injective function with high probability but $\mathcal{D}$ is computationally indistinguishable from a distribution over circuits with significantly smaller range. Lossy functions can be built under specific algebraic assumptions like DDH or LWE, but are known to be impossible to build from $i\mathcal{O}$ and OWFs alone. *Extremely lossy functions* [16] are a strengthening of the primitive that demands the existence of (T, ε)-lossy modes that are (T, ε)-indistinguishable from injective mode but have ranges of size $\mathrm{poly}(T, 1/\varepsilon)$. ELFs are currently only known to exist based on the exponential hardness of DDH.

In Section IX, we give a worst-case to average-case reduction for lossy functions (see Theorems IX.6 and IX.14), showing that assuming sub-exponential $i\mathcal{O}$ and OWFs, lossy functions exist under the additional assumption that there is no efficient algorithm distinguishing *worst-case* injective and lossy circuits. This reduction holds both in the setting of mild lossiness and extreme lossiness. In the "mild" case, the resulting worst-case problem seems fairly similar to variants of the *entropy approximation* problem that are known to be hard for NISZK [37, 38, 39], but we leave establishing a formal connection as an open problem.

c) Non-interactive proofs of quantumness: A non-interactive proof of quantumness is a challenge-response protocol in which an efficient quantum prover can make an efficient classical verifier accept with high probability, while any polynomial-time classical prover would be rejected with high probability. Such protocols are known to exist unconditionally in *oracle models* [40, 41, 42], but in the standard model, such protocols are only known under computational assumptions; namely, assumptions that some quantumly easy NP problem is classically hard, such as the factoring or discrete logarithm problems.

In more detail, subject to being in the standard model and having efficient verification, two natural protocol types come to mind:

1) Protocols whose security is proven under a tautological assumption, such as those based on factoring and discrete log.
2) Protocols obtained by heuristically obfuscating a computational problem that is unconditionally hard in an oracle model. This type of protocol does not come with a formal security proof, except under idealized assumptions such as black-box obfuscation.

In Section X (Theorem X.2), we construct a non-interactive proof of quantumness of type (2) and *prove its security* assuming sub-exponential $i\mathcal{O}$, sub-exponential OWFs, as well as a new *worst-case* hardness assumption about Simon's problem [40]. Briefly, Simon's oracle problem is to find a hidden period s with black-box access to a function $f : \{0,1\}^n \to \{0,1\}^m$ that is injective modulo s. Our hardness assumption is that the *white-box* version of this problem – where the algorithm is given a circuit C computing f – is hard in the worst case; that is, there is no efficient algorithm A that outputs s on *every* C of bounded polynomial size. Using $i\mathcal{O}$, we give a worst-case to average-case reduction for this white-box problem, which immediately yields a proof of quantumness.

We remark that although Theorem X.2 assumes $i\mathcal{O}$ and OWFs, these are not "quantum advantage" assumptions in that they do not assume any separation between efficient quantum and classical algorithms; the worst-case white-box Simon assumption is the only quantum advantage assumption that is needed. In contrast, protocols such as those based on factoring explicitly make an *average-case* quantum advantage assumption. Theorem X.2 can also be composed with worst-case complexity theoretic reductions, yielding proofs of quantumness assuming $i\mathcal{O}$ and the worst-case hardness of generalizations of Simon's problem (see Theorem X.6). We hope that this approach will lead to proofs of quantumness based on even weaker worst-case quantum advantage assumptions, ideally even a generic assumption such as $\mathsf{BQP} \not\subset \mathsf{BPP}$.

D. Prior work using complexity theory in cryptography

Our work is not the first to use strong worst-case complexity assumptions in cryptography. One general technique, pioneered by Barak, Ong, and Vadhan [43], is to use complexity theory assumptions (e.g., E lacks sub-exponential size non-deterministic circuits) for *derandomization* purposes. For instance, Barak, Ong, and Vadhan use this framework to remove interaction from commitment schemes and witness indistinguishable proof systems.

Moreover, indistinguishability obfuscation has been used to build average-case hardness in at least two notable settings. Komargodski et al. [27] show that assuming $i\mathcal{O}$, if $\mathsf{NP} \not\subset i.o.\mathsf{BPP}$, then one-way functions exist; we discuss the relationship between this work and [27] below. In addition, [44] construct time-lock puzzles assuming $i\mathcal{O}$ and that there exist languages that do not parallelize in the worst case.

Another relevant prior work is by Ilango, Li, and Williams [45]. They prove hardness for a problem (circuit range avoidance [46]) assuming indistinguishability obfuscation and $\mathsf{NP} \neq \mathsf{coNP}$. While their assumptions (indistinguishability obfuscation and non-deterministic hardness) are similar to ours, the techniques used by Ilango, Li, and Williams appear specific to the range avoidance problem and have a different flavor than ours. For example, while worst-case to average-case reductions and puncturable PRFs are key components in our proofs, their proofs involve neither.

E. Technical Overview

To give a taste of our techniques, we give an overview of Theorems I.11, I.12 and I.14. The proofs of Theorems I.16 to I.18 as well as Theorems IX.6 and X.2 are deferred to Sections VII to X.

1) A Fine-Grained Worst-Case to Average-Case Reduction for Search-SAT: We start with a "warm-up" fine-grained worst-case to average-case reduction for Search-SAT using

$i\mathcal{O}$, and explain in Section I-E2 how to extend this to an optimally secure OWF.

Our goal is to construct a distribution $\mathcal{D}$ such that solving Search-SAT on $\mathcal{D}$ is roughly as hard as solving SAT in the worst-case.

a) Why isn't this already solved: At first glance, one might think that this is already solved by the prior work of [27]. In particular, [27] show that if SAT is worst-case hard and $i\mathcal{O}$ exists, then a one-way function f exists (and, hence, Search-SAT is hard on average). Unfortunately, this approach fundamentally does not preserve fine-grained hardness.

In more detail, the corresponding hard Search-SAT distribution $\mathcal{D} = \{D_\lambda\}$ guaranteed by [27] is the following:

1) Sample $B \leftarrow i\mathcal{O}(0_\lambda)$, where 0_λ is the all zeroes circuit padded to length λ.
2) Output the Search-SAT instance given by the circuit $C(r) = \mathbb{1}[i\mathcal{O}(0_\lambda; r) = B]$.

Suppose we wanted to prove that this Search-SAT instance distribution is optimally hard. The key questions are

- What is the length of the NP witness? It is the *randomness complexity* of the $i\mathcal{O}$ scheme.
- How quantitatively hard is this Search-SAT distribution? Roughly speaking, it is the quantitative security of the $i\mathcal{O}$ scheme.

Thus, in order to obtain an optimally hard Search-SAT distribution, one would need to assume optimally hard $i\mathcal{O}$, a much stronger assumption than the conclusion we wanted.

In summary, it is not clear how to use [27] to get the desired result. Indeed, our fine-grained reduction from worst-case Search-SAT to average-case Search-SAT takes a different approach based on the following two ideas.

b) Idea 1: Valiant-Vazirani: First, we use the seminal work of Valiant and Vazirani [47]. Valiant and Vazirani show that solving Search-SAT reduces to the case where one is guaranteed that a circuit has *exactly* one satisfying assignment, which we will refer to as Search-Unique-SAT. Importantly, their reduction is fine-grained and preserves optimal hardness.

c) Idea 2: Uniquely Satisfiable Circuits are Rerandomizable: Our next idea is that uniquely satisfiable circuits are *rerandomizable* in a certain sense. Given an arbitrary uniquely satisfiable circuit C, one may not know where the unique satisfying assignment $w^\star$ is. However, even without knowing $w^\star$, one can move the uniquely satisfying assignment to a random place. To do this, choose a random shift $z \leftarrow \{0,1\}^n$ and consider the new circuit $C_z(x) = C(x \oplus z)$. Now the unique satisfying assignment for C_z is at $z \oplus w^\star$.

Restating the above in different notation, we have that the following two distributions are statistically identical:

- *Truth table of Random Point Function:*
  - Sample $x^\star \leftarrow \{0,1\}^n$
  - Output the truth table of $x \mapsto \mathbb{1}[x = x^\star]$
- *Truth Table of Rerandomized Circuit:*
  - Sample $z \leftarrow \{0,1\}^n$
  - Output the truth table of C_z

Thus, the security guarantee of $i\mathcal{O}$ says that the following two distributions are computationally indistinguishable:

- *Obfuscated Rerandomized Circuit:*
  - Sample $z \leftarrow \{0,1\}^n$
  - Output $i\mathcal{O}(C_z)$
- *Obfuscated Random Point Function:*
  - Sample $x^\star \leftarrow \{0,1\}^n$
  - Output $i\mathcal{O}(x \mapsto \mathbb{1}[x = x^\star])$

Hence, assuming subexponentially secure $i\mathcal{O}$ and setting the security parameter appropriately, we get that any T-time ε-successful algorithm for Search-SAT on an obfuscated random point function is also a (roughly) T-time ε-successful algorithm for Search-SAT on the distribution $i\mathcal{O}(C_z)$. Because one can extract a satisfying assignment for C from a satisfying assignment for C_z (just xor with z), this further means there is a roughly T-time ε-successful algorithm for Search-Unique-SAT. By success amplification, this yields a $\frac{T}{\varepsilon}$-time algorithm for Search-Unique-SAT.

Putting all this together, we get a fine-grained reduction from worst-case solving Search-Unique-SAT in time roughly $\frac{T}{\varepsilon}$ to solving Search-SAT on average on obfuscated random point functions in T-time with ε-success.

2) Optimally Secure Cryptography.: Our proofs of Theorems I.11, I.12 and I.14 all proceed in two steps:

1) A worst-case to average-case reduction showing that the desired cryptographic object exists assuming the worst-case hardness of an analogous complexity theoretic problem.
2) A worst-case reduction from the hardness of SAT (or the non-deterministic hardness of UNSAT) to the worst-case hardness of the problem from Step (1).

Step (2) is different for each of our three theorems. However, the cryptographic construction in Step (1) is the same for all three cases: just obfuscate a puncturable pseudorandom function (with the appropriate output length).

a) Optimal One-Way Functions: In more detail, the construction is $f \leftarrow i\mathcal{O}(F_{\mathsf{sk}})$ where $F_{\mathsf{sk}} : \{0,1\}^n \to \{0,1\}^{10n}$ is a $2^{O(n)}$-secure puncturable PRF. Observe that the parameters have been chosen so that f is injective with very high probability. We now sketch the proof of one-wayness.

Suppose a T-time inverter I succeeds at inverting f on a random point $y^\star = f(x^\star)$ with ε probability. Then, by $i\mathcal{O}$ security and puncturable PRF security, I must also succeed at inverting $f_{x^\star,r}$ on $y^\star = r$ with probability about ε, where we sample $r \leftarrow \{0,1\}^{10n}$ and run the inverter on

$$f_{x^\star,r} \leftarrow i\mathcal{O}\left(x \mapsto \begin{cases} r, & \text{if } x = x^\star \\ F_{\mathsf{sk}}(x), & \text{otherwise} \end{cases}\right).$$

Now, we can invoke the idea of our "warm-up" reduction: we use such an I to solve Search-Unique-SAT on a *worst-case* Search-Unique-SAT instance C by sampling a random mask z and running the inverter on the obfuscated program

$$i\mathcal{O}\left(x \mapsto \begin{cases} r, & \text{if } C(x \oplus z) = 1 \\ F_{\mathsf{sk}}(x), & \text{otherwise} \end{cases}\right).$$

By the argument above, this implies we can solve Search-SAT in the worst-case with a circuit of size roughly $\frac{T}{\varepsilon}$.

b) Optimal PRGs: The argument for PRGs is similar, with one major difference. While one-way function security is defined in terms of the hardness of a search problem, PRG security is defined in terms of a decision problem. As a result, a T-time ε-distinguisher for the PRG now implies a T-time algorithm for *(decisional)* Unique-SAT that is correct with $\frac{1}{2}+\varepsilon$ probability on all inputs.

Now we face two related problems.

- Distinguishing an unbiased coin from a ε-biased coin requires $\Theta(\frac{1}{\varepsilon^2})$ many samples. As a result, success amplification only gives a roughly $\frac{T}{\varepsilon^2}$-time algorithm for SAT. This extra squared dependence on ε would mean our PRG has not been proven optimally secure, even if SAT has optimal hardness.
- Alternatively, it would be nice if one could even just reduce from the (T,ε)-hardness of decisional SAT. However, the Valiant-Vazirani reduction also does not work in this parameter regime.

Thus, it is unclear how to base PRG security on anything simpler than the (T,ε)-hardness of decisional Unique-SAT.

To get around these issues, we turn to a different assumption: the hardness of refuting SAT. We show that the aforementioned T-time $(\frac{1}{2}+\varepsilon)$-successful algorithm A for Unique-SAT can be converted into a $\frac{T}{\varepsilon}$-size non-deterministic circuit for UNSAT. The main idea is that, by using non-determinism, one can certify A rejects too often for a circuit to have been satisfiable and hence must be unsatisfiable. This uses the fact that one can non-deterministically certify a lower bound on the size of a set via Goldwasser-Sipser's set lower bound protocol [48].

Unfortunately, this still does not work, as Goldwasser-Sipser also incurs an ε^{-2} blowup – indeed, for our setting of parameters, the Goldwasser-Sipser protocol simply estimates the bias of A using the standard solution to the "coin problem" (estimate A's success probability with $O(1/\varepsilon^2)$ samples). Instead, we actually give an improvement to the Goldwasser-Sipser protocol that only has an ε^{-1} dependence. Our key idea (which generalizes to the full setting of Goldwasser-Sipser) is a method for solving the coin problem in $\frac{1}{\varepsilon}$ "non-deterministic" samples. We give more intuition for this protocol in Section I-E4 and give the formal protocol in Section III.

3) One-Way Product Functions and k-fold SAT*:* By essentially the same argument as for (non-product) one-way functions, one can convert a T-time ε-successful inverter for a one-way product function into a roughly T-time ε-successful algorithm for an analogous worst-case problem we call Search-Unique-k-fold-SAT. In Search-Unique-k-fold-SAT, one is given k satisfiable instances of Search-Unique-SAT and the goal is to output satisfying assignments to all of them. To complete our proof of Theorem I.14, it now suffices to prove Theorem I.15: that one can convert such an algorithm for Search-Unique-k-fold-SAT into a roughly $\frac{T}{\varepsilon^{1/k}}$-time non-deterministic algorithm for UNSAT.

a) The hardness of Search-Unique-k-fold-SAT*:* Our goal is to non-deterministically reduce UNSAT to Search-Unique-k-fold-SAT. Our strategy will be as follows. Recall, Goldwasser-Sipser (and our improvement of it) lets us non-deterministically certify a lower bound on the probability that an event occurs. Our strategy is to use any algorithm for Search-Unique-k-fold-SAT in order to produce an event that occurs with (multiplicatively) $(1+\varepsilon^{-1/k})$ higher probability when φ is unsatisfiable than when it is satisfiable. Then, by certifying this event occurs with higher probability, we prove that circuit is unsatisfiable.

In more detail, suppose A is a probabilistic algorithm that given k satisfiable circuits $\Phi=(\varphi_1,\ldots,\varphi_k)$ outputs a satisfying assignment to all of them with probability ε.

We begin by making a *strong* simplifying assumption that A is *oblivious* in both of the following ways:

- *Success Probability Oblivious*: This roughly says that success probability of the algorithm only depends on the number of satisfying assignment its input has. Formally, there is a function $p:[2^n]\to[0,1]$ such that if $\varphi_1,\ldots,\varphi_k$ each have the same number i of satisfying assignments, then we have that the probability that $A(\varphi_1,\ldots,\varphi_k)$ succeeds (i.e., outputs satisfying assignments to all of $\varphi_1,\ldots,\varphi_k$) exactly equals $p(i)$.
- *Witness Oblivious*: This roughly says that the witnesses A produces are uniformly random. When $A(\Phi)$ succeeds and outputs a tuple of satisfying assignments to Φ, it outputs a uniformly random such tuple among the satisfying assignments of $\varphi_1,\ldots,\varphi_k$.

Although this assumption on A appears to be strong, we give a *cryptographic* reduction showing that any A for Search-Unique-k-fold-SAT can be converted into an "oblivious" A, by replacing each input circuit φ_i with a carefully randomized circuit $i\mathcal{O}(\varphi_i \circ PRF)$). See Section VI-A for details.

We will now use this oblivious A to create an event that occurs with higher probability when φ is unsatisfiable. Let $i\in[2^n]$ be a parameter we choose later. Let $\varphi'(x)=\varphi(x)\vee \mathbb{1}[\,x\in[i]\,]$, where we interpret $j\in[2^n]$ as an element of $\{0,1\}^n$ in the natural way. We can then look at the probability

$$\alpha=\Pr_A[A(\varphi',\ldots,\varphi')\in[i]^k].$$

Now if φ were unsatisfiable, then the satisfying assignments to φ' are exactly $[i]$. Hence, because A is oblivious, we get that $\alpha=p(i)$. On the other hand, if φ is satisfiable at exactly one point outside of $[i]$ (we can reduce to this case using Valiant-Vazirani and brute forcing over the elements of $[i]$), we would have that

$$\alpha=\frac{i^k}{(i+1)^k}\cdot p(i+1).$$

because A is oblivious.

A careful argument reveals we can choose an $i\le O(\varepsilon^{-1/k})$ such that the ratio between α in the unsatisfisable case and the satisfiable case is roughly $(1+\varepsilon^{1/k})$. Then by using our improved Goldwasser-Sipser lower bound protocol, we can

certify that α is large for unsatisfiable φ with a proof of length length roughly $\varepsilon^{-1/k}|A|$, as desired.

4) Improved Goldwasser-Sipser: Finally, we describe our improvement to Goldwasser-Sipser. It turns out our algorithm can be understood as a non-deterministic algorithm for solving the coin problem. Recall that, in the coin problem, the goal is to distinguish between a fair coin and a coin that has $1/2+\varepsilon$ probability of being heads.

We can formalize this as follows. Let $p \in \{1/2, 1/2+\varepsilon\}$. Let $f : \mathbb{N} \to \{0,1\}$ be the random variable where for each $n \in \mathbb{N}$ we independently set $f(n) = 1$ with probability p and $f(n) = 0$ with probability $1-p$.

Our goal is, given oracle access to f, to distinguish whether $p = 1/2$ or $p = 1/2+\varepsilon$. A standard result in information theory is that one requires $\Omega(\frac{1}{\varepsilon^2})$ queries in order to distinguish the two cases with constant probability. In contrast, we show that if one is allowed *non-deterministic queries*, then $O(\frac{1}{\varepsilon})$ queries suffice, a quadratic improvement.

The idea is to use the non-determinism to look for an *extremely rare* event. Let q be a parameter set later (it will be the number of queries we use). Let E_i be the event that there is a run of q ones starting at i, i.e., that $f(i) = \cdots = f(i+q-1) = 1$. It is easy to see that

$$\Pr[E_i] = p^q.$$

Now let E be the event that E_i occurs for some $i \in [2^w]$, where w is a parameter we set later (it will correspond to the bits of non-determinism we use).

We will use the approximation that $\Pr[E] \approx 2^w p^q$ (which turns out to give the right answer). If $p = 1/2$, we have that

$$\Pr[E] \approx 2^w 2^{-q} \leq .01$$

by setting $w = \Theta(q)$ sufficiently small. On the other hand, when $p = 1/2+\varepsilon$, we have

$$\Pr[E] \approx 2^w(1/2+\varepsilon)^q = 2^w 2^{-q}(1+2\varepsilon)^q \approx 2^w 2^{-q} e^{2\varepsilon q} \approx 1$$

by setting $q = O(1/\varepsilon)$ sufficiently large.

In other words, E occurs with small probability when $p = 1/2$ and with high probability when $p = 1/2+\varepsilon$. Thus, we have the following non-deterministic algorithm for the coin problem using $O(1/\varepsilon)$ bits of non-determinism and $O(1/\varepsilon)$ queries. Non-deterministically guess an $i \in [2^w]$ and use q queries to check that E_i holds (i.e., that $1 = f(i) = \ldots f(i+q-1)$). If so then say "$p = 1/2+\varepsilon$."

In Section III, we show how to use (a derandomized version of) this argument to show that nmCAPP (the non-deterministic multiplicative circuit acceptance probability problem) has an Arthur-Merlin protocol with an $1/\varepsilon$ dependence.

II. Preliminaries

A. Circuits and SAT

In this work, we consider the following types of circuits:

- *Non-deterministic circuit.* A non-deterministic circuit is a circuit $C : \{0,1\}^n \times \{0,1\}^w \to \{0,1\}$ where we treat the second input as "non-determinism." In particular, $C(x) = \mathbb{1}[C(x,z) = 1 \text{ for some } z]$.
- *Probabilistic circuit.* A probabilistic circuit is a circuit $C : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}$ where we treat the second input as "randomness." In particular, we say that C computes a function $f : \{0,1\}^n \to \{0,1\}$ if

$$\Pr_r[C(x,r) = f(x)] \geq 2/3$$

for all $x \in \{0,1\}^n$.
- *Aruthur-Merlin circuit.* An Arthur-Merline circuit is a circuit $C : \{0,1\}^n \times \{0,1\}^r \times \{0,1\}^w$ where we treat the second input as "randomness" and the third input as "non-determinism." In particular, we say that C computes a function $f : \{0,1\}^n \to \{0,1\}$ if we have that

$$\Pr_r[C(x,r,w) = 0 \text{ for all } w] \geq 2/3$$

whenever $f(x) = 0$ and if we have that

$$\Pr_r[C(x,r,w) = 1 \text{ for some } w] \geq 2/3$$

whenever $f(x) = 1$.

Adleman [49] shows that one can always remove randomness from a probabilistic circuit or Arthur-Merlin circuit with a small blow-up.

Lemma II.1 ("Adleman's trick" [49])**.** *Assume a probabilistic circuit (respsectively, Arthur-Merlin circuit) C computes a function $f : \{0,1\}^n \to \{0,1\}$. Then there is a deterministic (respectively, non-deterministic) circuit computing f of size at most $O(n \cdot |C|)$.*

Proof. First consider a probabilistic circuit $C : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}$. Let $k \in \mathbb{N}$ be an odd number that we set later. Sample $y_1, \ldots, y_k \leftarrow \{0,1\}^r$. Consider the (deterministic) circuit C' that on input x outputs the majority value of $C(x, y_i)$ over all $i \in [k]$. Because Majority has linear-sized circuits [50], we have that $|C'| = k \cdot |C| + O(k)$.

We claim that with positive probability that C' computes f. We do this by a union bound argument. For any fixed x, a Chernoff bound says that

$$\Pr_{y_1,\ldots,y_k}[C'(x) \neq f(x)] \leq e^{-\Omega(k)} < 2^{-2n}$$

by setting $k = O(n)$ sufficiently large. Union bounding over all 2^n many x, we get that C' computes f with positive probability. This completes the proof for probabilistic circuits. The argument for an Arthur-Merlin circuit is similar. □

a) Variants of SAT.: We use the following notation for variants of circuit satisfiability:

- SAT refers to (decisional) circuit satisfiability.
- UNSAT refers to the complement of SAT.
- Unique-SAT refers to SAT restricted to instances where one is promised the circuit has at most one satisfying assignment.
- Unique-UNSAT refers to UNSAT restricted to the promise of at most one satisfying assignment.

- Search-SAT refers to the search version of SAT (where one needs to output a satisfying assignment if the circuit has one)
- Search-Unique-SAT refers to the restriction of Search-SAT to instances with exactly one satisfying assignment.
- Search-k-fold-SAT refers to the following search problem: given a tuple $\Phi = (\varphi_1, \ldots, \varphi_k)$ of circuits, output a tuple $(w_1, \ldots w_k)$ of satisfying assignments ($\varphi_i(w_i) = 1$ for all i).
- Search-Unique-k-fold-SAT refers to the restriction of Search-k-fold-SAT to instances Φ where each circuit φ_i has exactly one satisfying assignment.

For any of these problems Π, we let $\Pi[n, s]$ correspond to the problem on n-input circuits of size s. We say $\Pi[n, s]$ is solvable in t-time with ε-success if there is a t-size probabilistic circuit that solves the problem with probability at least ε.

For problems where the input is a circuit, there are two natural parameters with respect to which one would like to measure hardness: the description size s of the circuit and the number of inputs n to the circuit. We measure hardness in the following way.

Definition II.2 (Hardness of Circuit Problems)**.** Let $f : \{0,1\}^\star \to \{0,1\}^\star$ be a function from s-length n-input circuits φ to strings.

Let $t : \mathbb{N} \to \mathbb{N}$. Let $\varepsilon : \mathbb{N} \to \mathbb{R}$. We say that f is (t, ε)-hard if there exists $s(n) = \mathrm{poly}(n)$ such that is no family of $t(n) \cdot \mathrm{poly}(s)$ size probabilistic circuits that solves f on n-input $s(n)$-description-length circuits with probability at least $\varepsilon(n)$ for infinitely many n.

B. Pairwise independence

For our "faster Goldwasser-Sipser" protocol (Theorem III.2), we make use of linear-time computable pairwise independent hash functions.

Lemma II.3 (Efficient Pairwise Independent Hashing)**.** *There is a pairwise independent hash family $\{H_k : \{0,1\}^n \to \{0,1\}^m\}_{k \in \{0,1\}^{n+m}}$ that is computable in time $(n + m) \cdot \mathrm{poly}(\log(nm))$.*[4]

C. Cryptographic Primitives

In this paper, we work with *keyed families* of one-way functions (OWFs) and pseudorandom generators (PRGs): the OWF/PRG has a public evaluation key sampled from some (efficiently sampleable) distribution. We give the formal definitions below.

Definition II.4 (Keyed family of functions)**.** A keyed family of functions $\mathcal{F}$ with input length n and output length $m = m(n)$ is described by the following algorithms:

- $\mathsf{Gen}(1^n) \to \mathsf{pp}$ is a randomized algorithm taking as input a security parameter λ and outputting public parameters pp.
- $\mathsf{Eval}(\mathsf{pp}, x) \to y$ is a deterministic algorithm taking as input public parameters pp and and input $x \in \{0,1\}^n$, returning an output $y \in \{0,1\}^m$.

We say that $\mathcal{F}$ is *efficiently computable* if $\mathsf{Gen}, \mathsf{Eval}$ are polynomial-time algorithms. We use the notation $f_{\mathsf{pp}}(x)$ to denote the output of $\mathsf{Eval}(\mathsf{pp}, x)$.

Definition II.5 ((T, ε)-one way function family)**.** We say that an efficiently computable function family $\mathcal{F}$ is a $(T(n), \varepsilon(n))$-one way function if for all $T(n)$-time adversaries $\mathcal{A}$, we have that

$$\Pr_{\substack{\mathsf{pp} \leftarrow \mathsf{Gen}(1^n) \\ x \leftarrow \{0,1\}^n}} \left[f(\mathcal{A}(\mathsf{pp}, f_{\mathsf{pp}}(x))) = f_{\mathsf{pp}}(x) \right] = O(\varepsilon(n)).$$

Definition II.6 ((T, ε)-pseudorandom generator family)**.** We say that an efficiently computable function family $\mathcal{F}$ is a $(T(n), \varepsilon(n))$-pseudorandom generator if the ensembles of distributions

$$(\mathsf{pp} \leftarrow \mathsf{Gen}(1^n), y \leftarrow f_{\mathsf{pp}}(U_n)) \approx_{T,\varepsilon} (\mathsf{pp} \leftarrow \mathsf{Gen}(1^n), y \leftarrow U_m)$$

are (T, ε)-computationally indistinguishable.

1) One-Way Product Functions: Following [12], we define families of *one-way product functions*, which are OWFs that are (T, ε)-hard to *simultaneously* invert on batches of k outputs.

Definition II.7 (OWPF Family)**.** Let $T(n), \varepsilon(n), k(n)$ be functions of n. An efficiently computable function family $\mathcal{F}$ is a (T, ε) k-OWPF family if for all $T(n)$-time adversaries $\mathcal{A}$,

$$\Pr_{\substack{\mathsf{pp} \leftarrow \mathsf{Gen}(1^n) \\ x_1, \ldots, x_k \leftarrow \{0,1\}^n \\ (x'_1, \ldots, x'_k) \leftarrow A(\mathsf{pp}, f_{\mathsf{pp}}(x_1), \ldots, f_{\mathsf{pp}}(x_k))}} \left[f_{\mathsf{pp}}(x'_i) = f_{\mathsf{pp}}(x_i) \text{ for all } i \right]$$

is at most $O(\varepsilon(n))$.

2) Indistinguishability Obfuscation: The following preliminaries on $i\mathcal{O}$ and puncturable PRFs are based on [51].

Definition II.8 (Indistinguishability Obfuscation [9])**.** An indistinguishability obfuscation scheme consists of a probabilistic polynomial-time algorithm $i\mathcal{O}(1^\lambda, C)$ that takes as input a security parameter λ as well as a circuit C. It outputs another circuit $\widetilde{C}$ with the same input and output length. An $i\mathcal{O}$ scheme must satisfy two properties:

- Correctness: for all circuits C and inputs x, we have that $C(x) = \widetilde{C}(x)$ with probability 1.
- (T, δ)-security: for all pairs of functionally equivalent circuits $C_0 \equiv C_1$ of size $|C_0| = |C_1| = \mathrm{poly}(\lambda)$, and all adversaries $\mathcal{A}$ of size $T(\lambda, |C|)$, it holds that

  $$\Pr[\mathcal{A}(1^\lambda, i\mathcal{O}(1^\lambda, C_0)) = 1] - \Pr[\mathcal{A}(1^\lambda, i\mathcal{O}(1^\lambda, C_1)) = 1]$$

  is at most $O(\delta(\lambda))$.

[4] For concreteness, our model of computation for this lemma statement is a multi-tape Turing machine. For our applications, it suffices for the lemma to be true in the circuit model.

a) Complexity leveraging.: Let n denote the input length of a given circuit C. As long as an $i\mathcal{O}$ scheme is *sub-exponentially secure* (meaning we can take $T(\lambda, C) = \mathrm{poly}(|C|) \cdot 2^{\lambda^\varepsilon}$ and $\delta(\lambda) = 2^{-\lambda^\varepsilon}$ for some $\varepsilon > 0$, we may set $\lambda = n^{O(1/\varepsilon)}$ sufficiently large so that the scheme is secure against algorithms running in time $2^{O(n)} \cdot \mathrm{poly}(|C|)$ with advantage $2^{-O(n)} \cdot \mathrm{negl}(n)$.

3) Puncturable Pseudorandom Functions:

Definition II.9 (Puncturable PRF [52, 53, 54, 7])**.** A puncturable PRF family is a family of functions

$$\mathcal{F} = \left\{ F_{\lambda,\mathsf{sk}} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)} \right\}_{\lambda \in \mathbb{N}, \mathsf{sk} \in \{0,1\}^{\ell(\lambda)}}$$

with associated (deterministic) polynomial-time algorithms $(\mathcal{F}.\mathsf{Eval}, \mathcal{F}.\mathsf{Puncture}, \mathcal{F}.\mathsf{PuncEval})$ satisfying

- For all $x \in \{0,1\}^{n(\lambda)}$ and all $\mathsf{sk} \in \{0,1\}^{\ell(\lambda)}$, $\mathcal{F}.\mathsf{Eval}(\mathsf{sk}, x) = F_{\lambda,\mathsf{sk}}(x)$.
- For all distinct $x, x' \in \{0,1\}^{n(\lambda)}$ and all $s \in \{0,1\}^{\ell(\lambda)}$,

$$\mathcal{F}.\mathsf{PuncEval}(\mathcal{F}.\mathsf{Puncture}(\mathsf{sk}, x), x') = \mathcal{F}.\mathsf{Eval}(\mathsf{sk}, x')$$

For ease of notation, we write $F_{\mathsf{sk}}(x)$ and $\mathcal{F}.\mathsf{Eval}(\mathsf{sk}, x)$ interchangeably, and we write $\mathsf{sk}\langle x \rangle$ to denote $\mathcal{F}.\mathsf{Puncture}(\mathsf{sk}, x)$.

$\mathcal{F}$ is said to be (s, δ)-secure if for every $\{x^{(\lambda)} \in \{0,1\}^{n(\lambda)}\}_{\lambda \in \mathbb{N}}$, the following two distribution ensembles (indexed by λ) are $\delta(\lambda)$-indistinguishable to circuits of size $s(\lambda)$:

$$(\mathsf{sk}\langle x^{(\lambda)} \rangle, F_{\mathsf{sk}}(x^{(\lambda)})) \text{ where } \mathsf{sk} \leftarrow \{0,1\}^{\ell(\lambda)}$$

and

$$(\mathsf{sk}\langle x^{(\lambda)} \rangle, U) \text{ where } \mathsf{sk} \leftarrow \{0,1\}^{\ell(\lambda)}, U \leftarrow \{0,1\}^{m(\lambda)}.$$

Theorem II.10 ([55, 54, 52, 53, 7])**.** *If $\{$polynomially secure, subexponentially secure$\}$ one-way functions exist, then for all functions $m : \mathbb{N} \to \mathbb{N}$ (with $1^{m(n)}$ polynomial-time computable from 1^n), and all $\delta : \mathbb{N} \to [0,1]$ with $\delta(n) \geq 2^{-\mathrm{poly}(n)}$, there are polynomials $\ell(\lambda), n(\lambda)$ and a $\{$polynomially secure, $(\frac{1}{\delta(n(\lambda))}, \delta(n(\lambda)))$-secure$\}$ puncturable PRF family*

$$\mathcal{F}_m = \left\{ F_{\lambda,s} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(n(\lambda))} \right\}_{\lambda \in \mathbb{N}, s \in \{0,1\}^{\ell(\lambda)}}.$$

a) Puncturing Arguments.: Throughout this paper, we will frequently make use of "puncturing arguments" [7], which capture the fact that for an indistinguishability obfuscator $i\mathcal{O}$ and puncturable pseudorandom function $F_{\mathsf{sk}} : \{0,1\}^n \to \{0,1\}^m$, for every input $x^* \in \{0,1\}^n$, the distribution on obfuscated programs

$$P \leftarrow i\mathcal{O}\left(x \mapsto F_{\mathsf{sk}}(x) \right)$$

for uniformly random sk is computationally indistinguishable from the following distribution on obfuscated programs

$$P_{x^*,r} \leftarrow i\mathcal{O}\left(x \mapsto \begin{cases} r, & \text{if } x = x^*, \\ F_{\mathsf{sk}}(x), & \text{otherwise} \end{cases} \right)$$

for uniformly random sk and uniformly random $r \leftarrow \{0,1\}^m$.

This indistinguishability holds by a simple hybrid argument involving the following additional distributions:

$$H_1 \leftarrow i\mathcal{O}\left(x \mapsto \begin{cases} F_{\mathsf{sk}}(x^*), & \text{if } x = x^*, \\ F_{\mathsf{sk}\langle x^* \rangle}(x), & \text{otherwise} \end{cases} \right),$$

$$H_2 \leftarrow i\mathcal{O}\left(x \mapsto \begin{cases} r, & \text{if } x = x^*, \\ F_{\mathsf{sk}\langle x^* \rangle}(x), & \text{otherwise} \end{cases} \right).$$

III. Faster Goldwasser-Sipser

In this section, we give an efficient non-deterministic algorithm for the following computational problem.

Definition III.1 (Non-Deterministic Multiplicative Circuit Approximation Probability Problem)**.** nmCAPP is the following problem:

- **Given:** a non-deterministic circuit $C : \{0,1\}^n \to \{0,1\}^w \to \{0,1\}$, a $\beta \in [2^n]$ and an $\frac{1}{\varepsilon} \in \mathbb{N}$
- **Accept:** if $\Pr_x[C(x) = 1] \geq (1+\varepsilon)\frac{\beta}{2^n}$.
- **Reject:** if $\Pr_x[C(x) = 1] < \frac{\beta}{2^n}$.

Goldwasser and Sipser [48] showed nmCAPP can be solved by an efficient Arthur-Merlin protocol with proof length and verification time $\mathrm{poly}(n, w) \cdot \varepsilon^{-2}$. We improve the dependence on ε to ε^{-1}, which is important for our later results.

Theorem III.2 (Improved Goldwasser-Sipser)**.** *There is an Arthur-Merlin algorithm for solving* nmCAPP *that runs in time $\frac{1}{\varepsilon} \cdot |C| \cdot \mathrm{poly}(n)$. Consequently,*[5] *there is a non-deterministic circuit solving* nmCAPP *of size $\frac{1}{\varepsilon} \cdot |C| \cdot \mathrm{poly}(n)$.*

Proof. We first give some intuition, focusing on the case $\beta = 2^{n-1}$. In this case, a protocol running in time $|C| \cdot 1/\varepsilon^2$ is trivial: the verifier samples $1/\varepsilon^2$ instances $x_1, \ldots, x_{1/\varepsilon^2}$ and asks the prover to send witnesses for at least a $1/2 + \varepsilon$ fraction of them. The key point is that $O(1/\varepsilon^2)$ samples are necessary and sufficient to distinguish the two cases with constant probability.

We improve on this complexity by making more non-trivial use of the prover's non-determinism, inspired by [48], who do this in order to handle arbitrary choices of β. The way in which we use the prover's non-determinism is in identifying *rare* events about a pseudorandom sequence of instances. Roughly speaking, the verifier specifies a long pseudorandom sequence of instances to the prover (using a pairwise-independent hash function (Lemma II.3)), and the prover searches for $\ell = O(1/\varepsilon)$ consecutive instances that have valid witnesses. The prover communicates the index of the first instance in this subsequence as well as all ℓ witnesses, and the verifier checks the validity of the ℓ witnesses.

Formally, the AM-protocol is described as follows:

Given $C : \{0,1\}^n \times \{0,1\}^w \to \{0,1\}$ and k and ε:

1) Set $\ell = c/\varepsilon + c$ and $m = (\frac{2^n}{\beta})^{\ell - c}$ where $c \in \mathbb{N}$ is a sufficiently large constant we choose later.

[5] This follows from Lemma II.1

2) Arthur samples a key $k \in \{0,1\}^{\tilde{O}(\ell \cdot n)}$ for the pairwise independent hash function $H_k : [m] \to (\{0,1\}^n)^\ell$ from Lemma II.3. Arthur sends this key to Merlin.
3) Arthur accepts if Merlin sends back an index $i \in [m]$ and witnesses $w_1, \ldots, w_\ell$ such that for all $j \in [\ell]$ we have $C(H_k(i)_j, w_j) = 1$ where $H_k(i)_j$ is the j'th entry of the tuple $H_k(i)$.

Since $C : \{0,1\}^n \times \{0,1\}^w \to \{0,1\}$, we assume (without loss of generality) that $\varepsilon \geq 2^{-n}$ and hence $\log(\frac{1}{\varepsilon}) \leq n$. Then it is easy to see that this algorithm runs in time

$$\tilde{O}(\ell \cdot n) + O(\ell \cdot |C|) = \frac{1}{\varepsilon} \cdot \Big(\mathrm{poly}(n) + O(|C|)\Big),$$

as desired.

It remains to argue for correctness. For $i \in [m]$, let X_i be the indicator event that $C(H_k(i)_j) = 1$ for all $j \in [\ell]$. Let $X = \sum_{i \in [m]} X_i$. Let $p = \Pr_x[C(x) = 1]$. By the pairwise independence of H_k, we get that

$$\mathbb{E}[X] = \sum_{i \in [m]} \mathbb{E}[X_i] = mp^\ell$$

and

$$\mathsf{Var}[X] = \sum_{i \in [m]} \mathsf{Var}[X_i] = m \cdot p^\ell \cdot (1 - p^\ell) \leq mp^\ell = \mathbb{E}[X].$$

By Chebyshev's inequality,

$$\Pr_k \left[|X - \mathbb{E}[X]| \geq 2\sqrt{\mathbb{E}[X]} \right] \leq \frac{1}{4}.$$

Thus, when $p \leq \frac{\beta}{2^n}$, we get that

$$\mathbb{E}[X] = (\frac{\beta}{2^n})^\ell m \leq 2^{-c},$$

so

$$\Pr_k[X \geq 1] \leq \frac{1}{4}$$

by setting c to be a sufficiently large constant. This shows the protocol is sound.

On the other hand, when $p \geq (1+\varepsilon)\frac{\beta}{2^n}$, we get that

$$\begin{aligned} \mathbb{E}[X] &= p^\ell m \\ &\geq (1+\varepsilon)^\ell (\frac{\beta}{2^n})^\ell m \\ &= (1+\varepsilon)^{(1+1/\varepsilon)c} 2^{-c} \\ &\geq e^c 2^{-c} \geq 2^{\Omega(c)} \end{aligned}$$

so

$$\Pr_k[X < 1] \leq \frac{1}{4}$$

by setting c to be a sufficiently large constant. This shows the protocol is complete. □

IV. Fine-grained hardness reductions for SAT

In this section, we discuss various forms of hardness for SAT and give fine-grained relationships between them.

A. Hardness Assumptions for SAT

We begin by listing two of the strongest forms of hardness that may hold. The first is optimal hardness for (the decision version) of SAT.

Assumption IV.1 (Optimal Hardness of SAT)**.** There exists a polynomial $p(n)$ such that SAT on circuits of size $p(n)$ and input length n requires time at least $2^n \cdot \frac{1}{p(n)}$ for all n.

The second, stronger assumption is optimal co-nondeterministic hardness of SAT.

Assumption IV.2 (Optimal Hardness of Refuting Circuit SAT)**.** There exists a polynomial $p(n)$ such that UNSAT on circuits of size $p(n)$ and input length n requires *nondeterministic circuits* of size at least $2^n \cdot \frac{1}{p(n)}$ for all n.

For our result on *post-quantum* one-way functions, we additionally make use of the following incomparable assumption: optimal quantum hardness of SAT.

Assumption IV.3 (Optimal Quantum Hardness of SAT)**.** There exists a polynomial $p(n)$ such that SAT on circuits of size $p(n)$ and input length n requires quantum time at least $2^{n/2} \cdot \frac{1}{p(n)}$ for all n.

As in the case of probabilistic algorithms, we say that a quantum algorithm solves SAT if it is correct on all inputs with probability $2/3$. We remark that Grover's algorithm runs in quantum time roughly $2^{n/2} \cdot p(n)$, so the assumption asserts that this runtime is optimal up to polynomial factors.

B. Reductions to Unique Versions

The celebrated work of Valiant and Vazirani [47] gives a probabilistic reduction from SAT to Unique-SAT (where one is promised that a circuit has at most one satisfying assignment). They do this by giving a probabilistic polynomial time algorithm that, given a satisfiable circuit, outputs another circuit that with $1/\mathrm{poly}(n)$ probability has exactly one satisfying assignment.

Theorem IV.4 (Valiant-Vazirani [47])**.** *There is a probabilistic polynomial-time algorithm* VV *that given a circuit* $\varphi : \{0,1\}^n \to \{0,1\}$ *outputs a circuit* $\varphi' : \{0,1\}^n \to \{0,1\}$ *with the following properties:*

- $|\varphi'| = |\varphi| + \mathrm{poly}(n)$.
- *If* φ *is unsatisfiable, then* φ' *is unsatisfiable.*
- *If* φ *is satisfiable, then with probability at least* $\frac{1}{\mathrm{poly}(n)}$ *there is a unique satisfying assignment to* φ'.

An immediate corollary of Valiant-Vazirani is a fine-grained reduction from Search-SAT to Search-Unique-SAT.

Corollary IV.5 (Search Hardness implies Unique Search Hardness)**.** *If* Search-Unique-SAT$[n, s]$ *is solvable in* t*-time with* ε*-success, then* Search-SAT$[n, s - \mathrm{poly}(n)]$ *is solvable in* $(t + \mathrm{poly}(n))$*-time with* $\frac{\varepsilon}{\mathrm{poly}(n)}$ *success.*

Proof. Let A be the t-time ε-success circuit for Search-Unique-SAT$[n, s]$. Consider the circuit B that given

an instance φ of Search-SAT$[n, s']$ outputs $A(\mathsf{VV}(\varphi))$ where VV is the Valiant-Vazirani procedure from Theorem IV.4. Set $s' = s - \mathrm{poly}(n)$ so that the output of $\mathsf{VV}(\varphi)$ has size at most s.

B clearly has size $t + \mathrm{poly}(n)$. Since VV transforms a satisfiable circuit to a uniquely satisfiable circuit with probability $1/\mathrm{poly}(n)$, we have that B solves Search-SAT$[n, s']$ with probability $\varepsilon/\mathrm{poly}(n))$. □

Similarly, another corollary is a fine-grained reduction from UNSAT to Unique-UNSAT.

Corollary IV.6 (Refutation Hardness implies Unique Refutation Hardness)**.** *If there is a t-size non-deterministic circuit for* Unique-UNSAT$[n, s]$*, then there is a $t \cdot \mathrm{poly}(n, s)$-size non-deterministic circuit for* UNSAT$[n, s - \mathrm{poly}(s)]$.

Proof. Let A be the t-size non-deterministic circuit for Unique-UNSAT. Let B the Arthur-Merlin circuit that works as follows. It uses its randomness to sample $\varphi'_1, \ldots, \varphi'_q \leftarrow \mathsf{VV}(\varphi)$, where $q = \mathrm{poly}(n)$ is sufficiently large. It accepts if it is given witnesses that $A(\varphi'_i) = 1$ for all i.

First, we show completeness. If φ is unsatisfiable, then φ'_i is unsatisfiable for all i. Hence $A(\varphi'_i)$ accepts for all i. So in particular, there are witnesses that $A(\varphi'_i) = 1$ for all i. This proves completeness.

Next, we show soundness. If φ is satisfiable, then by setting $q = \mathrm{poly}(n)$ sufficiently large, we have that at least one of φ'_i is uniquely satisfiable with high probability. For that i, we have that $A(\varphi'_i) = 0$, so there are no witnesses that $A(\varphi'_i) = 1$. This proves soundness.

Finally, we argue for efficiency. It is easy to see that $|B| = \mathrm{poly}(q)t = \mathrm{poly}(n)t$. Using Adleman's trick (Lemma II.1), we can convert B from an Arthur-Merlin circuit to a non-deterministic circuit of size $t \cdot \mathrm{poly}(n, s)$. □

C. Success Probability Amplification for Search-SAT

Assumption IV.1 and Assumption IV.3 were stated for algorithms that are correct with constant probability. We briefly remark that such assumptions easily imply optimal hardness of Search-SAT (but not SAT!) for algorithm with low success probability.

Remark IV.7. *Given a time T algorithm solving* Search-SAT *with probability ε, there is a time $O(T/\varepsilon)$ algorithm solving* Search-SAT *with probability* .99.

The reduction is trivial: run the low success probability algorithm $O(1/\varepsilon)$ times and check if a satisfying assignment is found.

The quantum case is not as simple, but is also well-understood.

Remark IV.8. *Given a time T algorithm (classical or quantum) solving* Search-SAT *with probability ε, there is a time $O(T/\sqrt{\varepsilon})$* quantum *algorithm solving* Search-SAT *with probability* .99.

This can be accomplished by emplying a variant of amplitude amplification [56] that is oblivious to the initial success probability (see [56] Theorem 3).

Since both of these reductions work with respect to a fixed Search-SAT instance, they also apply to restricted versions of the problem such as Search-Unique-SAT.

D. Hardness of Unique SAT in the Low Success Regime

For our applications, it will be useful to have hardness of solving Unique-SAT even with probability $1/2 + \varepsilon$ for very small ε (e.g., $\varepsilon = 2^{-.99n}$). In this regime of parameters, the standard search-to-decision reduction for SAT breaks down (it seems to require running the Unique-SAT algorithm at least $\frac{1}{\varepsilon^2}$ times, which is too much). Relatedly, we do not know how to do a fine-grained Valiant-Vazirani transformation from SAT to Unique-SAT in this setting.

Instead, we show that there is a fine-grained reduction from non-deterministically solving UNSAT to low-success algorithms for Unique-SAT. The reduction crucially relies on our improved Goldwasser-Sipser result. Intuitively, this makes sense as we would like to avoid paying a ε^{-2} cost.

Theorem IV.9 (Refutation Hardness implies Decisional Unique Hardness)**.** *If* Unique-SAT$[n, s]$ *is solvable in t-time with $(1/2+\varepsilon)$-success, then there is a non-deterministic circuit for* UNSAT$[n, s]$ *of size at most* $\mathrm{poly}(n) \cdot t \cdot \frac{1}{\varepsilon}$.

Proof. Let A be the t-time $(1/2 + \varepsilon)$-success circuit for Unique-SAT$[n, s]$. We will show how to solve Unique-UNSAT$[n, s]$ and then appeal to Corollary IV.6.

If φ is unsatisfiable, then

$$\Pr_A[A(\varphi) = 0] \geq 1/2 + \varepsilon. \tag{1}$$

On the other hand, if φ is uniquely satisfiable, we have that

$$\Pr_A[A(\varphi) = 0] \leq 1/2 - \varepsilon. \tag{2}$$

Thus consider the following non-deterministic circuit B for solving Unique-UNSAT$[n, s - \mathrm{poly}(n)]$ on input φ:

- Accept if running the improved Goldwasser-Sipser non-deterministic circuit (Theorem III.2) for nmCAPP to distinguish whether $\Pr_A[A(\varphi) = 0]$ is at most $1/2$ or at least $1/2 + \varepsilon$ outputs "at least $1/2 + \varepsilon$."

By construction B is a non-deterministic circuit of size $t \cdot \frac{1}{\varepsilon} \cdot \mathrm{poly}(n)$. B accepts all unsatisfiable circuits with probability $2/3$ because of (1). B rejects all satisfiable circuits with probability $2/3$ because of (2). The theorem then follows from Corollary IV.6. □

V. Optimal Hardness of OWFs, PRGs, and OWPFs

In this section, we prove our fine-grained worst-case to average-case reduction for SAT in the search, decision, and k-fold search settings. We prove the following theorem.

Theorem V.1. *Suppose that sub-exponentially secure $i\mathcal{O}$ and OWFs exist. Then:*

- *If* SAT *is optimally hard (Assumption IV.1), then for all* (T, ε) *such that* $\frac{T(n)}{\varepsilon(n)} \leq 2^n \cdot n^{-\omega(1)}$*, there exist* (T, ε)*-secure OWF families.*
- *If refuting Circuit-SAT is optimally hard (Assumption IV.2), then for all* (T, ε) *such that* $\frac{T(n)}{\varepsilon(n)} \leq 2^n \cdot n^{-\omega(1)}$*, there exist* (T, ε)*-secure PRG families.*
- *Let* $\varepsilon(n) \geq 2^{-n}$*. If* Search-Unique-$k$-fold-SAT *is* (T, ε^k)*-hard for all* $k \leq \ell$*, then there exist* $(T, (\varepsilon + 2^{-n})^\ell)$*-secure OWPF families.*
- *If* SAT *is optimally hard for quantum algorithms (Assumption IV.3), and* post-quantum *sub-exponentially secure* $i\mathcal{O}$ *and OWFs exist, then for all* (T, ε) *such that* $\frac{T(n)^2}{\varepsilon(n)} \leq 2^n \cdot n^{-\omega(1)}$*, there exist* (T, ε)*-secure post-quantum OWF families.*

The construction of our candidate OWF/PRG/OWPF family is described as follows:

- $\mathsf{Gen}(1^n, 1^k)$: sample public parameters
$$\mathsf{pp} = P \leftarrow i\mathcal{O}\Big(x \mapsto F_{\mathsf{sk}}(x)\Big),$$
where $i\mathcal{O}$ is an indistinguishability obfuscator and $\{F_{\mathsf{sk}} : \{0,1\}^n \to \{0,1\}^{10nk}\}$ is a puncturable PRF family. To obtain a OWF or PRG, the parameter k is set to 1.
- $\mathsf{Eval}(\mathsf{pp}, x)$: to evaluate the function on n, simply run the program:
$$\mathsf{Eval}(\mathsf{pp}, x) = P(x).$$
where $P(x)$ denotes obfuscated program evaluation.

Throughout this section, we assume that the $i\mathcal{O}$ scheme is subexponentially secure, and that $\{F_{\mathsf{sk}}\}$ is a subexponentially secure puncturable PRF. The security parameters for these schemes are set to be a sufficiently large $\mathrm{poly}(n, k)$ to obtain exponential indistinguishability in the security proofs below. In particular, the PRF security parameter is chosen large enough so that f_{pp} is injective with probability at least $1 - 2^{-10kn}$ over the choice of pp.

We now proceed to prove Theorem V.1. Specifically, we prove the second and third bullet points of the theorem statement. The first and fourth bullet points follow from the third bullet point, observing (for the fourth bullet point) that the security proof applies equally well to quantum adversaries.

A. Proof of Pseudorandomness

In this section, we additionally assume the $(T + \mathrm{poly}(n), \varepsilon - O(2^{-2n}))$-hardness of Unique-SAT. By Theorem IV.9, this assumption follows from the $(T + \mathrm{poly}(n)) \cdot 1/\varepsilon$-hardness of refuting SAT, whenever $\varepsilon \geq 2^{-n}$.

Lemma V.2. *$f(\mathsf{crs}, \cdot)$ is a $(T, O(\varepsilon))$-secure PRG.*

Proof. We want to show that for a uniformly random $x^* \leftarrow \{0,1\}^n, r \leftarrow \{0,1\}^m$,
$$(P, P(x^*)) \approx_{T, O(\varepsilon)} (P, r).$$
If $F_{\mathsf{sk}}(\cdot)$ is a $(2^{2n}, 2^{-2n})$-secure puncturable PRF and $i\mathcal{O}$ is $(2^{2n}, 2^{-2n})$-secure, then we know by a puncturing argument that (even given x^*)
$$P \approx_{T, \varepsilon} P_{x^*, r^*},$$
where P_{x^*, r^*} is an obfuscated program of the form
$$P_{x^*, r^*} \leftarrow i\mathcal{O}\Big(x \mapsto F_{\mathsf{sk}}(x) \text{ if } x \neq x^*, \text{ else } r^*\Big)$$
and $r^* \leftarrow \{0,1\}^m$ is a uniformly random string.

We now show that if our claimed indistinguishability does *not* hold, then there is a time $T + \mathrm{poly}(n, \lambda)$ algorithm solving decisional unique-SAT with advantage ε. Given any distinguisher D, the worst-case algorithm is described as follows.

Algorithm $\mathcal{A}$ for Unique-SAT

Given $\varphi : \{0,1\}^n \to \{0,1\}$:

1) Sample randomness $a \leftarrow \{0,1\}^n, r^* \leftarrow \{0,1\}^m, r \leftarrow \{0,1\}^m$.
2) Sample an obfuscated program
$$P_{\varphi, a, r^*} \leftarrow i\mathcal{O}\Big(x \mapsto F_{\mathsf{sk}}(x) \text{ if } \varphi(x \oplus a) = 0, \text{ else } r^*\Big).$$
3) Sample a bit b. If $b = 0$, set $y = r^*$. If $b = 1$, set $y = r$.
4) Run the distinguisher
$$D(P_{\varphi, a, r^*}, y),$$
obtaining an outcome b'.
5) Output "satisfiable" if $b' = b$, else output "unsatisfiable."

Claim V.3. *If φ is unsatisfiable, then the algorithm $\mathcal{A}(\varphi)$ accepts with probability at most $\leq 1/2 + O(2^{-2n})$.*

Claim V.3 holds because of the following computational indistinguishability in the presence of (a, r^*, r):
$$P_{\varphi, a, r^*} \approx_{2^{10n}, 2^{-10n}} i\mathcal{O}\Big(x \mapsto F_{\mathsf{sk}}(x)\Big),$$
which holds by $i\mathcal{O}$ security and the unsatisfiability of φ. The latter obfuscated program does not depend on r or r^*, completing the proof of the claim.

Claim V.4. *If φ is* uniquely *satisfiable, then the algorithm $\mathcal{A}(\varphi)$ accepts with probability at least $\leq 1/2 + \varepsilon - O(2^{-2n})$.*

For any uniquely satisfiable φ, let $w \in \{0,1\}^n$ denote the unique satisfying assignment to φ. Claim V.4 holds because of the following computational indistinguishability in the presence of (a, r^*, r):
$$P_{\varphi, a, r^*} \approx_{2^{10n}, 2^{-10n}} P_{w \oplus a, r^*},$$
which holds by $i\mathcal{O}$ security and the fact that $\varphi(x \oplus a)$ is functionally equivalent to the indicator function $\chi(x = w \oplus a)$.

Thus, the circuit-output pair $(P_{\varphi, a, r^*}, r^*)$ is distributionally $(2^{10n}, 2^{-10n})$-indistinguishable from (P_{x^*, r^*}, r^*), while (P_{φ, a, r^*}, r) is distributionally $(2^{10n}, 2^{-10n})$-indistinguishable from (P_{x^*, r^*}, r). This completes the proof of the claim.

Combining Claim V.3 and Claim V.4, we conclude the proof of Lemma V.2.

□

B. Proof of Product One-Wayness

In this section, we assume that Search-Unique-k-fold-SAT is $(T + \mathrm{poly}(k,n), \varepsilon^k - 2^{-kn})$-hard for every $k \le \ell$.

Lemma V.5. *$\mathcal{F}$ is $(T, (\varepsilon + 2^{-n})^\ell)$-hard to invert ℓ times in parallel.*

Proof. We have set parameters so that the probability that f_{pp} is not injective is at most 2^{-10kn}. We prove a slightly modified definition of product one-wayness.

Claim V.6. *For all time-T adversaries A run on random* distinct *OWF outputs, we have:*

$$\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Gen}(1^n)\\ \text{distinct } x_1^*,\ldots,x_k^*\leftarrow\{0,1\}^n}}\Big[A(\mathsf{pp}, f_{\mathsf{pp}}(x_1^*),\ldots,f_{\mathsf{pp}}(x_k^*)) = (x_1^*,\ldots,x_k^*)\Big] = O(\varepsilon^k).$$

This claim implies the lemma by the following calculation:

$$\begin{aligned}
&\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Gen}(1^n)\\ \text{i.i.d. } x_1^*,\ldots,x_\ell^*\leftarrow\{0,1\}^n\\ y_i^*=f_{\mathsf{pp}}(x_i^*)}}\Big[A(\mathsf{pp},(y_i^*)_{i=1}^\ell) = (x_i^*)_{i=1}^\ell\Big]\\
=&\sum_{k\le\ell}\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Gen}(1^n)\\ \text{i.i.d. } x_1^*,\ldots,x_\ell^*\leftarrow\{0,1\}^n\\ y_i^*=f_{\mathsf{pp}}(x_i^*)}}\Big[|\{x_i^*\}_{i=1}^\ell| = k \wedge A(\mathsf{pp},(y_i^*)_{i=1}^\ell)) = (x_i^*)_{i=1}^\ell\Big]\\
\le&\sum_{k\le\ell}\binom{k}{\ell}2^{-n(\ell-k)}\cdot O(\varepsilon^k)\\
\le&O((\varepsilon+2^{-n})^\ell).
\end{aligned}$$

The second-to-last inequality holds by the claim (since the information of which x_i^* are equal can be hard-coded as polynomial-length advice in a reduction), and the last inequality holds by the binomial theorem.

Thus, it suffices to prove the claim.

Suppose that a time T adversary $\mathcal{A}(\mathsf{pp}, y_1^*,\ldots,y_k^*) \to (x_1^*,\ldots,x_k^*)$ that, given pp and $y_i^* = F_{\mathsf{sk}}(x_i^*))$ for *distinct* inputs x_i^*, successfully k-fold inverts the OWF with probability $\ge \varepsilon^k$. By a puncturing argument, up to a 2^{-10nk} additive loss, the adversary must still succeed on an obfuscated program $P_{x_1^*,r_1^*,\ldots,x_k^*,r_k^*}$ of the form

$$P_{x_1^*,r_1^*,\ldots,x_k^*,r_k^*} \leftarrow i\mathcal{O}\Big(x \mapsto r_i^* \text{ if } x = x_i^*, \text{ else } F_{\mathsf{sk}}(x)\Big),$$

where each r_i^* is sampled uniformly at random. Using this, we construct an algorithm for Search-Unique-k-fold-SAT:

Algorithm $\mathcal{A}$ for Search-Unique-k-fold-SAT

Given $\Phi = (\varphi_1 : \{0,1\}^n \to \{0,1\}, \ldots, \varphi_k : \{0,1\}^n \to \{0,1\})$:

1) Sample uniformly random strings $\alpha_i \leftarrow \{0,1\}^n$ and $r_i^* \leftarrow \{0,1\}^n$.
2) Sample an obfuscated program
$$P_{\Phi,\alpha_1,\ldots,\alpha_k,r_1^*,\ldots,r_k^*} \leftarrow i\mathcal{O}\Big(x \mapsto F_{\mathsf{sk}}(x) \text{ if } \varphi_i(x\oplus\alpha_i) = 0 \text{ for all } i, \text{ else output } r_i^* \text{ for the smallest } i \text{ such that } \varphi_i(x\oplus\alpha_i) = 1\Big).$$
3) Run the algorithm
$$A(P_{\varphi,\alpha_1,\ldots,\alpha_k,r_1^*,\ldots,r_k^*}, r_1^*,\ldots,r_k^*),$$
obtaining inputs $x_1',\ldots,x_k'$.
4) Return $x_1'\oplus\alpha_1,\ldots,x_k'\oplus\alpha_k$.

We claim that $\mathcal{A}$ succeeds with probability at least $\frac{1}{2}\Big(\varepsilon^k - 10\cdot 2^{-10nk}\Big)$. Indeed, letting $w_1,\ldots,w_k$ denote the unique witnesses for $\varphi_1,\ldots,\varphi_k$, we show:

$$\Pr\Big[\mathcal{A} \text{ succeeds and } |\{w_i\oplus\alpha_i\}| = k\Big] \ge \frac{1}{2}\Big(\varepsilon^k - 10\cdot 2^{-10nk}\Big).$$

To see this, we observe that $\Pr\Big[|\{w_i\oplus\alpha_i\}| = k\Big] \ge 1/2$ and the *conditional* probability of success is, by definition, the probability that $\mathcal{A}$ succeeds when $\alpha_1,\ldots,\alpha_k$ are sampled uniformly random subject to the strings $\alpha_i\oplus w_i$ being distinct. For such strings, the program $P_{\Phi,\alpha_1,\ldots,\alpha_k,r_1^*,\ldots,r_k^*}$ is 2^{-10nk}-indistinguishable from the obfuscated program

$$P'_{\Phi,\alpha_1,\ldots,\alpha_k,r_1^*,\ldots,r_k^*} \leftarrow i\mathcal{O}\Big(x \mapsto r_i \text{ if } x = \alpha_i\oplus w_i, \text{ else } F_{\mathsf{sk}}(x)$$

by $i\mathcal{O}$ security. In the above description, the strings $\alpha_i \oplus w_i$ are hard-coded (rather than computed). This distribution on obfuscated programs is *identical* to $P_{x_1^*,\ldots,x_k^*,r_1^*,\ldots,r_k^*}$ for uniform distinct x_i^* and uniform r_i^*, completing the proof. □

VI. Hardness of k-fold SAT

In this section, we prove the following theorem.

Theorem VI.1. *Assume subexponentially secure iO and subexponentially secure puncturable PRFs exist. Whenever* Search-Unique-k-fold-SAT$[n, s' = \mathrm{poly}(s,n,k)]$ *is solvable in T-time with ε-success, there is a $\frac{1}{\varepsilon^{1/k}}\cdot T\cdot\mathrm{poly}(n,k,s)$-size non-deterministic circuit for* UNSAT$[n,s]$.

By combining this with Theorem V.1, we obtain the following result.

Theorem VI.2. *If refuting Circuit-SAT is $\frac{1}{\varepsilon^{1/k}}\cdot T\cdot\mathrm{poly}(n,s)$-hard, then there exist (T,ε)-secure (injective) k-OWPF families.*

In the "optimal" parameter regime, this gives the following corollary.

Corollary VI.3. *If refuting Circuit-SAT is optimally hard (Assumption IV.2), then there exists a k-OWPF family with $\Big(T, \Big(\frac{T\cdot\mathrm{poly}(n,k)}{2^n}\Big)^k\Big)$-security for all T.*

We will prove Theorem VI.1 in two parts.

- Section VI-A: First, we show than any algorithm solving Search-Unique-k-fold-SAT can be made "oblivious."

This roughly means that the success probability of the algorithm only depends on the number of satisfying assignments of the underlying formulas and nothing else. To do this, we will use ideas from cryptography, specifically indistinguishability obfuscation and puncturable pseudorandom functions.

- Section VI-B: Next, we show how to use an oblivious algorithm for Search-Unique-k-fold-SAT to nondeterministically solve UNSAT. This will crucially rely on the optimal parameters in our our improved Goldwasser-Sipser algorithm.

A. Making k-fold SAT Solvers Oblivious

First, we introduce some helpful notation:

- For $x, y, \delta \in \mathbb{R}$, we write $x \approx_\delta y$ to mean that $|x-y| < \delta$.
- Let $\mathsf{SIZE}_n[s]$ denote the set of s-size circuits with n-inputs.
- Let Φ denote the tuple of circuits $(\varphi_1, \ldots, \varphi_k)$. We will write $|\Phi| = |\Phi'|$ if $|\varphi_i| = |\varphi_i'|$ for all i.
- Let $\#\mathsf{SAT}(\varphi)$ denote the number of satisfying assignments to φ. Let $\#\mathsf{SAT}(\Phi)$ denote the corresponding tuple $(\#\mathsf{SAT}(\varphi_1), \ldots, \#\mathsf{SAT}(\varphi_k))$
- Let $\mathsf{SAT}(\Phi)$ denote the set of k-tuples of satisfying assignments to the corresponding φ_i. In other words,

$$\mathsf{SAT}(\Phi) = \{(w_1, \ldots, w_k) : \varphi_i(w_i) = 1 \ \ \forall i \in [k]\}.$$

Next, we make a definition of what it means for a k-fold SAT solver to be oblivious. Informally, it means that the chance of outputting any $w \in \mathsf{SAT}(\Phi)$ only depends on $\#\mathsf{SAT}(\Phi)$ and nothing else about Φ or w.

Definition VI.4 (Oblivious k-fold SAT solver)**.** A probabilistic circuit C for k-fold SAT is *β-oblivious* if whenever $|\Phi| = |\Phi'|$, $\#\mathsf{SAT}(\Phi) = \#\mathsf{SAT}(\Phi')$, $w \in \mathsf{SAT}(\Phi)$ and $w' \in \mathsf{SAT}(\Phi')$, we have

$$\Pr[C(\Phi) = w] \approx_\beta \Pr[C(\Phi') = w'].$$

Equivalently, there exists a function $p : ([2^n])^k \to \mathbb{R}$ such that

$$\Pr[C(\Phi) = w] \approx_{\beta/2} \frac{p(\#\mathsf{SAT}(\Phi))}{|\#\mathsf{SAT}(\Phi)|}$$

for all $w \in \mathsf{SAT}(\phi)$.

Lemma VI.5 (Oblivious is Without Loss of Generality)**.** *Assume subexponentially secure iO and subexponentially secure puncturable PRFs exist. Let $C' : (\mathsf{SIZE}_n[s'])^k \to (\{0,1\}^n)^k$ be a probabilistic circuit. There is another probabilistic circuit $C : (\mathsf{SIZE}_n[s])^k \to (\{0,1\}^n)^k$ where $s = (\frac{s'}{\mathrm{poly}(n,k)})^{\Omega(1)}$ and $|C| = |C'| + \mathrm{poly}(n,k,s)$ such that*

1) *For all Φ, we have that $\Pr[C(\Phi) \in \mathsf{SAT}(\Phi)]$ is at least*

$$\left(\prod_{i\in[k]} \frac{1}{4 \cdot \#\mathsf{SAT}(\varphi_i)}\right) \cdot \left(\min_{\Phi' : \#\mathsf{SAT}(\Phi') = \#\mathsf{SAT}(\Phi)} \Pr[C'(\Phi') \in \mathsf{SAT}(\Phi')] - 2^{-10nk}\right)$$

2) *C is a 2^{-10nk}-oblivious k-fold SAT solver*

Proof. First we construct C. Let $\lambda = \lambda(n,k) = \mathrm{poly}(n,k)$ be a sufficiently large polynomial we set later.

Circuit C

Given $\varphi_1, \ldots, \varphi_k : \{0,1\}^n \to \{0,1\}$:

1) For each $i \in [k]$, sample a secret key sk_i for a puncturable PRF $F_i = F_{\mathsf{sk}_i} : \{0,1\}^n \to \{0,1\}^n$ with security parameter λ
2) Define Φ' by $\varphi_i' \leftarrow i\mathcal{O}(x \mapsto \varphi_i(F_i(x))$ with security parameter λ and where we pad the circuit being obfuscated to size at least λ.
3) Let $w' = (w_1', \ldots, w_k') \leftarrow C'(\Phi')$.
4) Output $w = (F_1(w_1'), \ldots, F_k(w_k'))$.

Note that by construction $|C| = |C'| + \mathrm{poly}(n, \lambda, k, s) = |C| + \mathrm{poly}(n,k,s)$. Also, we need $s' = |\varphi'| = \mathrm{poly}(|\varphi|, \lambda, n) = \mathrm{poly}(|\varphi|, n) = \mathrm{poly}(s,n)$. Hence we have that $s = (\frac{s'}{\mathrm{poly}(n)})^{\Omega(1)}$. It remains to show that C has properties (1) and (2).

Claim VI.6. *C has property (1).*

Proof. Fix Φ and let $i \in [k]$ be arbitrary. If F_i were a truly random function, then the random variable $\#\mathsf{SAT}(\varphi_i')$ would be a binomial random variable with $N = 2^n$ trials and probability $q = \frac{\#\mathsf{SAT}(\varphi_i)}{2^n}$ of success in each trial. The binomial distribution has the property that if $N \cdot q \in \mathbb{Z}$ (as it is in our case), then its mean, median, and mode is exactly $N \cdot q$, which in our case is exactly $\#\mathsf{SAT}(\varphi)$. Thus, we get that

$$\Pr_{\text{truly random } F_i}[\#\mathsf{SAT}(\varphi_i') = \#\mathsf{SAT}(\varphi_i)] \geq \frac{1}{4\#\mathsf{SAT}(\varphi_i)}$$

by applying a Markov inequality and averaging argument and using that $\#\mathsf{SAT}(\varphi)$ is both the mean and the mode of the distribution. Hence, we get that

$$\Pr_{\text{truly random } F_1, \ldots, F_k}[\#\mathsf{SAT}(\Phi') = \#\mathsf{SAT}(\Phi)] \geq \prod_{i\in[k]} \frac{1}{4\#\mathsf{SAT}(\varphi_i)}.$$

Since F_i is subexponentially secure, we can set $\lambda = \mathrm{poly}(n,k)$ sufficiently large, so that

$$\Pr_{F_1, \ldots, F_k \leftarrow PRF}[\#\mathsf{SAT}(\Phi') = \#\mathsf{SAT}(\Phi)] \geq \left(\prod_{i\in[k]} \frac{1}{4\#\mathsf{SAT}(\varphi_i)}\right) - 2^{-10nk}.$$

Thus, we have

$$\begin{aligned} &\Pr[C(\Phi) \in \mathsf{SAT}(\Phi)] \\ &= \Pr[C'(\Phi') \in \mathsf{SAT}(\Phi')] \\ &\geq \left(\prod_{i\in[k]} \frac{1}{4\#\mathsf{SAT}(\varphi_i)}\right) \cdot \\ &\quad \left(\min_{\Phi'' : \#\mathsf{SAT}(\Phi'') = \#\mathsf{SAT}(\Phi)} \Pr[C'(\Phi'') \in \mathsf{SAT}(\Phi'')] - 2^{-10nk}\right). \end{aligned}$$

Now, it remains to show property (2).

Claim VI.7. *C has property (2).*

Proof. For this proof, we introduce some convenient notation for tuples of circuits:

- Given PRF instances $F_1, \ldots, F_k : \{0,1\}^n \to \{0,1\}^n$, we define $F : (\{0,1\}^n)^k \to (\{0,1\}^n)^k$ as $F(w) = (F_1(w_1), \ldots, F_k(w_k))$.
- For a tuple Φ of circuits $(\varphi_1, \ldots, \varphi_k)$, we define $\Phi \circ F$ to be the tuple $(\varphi_1 \circ F_1, \ldots, \varphi_k \circ F_k)$.
- For a tuple $\Phi = (\varphi_1, \ldots, \varphi_k)$, we define the notation $i\mathcal{O}(\Phi)$ to mean $(i\mathcal{O}(\varphi_1), \ldots, i\mathcal{O}(\varphi_k))$ for independent choices of obfuscation randomness.

Fix any Φ^1 and Φ^2 with $\#\mathsf{SAT}(\Phi^1) = \#\mathsf{SAT}(\Phi^2)$ and let $w^1 \in \mathsf{SAT}(\Phi^1)$ and $w^2 \in \mathsf{SAT}(\Phi^2)$. Our goal is to show that

$$\Pr[C'(i\mathcal{O}(\Phi^1 \circ F)) = w^1] \approx_{2^{-10nk}} \Pr[C'(i\mathcal{O}(\Phi^2 \circ F)) = w^2].$$

As an intermediate step, we will show that for any $w' = (w'_1, \ldots, w'_k) \in (\{0,1\}^n)^k$ that

$$\Pr\left[C'(i\mathcal{O}(\Phi^1 \circ F)) = w' \text{ and } F(w') = w^1\right] \approx_{2^{-11nk}} \Pr\left[C'(i\mathcal{O}(\Phi^2 \circ F)) = w' \text{ and } F(w') = w^2\right]. \quad (3)$$

Assuming we can show (3), we are done since then

$$\begin{aligned}
&\Pr[C'(\Phi^{1'}) = w^1] \\
&= \sum_{w'} \Pr_{F \leftarrow PRF}[C'(i\mathcal{O}(\Phi^1 \circ F)) = w' \text{ and } F(w') = w^1] \\
&\approx_{2^{-10nk}} \sum_{w'} \Pr_{F \leftarrow PRF}[C'(i\mathcal{O}(\Phi^2 \circ F)) = w' \text{ and } F(w') = w^2] \\
&= \Pr[C'(\Phi^{2'}) = w^2].
\end{aligned}$$

Now we focus on (3). Fix such a w'. We go argue via a hybrid argument that goes input by input. Below, we often interpret x as an integer in the set $\{0, \ldots, 2^n - 1\}$

- **Hybrid 1**: Here the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$, defining $F_1, \ldots, F_k$.
  2) Output
     - The bit $\mathbb{1}[F_1(w'_1) = w^1_1] \wedge \cdots \wedge \mathbb{1}[F_k(w'_k) = w^1_k]$
     - $i\mathcal{O}(\Phi^1 \circ F)$.

- **Hybrid** $(2, i)$: Here the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$, defining $F_1, \ldots, F_k$.
  2) Let $b_j = \begin{cases} \mathbb{1}[w^2_j = F_j(w'_j)], & \text{if } w'_j < i \\ \mathbb{1}[w^1_j = F_j(w'_j)], & \text{if } w'_j \geq i \end{cases}$
  3) Output
     - The bit $b_1 \wedge \cdots \wedge b_k$
     - $i\mathcal{O}\left(x \mapsto \begin{cases} \Phi^2 \circ F, & \text{if } x < i \\ \Phi^1 \circ F, & \text{if } x \geq i \end{cases}\right)$.

- **Hybrid** $(3, i)$: Here the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$ to be keys punctured at the point $i+1$.
  2) Sample $v \leftarrow (\{0,1\}^n)^k$ and set $c = \Phi^1(v) \in \{0,1\}^k$. For all j, set $d_j = \mathbb{1}[w^1_j = v_j]$.
  3) Let $b_j = \begin{cases} \mathbb{1}[w^2_j = F_j(w'_j)], & \text{if } w'_j < i \\ d_j, & \text{if } w'_j = i+1 \\ \mathbb{1}[w^1_j = F_j(w'_j)], & \text{if } w'_j \geq i+2. \end{cases}$
  4) Output
     - The bit $b_1 \wedge \cdots \wedge b_k$
     - $i\mathcal{O}\left(x \mapsto \begin{cases} \Phi^2 \circ F, & \text{if } x < i \\ c, & \text{if } x = i+1 \\ \Phi^1 \circ F, & \text{if } x \geq i+2 \end{cases}\right)$.

- **Hybrid** $(4, i)$: Let $Ber(p)$ denote the Bernoulli distribution with probability p of being one. In this hybrid, the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$ to be keys punctured at the point $i+1$.
  2) Sample $c \leftarrow Ber(\frac{\#\mathsf{SAT}(\Phi^1)}{2^n}) \in \{0,1\}^k$. For all j, set $d_j \leftarrow c_j \wedge Ber(\frac{1}{\#\mathsf{SAT}(\varphi^1_j)})$.
  3) Let $b_j = \begin{cases} \mathbb{1}[w^2_j = F_j(w'_j)], & \text{if } w'_j < i \\ d_j, & \text{if } w'_j = i+1 \\ \mathbb{1}[w^1_j = F_j(w'_j)], & \text{if } w'_j \geq i+2. \end{cases}$
  4) Output
     - The bit $b_1 \wedge \cdots \wedge b_k$
     - $i\mathcal{O}\left(x \mapsto \begin{cases} \Phi^2 \circ F, & \text{if } x < i \\ c, & \text{if } x = i+1 \\ \Phi^1 \circ F, & \text{if } x \geq i+2 \end{cases}\right)$.

- **Hybrid (5,** i**)**: $i \in \{0, 2^n + 1\}$. Here the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$ to be keys punctured at the point $i+1$.
  2) Sample $v \leftarrow (\{0,1\}^n)^k$ and set $c = \Phi^1(v) \in \{0,1\}^k$. For all j, set $d_j = \mathbb{1}[w^2_j = v_j]$.
  3) Let $b_j = \begin{cases} \mathbb{1}[w^2_j = F_j(w'_j)], & \text{if } w'_j < i \\ d_j, & \text{if } w'_j = i+1 \\ \mathbb{1}[w^1_j = F_j(w'_j)], & \text{if } w'_j \geq i+2. \end{cases}$
  4) Output
     - The bit $b_1 \wedge \cdots \wedge b_k$
     - $i\mathcal{O}\left(x \mapsto \begin{cases} \Phi^2 \circ F, & \text{if } x < i \\ c, & \text{if } x = i+1 \\ \Phi^1 \circ F, & \text{if } x \geq i+2 \end{cases}\right)$.

- **Hybrid 6**: Here the distribution is

  1) Sample $\mathsf{sk}_1, \ldots, \mathsf{sk}_k$, defining $F_1, \ldots, F_k$.
  2) Output
     - The bit $\mathbb{1}[F_1(w'_1) = w^2_1] \wedge \cdots \wedge \mathbb{1}[F_k(w'_k) = w^2_k]$
     - $i\mathcal{O}(\Phi^2 \circ F)$.

We now show that the following pairs of hybrids is $2^{-\lambda^{\Omega(1)}}$ indistinguishable:

- Hybrid 1 to Hybrid $(2,0)$: This is by construction. The distributions are identical.
- Hybrid $(2,i)$ to $(3,i)$: This is by puncturing the PRFs $F_1,\ldots,F_k$ at $i+1$ and hard-coding uniformly random values v_j.
- Hybrid $(3,i)$ to $(4,i)$: This is because the output distribution only depends on v through the bits b and c, so it suffices to sample the bits from the distribution induced by b and c.
- Hybrid $(4,i)$ to $(5,i)$: This is because the new distribution on b and c is statistically identical to the previous distribution on b and c.
- Hybrid $(5,i)$ to $(2,i+1)$: This is by unpuncturing and replacing each v_j with $F_j(i+1)$.
- Hybrid $(2,2^{n+1})$ to Hybrid 6: This is by construction. The distributions are identical.

Hence, we get that Hybrid 1 and 6 are indistinguishable up to $O(2^n 2^{-\lambda^{\Omega(1)}}) < 2^{-11nk}$ by setting λ sufficiently large. (3) follows immediately from this indistiguishability. □

□

B. Refuting SAT Using an Oblivious k-fold SAT Solver

Now we show how to use an oblivious k-fold SAT solver to non-deterministically solve UNSAT.

Lemma VI.8. *Let $C : (\mathsf{SIZE}_n[s'])^k \to (\{0,1\}^n)^k$ be a probabilistic circuit. Assume that C is 2^{-10nk}-oblivious and hence there exists a function $p : ([2^n])^k \to \mathbb{R}$ such that for every all $\Phi \in (\mathsf{SIZE}_n[s'])^k$ and all $w \in \mathsf{SAT}(\Phi)$*

$$\frac{p(\#\mathsf{SAT}(\Phi))}{|\mathsf{SAT}(\Phi)|} - 2^{-10nk} \le \Pr_C[C(\Phi)=w] \le \frac{p(\#\mathsf{SAT}(\Phi))}{|\mathsf{SAT}(\Phi)|}. \tag{4}$$

There there is a non-deterministic circuit D for $\mathsf{UNSAT}[n, s-\mathrm{poly}(n)]$ with

$$|D| \le \varepsilon^{-1/k}\cdot|C|\cdot\mathrm{poly}(n,k,s),$$

where $\varepsilon = p(1,\ldots,1)$.

Proof. First, we note that p roughly gives the probability that C outputs an element of $\mathsf{SAT}(\Phi)$. (This is just unraveling definitions.)

Claim VI.9. *For all Φ,*

$$p(\#\mathsf{SAT}(\Phi)) - 2^{-9nk} \le \Pr_C[C(\Phi)\in\mathsf{SAT}(\Phi)] \le p(\#\mathsf{SAT}(\Phi)).$$

Consequently, we have $p(\#\mathsf{SAT}(\Phi)) \le 1.1$.

Proof. This immediately follows from (4) and the fact that $|\mathsf{SAT}(\Phi)| \le 2^{nk}$. □

Next, we introduce some more notation. For $i \in [2^n]$, let $p(i) = p(i,\ldots,i)$. Recall, we set $\varepsilon = p(1)$. We show there must be an i where $p(i)$ is not too much smaller than $p(i+1)$.

Claim VI.10. *There exists an $i \le 2\varepsilon^{-1/k}$ such that*

$$p(i) > (1+\frac{\varepsilon^{1/k}}{4})\frac{i^k}{(i+1)^k}\cdot p(i+1).$$

Proof. Suppose not. Then we have that

$$p(i) \le (1+\frac{\varepsilon^{1/k}}{4})\frac{i^k}{(i+1)^k}\cdot p(i+1)$$

for all $i \le 2\varepsilon^{-1/k}$. Rearranging, we get that

$$\frac{p(i+1)}{p(i)} \ge \frac{(i+1)^k}{i^k}(1+\frac{\varepsilon^{1/k}}{4})^{-1}.$$

Then for $i^\star = 2\varepsilon^{-1/k}$ we have

$$\begin{aligned}
p(i^\star) &= \varepsilon\cdot\prod_{i=1}^{i^\star-1}\frac{p(i+1)}{p(i)}\\
&\ge \varepsilon\cdot(1+\frac{\varepsilon^{1/k}}{4})^{-i^\star}\cdot\prod_{i=1}^{i^\star-1}\frac{(i+1)^k}{i^k}\\
&= \varepsilon\cdot(1+\frac{1}{2i^\star})^{-i^\star}\cdot(i^\star)^k\\
&= \varepsilon\cdot(1+\frac{1}{2i^\star})^{-i^\star}\cdot(2\varepsilon^{-1/k})^k\\
&= (1+\frac{1}{2i^\star})^{-i^\star}\cdot 2^k\\
&\ge \frac{3}{5}\cdot 2 = 1.2,
\end{aligned}$$

which contradicts that $p(i^\star)\le 1.1$. □

Indeed, we can extend Claim VI.10 to hold even with a 2^{-9nk} additive loss.

Claim VI.11. *There exists an $i \le 2\varepsilon^{-1/k}$ such that*

$$p(i) - 2^{-9nk} > (1+\frac{\varepsilon^{1/k}}{16})\frac{i^k}{(i+1)^k}p(i+1).$$

Proof. We have

$$\begin{aligned}
p(i) - 2^{-9nk} &> (1+\frac{\varepsilon^{1/k}}{4})\frac{i^k}{(i+1)^k}\cdot p(i+1) - 2^{-9nk}\\
&> (1+\frac{\varepsilon^{1/k}}{4} - 2^{-7nk})\frac{i^k}{(i+1)^k}\cdot p(i+1)\\
&> (1+\frac{\varepsilon^{1/k}}{8})\frac{i^k}{(i+1)^k}\cdot p(i+1)
\end{aligned}$$

where the each line comes from

- Claim VI.10
- Without loss of generality, $p(i+1) \ge 2^{-nk}$. (This is because we can modify C to also try a uniformly random guess $(w_1,\ldots,w_k)$ and output that if those are a satisfying assignment.) Thus,

$$\begin{aligned}
2^{-7nk}\cdot\frac{i^k}{(i+1)^k}\cdot p(i+1) &\ge 2^{-8nk}\cdot(\frac{i}{i+1})^k\\
&= 2^{-8nk}\cdot(1-\frac{1}{i+1})^k\\
&\ge 2^{-8nk}2^{-k}\\
&\ge 2^{-9nk}.
\end{aligned}$$

- Without loss of generality, $\varepsilon \geq 2^{-nk}$. (If not, the theorem is trivial.) Thus,

$$\frac{\varepsilon^{1/k}}{4} - 2^{-7nk} \leq \frac{\varepsilon^{1/k}}{4} - \frac{2^{-nk}}{8} \leq \frac{\varepsilon^{1/k}}{4} - \frac{\varepsilon}{8} \leq \frac{\varepsilon^{1/k}}{16},$$

where the last inequality comes from the following case analysis. If $\varepsilon \leq 1$, then we have

$$\frac{\varepsilon^{1/k}}{4} - \frac{\varepsilon}{8} \leq \frac{\varepsilon^{1/k}}{4} - \frac{\varepsilon^{1/k}}{8} \leq \frac{\varepsilon^{1/k}}{8}.$$

Otherwise we have $1 \leq \varepsilon = p(1) \leq 1.1$ so

$$\frac{\varepsilon^{1/k}}{4} - \frac{\varepsilon}{8} \leq \frac{1}{4} - \frac{1.1}{8} \leq \frac{1}{16} \leq \frac{\varepsilon^{1/k}}{16}.$$

□

Now we construct D. By Corollary IV.6, it suffices to solve Unique-UNSAT. Fix an $i \leq 2\varepsilon^{-1/k}$ with the guarentee from Claim VI.11.

Non-deterministic circuit D for Unique-UNSAT

Given an n-input s'-size formula ψ with at most one satisfying assignment:

1) Using brute force check if ψ is satisfiable on any x with $x < i$ (interpreting x as a non-negative integer using binary). If so, reject.
2) Otherwise, let ψ' be the formula given by $\psi'(x) = \psi(x) \vee \mathbb{1}[x < i]$. Note that $|\psi'| \leq s' + \mathrm{poly}(n) \leq s$ by setting $s' = s - \mathrm{poly}(n)$.
3) Output the output of the improved Goldwasser-Sipser non-deterministic circuit (Theorem III.2) for nmCAPP that accepts when $\alpha \geq (1 + \frac{\varepsilon^{1/k}}{16}) \cdot \frac{i^k}{(i+1)^k} \cdot p(i+1)$ and rejects if $\alpha \leq \frac{i^k}{(i+1)^k} \cdot p(i+1)$, where

$$\alpha \coloneqq \Pr_C[C(\psi', \ldots, \psi') \text{ outputs } (w_1, \ldots w_k) \text{ such that all } w_j \text{ satisfy } \psi' \text{ but none satisfy } \psi\ ].$$

By construction D has size

$$\mathrm{poly}(i, n, s) + \mathrm{poly}(n, s) + \varepsilon^{-1/k} \cdot |C| \cdot \mathrm{poly}(n, k, s) = \varepsilon^{-1/k} \cdot |C| \cdot \mathrm{poly}(n, k, s).$$

Now we show correctness. If ψ is unsatisfiable, then ψ' has exactly i satisfying assignments and none of them satisfy ψ. Thus, by Claim VI.11 and Claim VI.9, we get

$$\alpha \geq p(i) - 2^{-9nk} \geq (1 + \frac{\varepsilon^{1/k}}{16}) \cdot \frac{i^k}{(i+1)^k} \cdot p(i+1).$$

This gives us completeness.

On the other hand, if ψ has exactly one satisfying assignment w, then either $w < i$ and we reject, or the probability we want to lower bound in step 3 is, by (4), at most

$$\begin{aligned} &\sum_{(w_1,\ldots,w_k) \in [i]^k} \Pr_C[C(\varphi', \ldots, \varphi') = (w_1, \ldots, w_k)] \\ \leq &\sum_{(w_1,\ldots,w_k) \in [i]^k} \frac{p(i+1)}{(i+1)^k} \\ \leq &\frac{i^k}{(i+1)^k} \cdot p(i+1) \end{aligned}$$

so we reject. This gives soundness. □

C. Putting It Together

Now we complete the proof of Theorem VI.1.

Proof of Theorem VI.1. By Lemma VI.5, there is a circuit C' of size $T' = T + \mathrm{poly}(n, k, s)$ that is 2^{-10nk}-oblivious and ε'-successful on Search-Unique-k-fold-SAT$[n, s'']$, where $\varepsilon' \geq 4^{-k}\varepsilon - 2^{-10nk}$ and $s'' = (\frac{s'}{\mathrm{poly}(n,k)})^{\Omega(1)}$. Then by Lemma VI.8, there is a non-deterministic circuit D for UNSAT$[n, s]$ of size

$$(\varepsilon')^{-1/k} \cdot T' \cdot \mathrm{poly}(n, k, s) = \varepsilon^{-1/k} \cdot T \cdot \mathrm{poly}(n, k, s),$$

where we use $\varepsilon \geq 2^{-nk}$ (otherwise the theorem is trivial) and that $s'' - \mathrm{poly}(n) = (\frac{s'}{\mathrm{poly}(n,k)})^{\Omega(1)} - \mathrm{poly}(n) \geq s$ by setting $s' = \mathrm{poly}(s, n, k)$ sufficiently large. □

VII. Distributional CRHFs from $i\mathcal{O}$ and NP $\not\subset$ i.o.-coNP/poly

In this section, we prove the following theorem.

Theorem VII.1. *Assume subexponentially secure indistinguishably obfuscation exists, subexponentially secure puncture pseudorandom functions exist, and that* NP $\not\subseteq$ *i.o.-*coNP/poly*. Then a distributional collision resistant hash function family exists.*

We begin by recalling the definition of a distribution collision resistant hash function.

Definition VII.2 (dCRHFs, [35, 57])**.** Let $\mathcal{H} = \{H_n\}$ be a distribution on polynomial-sized circuits. We say $\mathcal{H}$ is a *distributional collision resistant hash function* if there exists a polynomial $p = p(n)$ such that the following holds. For every probabilistic polynomial-time algorithm A,

$$\Delta((h \leftarrow H_n, A(h, 1^n)), (h \leftarrow H_n, \mathsf{Simon}(h)) > \frac{1}{p(n)},$$

for all but finitely many n, where Δ denotes statistical distance and $\mathsf{Simon}(h)$ denotes the (inefficient) probabilistic algorithm that samples $x \leftarrow \{0,1\}^n$, samples $y \leftarrow h^{-1}(x)$ and outputs (x, y).

Our construction is very simple: it is (again) an obfuscated puncturable PRF.

Construction VII.3 (Candidate dCRHF $\mathcal{H}$)**.** Let $n, \lambda = \mathrm{poly}(n) \in \mathbb{N}$. λ is chosen so that the $i\mathcal{O}$ and puncturable PRF families are $(2^{O(n)}, 2^{-O(n)})$-secure.

The construction is the distribution $\mathcal{H} = \{H_n\}$ on circuits, where H_n is given by

- Sample $s \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$
- $F_s : \{0,1\}^n \to \{0,1\}^{n-1}$ is a subexponentially secure puncturable PRF with key s
- The circuit $H_n \leftarrow i\mathcal{O}(F_s)$, where $i\mathcal{O}$ is a subexponentially secure indistinguishability obfuscation scheme.

We now prove Theorem VII.1.

Proof. We will prove that the dCRHF property holds with $p(n) = n^3$. Let $\varepsilon = 1/p(n)$. For contradiction, assume that there is a polynomial time probabilistic adversary A and a constant c such that

$$\Delta((h, A(h, 1^n)), (h, \mathsf{Simon}(h))) > \varepsilon$$

for infinitely many n. We will show that $\mathsf{NP} \subseteq$ i.o.-coNP/poly. By Corollary IV.6, it suffices to give an polynomial-size refutation for Unique-UNSAT$[n, n]$.

Now we define some notation. For $y \in \{0,1\}^{n-1}$, $z \in \{0,1\}^n$, and an n-input circuit φ, we define

$$h_{\varphi,y,z} = i\mathcal{O}\left(x \mapsto \begin{cases} y, & \text{if } \varphi(x \oplus z) = 1, \\ F_s(x), & \text{otherwise} \end{cases}\right)$$

where $F_s : \{0,1\}^n \to \{0,1\}^{n-1}$ is the PRF from our construction.

We will be concerned with analyzing the probability p^*_φ given by

$$\Pr_{\substack{y \leftarrow \{0,1\}^m \\ z \leftarrow \{0,1\}^n \\ (x_0,x_1) \leftarrow A(h_{\varphi,y,z})}} \Big[h_{\varphi,y,z}(x_0) = h_{\varphi,y,z}(x_1) = y \text{ and } \varphi(x_0 \oplus z) = \varphi(x_1 \oplus z) = 0 \Big].$$

Claim VII.4. *Assume* $(\mathrm{poly}(n), 2^{-2n})$*-security of the* $i\mathcal{O}$ *scheme. Then, for all* $\varphi \in$ UNSAT*, we have that* $p_\varphi \geq (1-\varepsilon)2^{-m} - 2^{-2n}$.

Proof. To prove this claim, we define the additional quantity $\bar{p}_\varphi$ equal to

$$\Pr_{\substack{y \leftarrow \{0,1\}^m \\ z \leftarrow \{0,1\}^n \\ (x_0,x_1) \leftarrow A(h)}} \Big[h(x_0) = h(x_1) = y \text{ and } \varphi(x_0 \oplus z) = \varphi(x_1 \oplus z) = 0 \Big].$$

We observe that for all $\varphi \in$ UNSAT, $p^*_\varphi \geq \bar{p}_\varphi - 2^{-2n}$ by the security of the $i\mathcal{O}$ (as the condition $\varphi(x \oplus z) = 1$ is not satisfied by any input x).

We now also claim that $\bar{p}_\varphi \geq (1-\varepsilon) \cdot 2^{-m}$. To see this, we observe that:

- The probability that $A(h)$ outputs a collision such that $\varphi(x_0 \oplus z) = \varphi(x_1 \oplus z) = 0$ is at least $1-\varepsilon$ by assumption on A (the additional condition is always true because $\varphi \in$ UNSAT).
- The string y is independent of the view of A.

This completes the proof of Claim VII.4. □

Claim VII.5. *Assume the* $(2^{2n}, 2^{-2n})$*-security of the* $i\mathcal{O}$ *and puncturable PRF. Then, if* $\varphi \in$ Unique-SAT*, we have* $p^*_\varphi \leq 2^{-m} \cdot (1 - \Omega(1)) + O(2^{-2n})$.

Proof. We prove this by a hybrid argument.

Let w denote the unique satisfying assignment to φ. We define the following auxiliary probability p_φ:

$$\Pr_{\substack{z \leftarrow \{0,1\}^n \\ (x_0,x_1) \leftarrow A(h)}} \Big[h(x_0) = h(x_1) = h(w \oplus z) \text{ and } \tag{5}$$

$$w \oplus z \notin \{x_0, x_1\} \Big]. \tag{6}$$

We first observe that

$$p^*_\varphi \leq p_\varphi + O(2^{-2n})$$

by the following hybrid argument: define the circuit distribution

$$h_{x^*,y} = i\mathcal{O}\left(x \mapsto \begin{cases} y, & \text{if } x = x^*, \\ F_s(x), & \text{otherwise} \end{cases}\right)$$

and associated probability $p^{(1)}_\varphi$ given by

$$p^{(1)}_\varphi = \Pr_{\substack{y \leftarrow \{0,1\}^m \\ z \leftarrow \{0,1\}^n \\ (x_0,x_1) \leftarrow A(h_{w \oplus z, y})}} \begin{bmatrix} h_{w\oplus z,y}(x_0) = h_{w\oplus z,y}(x_1) \\ = h_{w\oplus z,y}(w \oplus z) \\ \text{and} \\ w \oplus z \notin \{x_0, x_1\} \end{bmatrix}$$

$$= \Pr_{\substack{y \leftarrow \{0,1\}^m \\ z \leftarrow \{0,1\}^n \\ (x_0,x_1) \leftarrow A(h_{w \oplus z, y})}} \begin{bmatrix} h_{w\oplus z,y}(x_0) = h_{w\oplus z,y}(x_1) \\ = y \\ \text{and} \\ w \oplus z \notin \{x_0, x_1\} \end{bmatrix}$$

We then see that

- $p^{(1)}_\varphi \leq p_\varphi + O(2^{-2n})$ by puncturing the PRF $F_s(\cdot)$ at the point $w \oplus z$.
- $p^*_\varphi \leq p^{(1)}_\varphi + O(2^{-2n})$ because:
  - The condition $w \oplus z \notin \{x_0, x_1\}$ is equivalent to the condition that $\varphi(x_0 \oplus z) = \varphi(x_1 \oplus z) = 0$, and
  - The functions computed by $h_{w\oplus z,y}$ and $h_{\varphi,y,z}$ are equal.

It remains to prove that $p_\varphi \leq 2^{-m}(1 - \Omega(1))$. We next note that in (6), the inputs $(x_0, x_1) \leftarrow A(h)$ are independent of the random string z. We analyze p_φ by defining the following two quantites:

- Define $\mathsf{load}_h(x_0) = \left|h^{-1}(h(x_0))\right|$.
- Define the event $\mathsf{Col}_h(x_0, x_1)$ to be the predicate checking that (x_0, x_1) forms a *non-trivial* collision in h: that is, $h(x_0) = h(x_1)$ and $x_0 \neq x_1$.

With this in mind, we see that for a fixed hash function h and fixed strings (x_0, x_1),

$$\Pr_{z \leftarrow \{0,1\}^n} \Big[h(x_0) = h(x_1) = h(w \oplus z) \text{ and } w \oplus z \notin \{x_0, x_1\} \Big]$$

is at most

$$\Pr_{z\leftarrow\{0,1\}^n}\Big[h(x_0)=h(w\oplus z) \text{ and } w\oplus z\notin\{x_0,x_1\}\Big]$$

which is equal to

$$\mathsf{Col}_h(x_0,x_1)\cdot\Big(2^{-n}\cdot(\mathsf{load}_h(x_0)-2)\Big)+$$

$$(1-\mathsf{Col}_h(x_0,x_1))\cdot\Big(2^{-n}\cdot(\mathsf{load}_h(x_0)-1)\Big).$$

This is true because if $\mathsf{Col}_h(x_0,x_1)=1$:

- The probability that $h(w\oplus z)$ is equal to $h(x_0)$ is $\frac{\mathsf{load}_h(x_0)}{2^n}$, and
- The conditional probability that $w\oplus z\notin\{x_0,x_1\}$ is $1-\frac{2}{\mathsf{load}_h(x_0)}$, because $w\oplus z$ is (conditionally) uniform over $h^{-1}(h(x_0))\supset\{x_0,x_1\}$.

On the other hand, if $\mathsf{Col}_h(x_0,x_1)=0$:

- The probability that $h(w\oplus z)$ is equal to $h(x_0)$ is $\frac{\mathsf{load}_h(x_0)}{2^n}$, and
- The conditional probability that $w\oplus z\notin\{x_0,x_1\}$ is $1-\frac{1}{\mathsf{load}_h(x_0)}$, because $w\oplus z$ is (conditionally) uniform over $h^{-1}(h(x_0))$, and in this case $|h^{-1}(h(x_0))\cap\{x_0,x_1\}|=1$.

Rewriting, we have

$$\Pr_{z\leftarrow\{0,1\}^n}\Big[h(x_0)=h(w\oplus z) \text{ and } w\oplus z\notin\{x_0,x_1\}\Big]$$
$$=2^{-n}\cdot\Big(\mathsf{load}_h(x_0)-1-\mathsf{Col}_h(x_0,x_1)\Big).$$

Thus, we conclude that p_φ is at most

$$2^{-n}\cdot\mathop{\mathbb{E}}_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)\Big]-2^{-n}-$$

$$2^{-n}\cdot\Pr_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{Col}_h(x_0,x_1)\Big].$$

By assumption on A, we have that

$$\Pr_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{Col}_h(x_0,x_1)\Big]\geq 1/2-\varepsilon,$$

because the event occurs with probability at least $1/2$ for the Simon distribution.

Finally, we claim that

$$2^{-n}\mathop{\mathbb{E}}_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)\Big]$$
$$\leq 2^{-m}+2^{-n}+4\cdot\varepsilon\cdot n^2\cdot 2^{-n}+O(2^{-2n}).$$

To see this, we upper bound $\mathop{\mathbb{E}}_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)\Big]$ by

$$\Pr_h\Big[h \text{ has an } 2n\text{-collision}\Big]+$$
$$\mathop{\mathbb{E}}_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)\wedge h \text{ has no } 2n\text{-collision}\Big].$$

The first summand is at most 2^{-2n} by the $(2^{2n},2^{-2n})$-security of $F_s(\cdot)$ and the fact that such a bound holds for random functions, in the setting setting $m=n-1$.

To bound the second summand, we write

$$\mathop{\mathbb{E}}_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)\wedge h \text{ has no } 2n\text{-collision}\Big]=$$

$$\sum_{i=1}^{2n-1}\Pr_{(x_0,x_1)\leftarrow A(h)}\Big[\mathsf{load}_h(x_0)=i\Big]\cdot i.$$

Since (h,x_0) is ε-close to (h,r) for a uniformly random r by assumption on A, the above is upper bounded by

$$4n^2\cdot\varepsilon+\sum_{i=1}^{2n-1}\Pr_{\substack{h\leftarrow H_n\\ r\leftarrow\{0,1\}^n}}\Big[\mathsf{load}_h(r)=i\Big]\cdot i$$
$$\leq 4n^2\cdot\varepsilon+4n^2\cdot 2^{-2n}+\mathop{\mathbb{E}}_{\substack{F:\{0,1\}^n\to\{0,1\}^m\\ r\leftarrow\{0,1\}^n}}\Big[\mathsf{load}_F(r)\Big],$$

where $F:\{0,1\}^n\to\{0,1\}^m$ is a uniformly random function, by the 2^{-2n}-security of the PRF.

Finally, we have that $2^{-n}\cdot\mathop{\mathbb{E}}_{\substack{F:\{0,1\}^n\to\{0,1\}^m\\ r\leftarrow\{0,1\}^n}}\Big[\mathsf{load}_F(r)\Big]$ is exactly the expected collision probability of a uniformly random F, which is equal to $2^{-n}+2^{-m}$.

To conclude, we have that

$$p_\varphi\leq 2^{-m}+2^{-n}+4\cdot\varepsilon\cdot n^2\cdot 2^{-n}-\frac{1}{2^n}(\frac{3}{2}-\varepsilon)+O(2^{-2n}) \tag{7}$$
$$=2^{-m}-\frac{1}{2}\cdot 2^{-n}+O(\varepsilon n^2\cdot 2^{-n}) \tag{8}$$
$$=2^{-m}(1-\Omega(1)). \tag{9}$$

This completes the proof of Claim VII.5. □

With the achieved gap between the UNSAT and SAT instances (under the promise), this implies the existence of an infinitely-often correct AM/poly protocol for Unique-UNSAT by Goldwasser-Sipser.

□

VIII. Worst-Case to Average-Case Reduction for Correlation Intractability

In this section, we prove a worst-case to average-case reduction for various forms of single-input correlation intractability [23], defined as follows.

We consider binary relations R that are subsets of $\bigcup_{\lambda\in\mathbb{N}}\{0,1\}^{n(\lambda)}\times\{0,1\}^{m(\lambda)}$ where $n(\lambda),m(\lambda)=\mathrm{poly}(\lambda)$. Throughout, n and m denote the functions implicitly associated with the relation R.

Definition VIII.1 (Single-input Correlation Intractability)**.** Let $R\subset\{0,1\}^*$ be a binary relation. A hash function family $\mathcal{H}=(\mathsf{Gen},\mathsf{Hash})$ with domain $\{0,1\}^{n(\lambda)}$ and range $\mathbb{Z}_p^{m(\lambda)}$ is defined to be R-correlation intractable (CI) if for all polynomial time adversaries $\mathcal{A}(k)$ given a hash key $k\leftarrow\mathsf{Gen}(1^\lambda)$, the probability that $\mathcal{A}(k)\to x$ such that $x\in\{0,1\}^{n(\lambda)}$ and $(x,H_k(x))\in R$ is negligible.

Of particular interest are hash families that are correlation-intractable for entire *classes* of relations. We focus on two specific cases of interest:

- The class $\mathcal{R}_{\mathrm{Sparse}}$ of all *sparse* relations, which are relations $R = \{R_\lambda \subset \{0,1\}^{n(\lambda)} \times \mathbb{Z}_p^{m(\lambda)}\}$ such that for all $x \in \{0,1\}^{n(\lambda)}$, the probability that a uniformly random Y satisfies $(x, Y) \in R$ is $\mathrm{negl}(\lambda)$.
- The class $\mathcal{R}_{\mathrm{poly}}$ of all sparse relations such that the computational task of deciding whether $(x, y) \in R$ has a $\mathrm{poly}(\lambda)$-time algorithm.

For both of these relation classes, we have very few candidate hash families that are plausibly CI for the entire class of relations [14, 24, 15], and they all rely on extremely strong assumptions such as optimally unbounded KDM-secure encryption and optimally circular-secure FHE.

In this section, we construct hash function families that are CI for either $\mathcal{R}_{\mathrm{Sparse}}$ or $\mathcal{R}_{\mathrm{poly}}$ under two assumptions: one standard cryptographic assumption (shift-hiding shiftable functions), and one *worst-case hardness assumption* about correlation-finding tasks (Definition VIII.4).

A. Shift-Hiding Shiftable Functions

The following preliminaries are due to [51] (although we remove unnecessary public parameters from the definition).

Definition VIII.2 (Shift-Hiding Shiftable Functions [58, 51])**.** Let $p = p(\lambda)$ be an efficiently computable function of λ. We define a family of shift-hiding shiftable functions with input space $\{0,1\}^{n(\lambda)}$ and output space $\mathbb{Z}_p^{m(\lambda)} \simeq \{0,1\}^{m(\lambda)\log p(\lambda)}$ for arbitrary polynomial functions $(n(\lambda), m(\lambda))$.

For a given class $\mathcal{C}$ of function ensembles $\mathcal{F} = \{f_\lambda : \{0,1\}^{n(\lambda)} \to \mathbb{Z}_p^{m(\lambda)}\}$, a shift-hiding shiftable function family $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval}, \mathsf{SEval})$ consists of four PPT algorithms:

- $\mathsf{Gen}(1^\lambda)$ outputs a master secret key msk.
- $\mathsf{Shift}(\mathsf{msk}, f)$ takes as input a secret key msk and a function $f \in \mathcal{F}$. It outputs a shifted key sk_f.
- $\mathsf{Eval}(\mathsf{msk}, x)$, given a secret key msk and input $x \in \{0,1\}^{n(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{m(\lambda)}$.
- $\mathsf{SEval}(\mathsf{sk}_f, x)$, given a shifted key sk_f and input $x \in \{0,1\}^{n(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{m(\lambda)}$.

We will sometimes use the notation $F_{\mathsf{sk}}(x)$ to mean either $\mathsf{Eval}(\mathsf{sk}, x)$ or $\mathsf{SEval}(\mathsf{sk}, x)$ when the context is clear.

We require that SHSF satisfies the following two properties:

- **Computational Correctness**: for every polynomial-time adversary A, there is a negligible function $\mathrm{negl}(\lambda)$ such that for any function $f \in \mathcal{C}$, given a shifted key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$ (for $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$), the probability that $A(\mathsf{sk}_f)$ finds an input $x \in \{0,1\}^{n(\lambda)}$ such that $\mathsf{Eval}(\mathsf{sk}_f, x) \neq \mathsf{Eval}(\mathsf{msk}, x) + f(x) \pmod p$ is $\mathrm{negl}(\lambda)$.
- **Shift Hiding**: for any polynomial-time adversary A, there is a negligible function $\mathrm{negl}(\lambda)$ such that for any pair of functions $f, g \in \mathcal{C}$, the distinguishing advantage achieved by A between sk_f and sk_g is $\mathrm{negl}(\lambda)$, where $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$, $\mathsf{sk}_g \leftarrow \mathsf{Shift}(\mathsf{msk}, g)$, and $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.

Lemma VIII.3 ([58, 51])**.** *Suppose one of the following two pairs of assumptions holds:*

- *Sub-exponentially secure $i\mathcal{O}$ and sub-exponentially secure OWFs exist, OR*
- *The learning with errors and 1D-R-SIS problems (see [51]) are computationally hard.*

Then, there exists an efficiently computable function $p(\lambda)$ such that for any polynomial functions $n(\lambda), m(\lambda), s(\lambda)$, there exists a family of SHSFs mapping $\{0,1\}^{n(\lambda)} \to \mathbb{Z}_p^{m(\lambda)}$ and supporting the class $\mathcal{C}_s$ of all circuits of size s. Specifically,

- *For the $i\mathcal{O}$-based construction, $p(\lambda)$ can be an arbitrary power of 2. (we will simply take $p(\lambda) = 2$ in this case). Moreover, the $i\mathcal{O}$-based construction can be made to have 2^{n+m}-security.*
- *For the lattice-based construction, $p(\lambda) \leq 2^{\lambda^\varepsilon}$ can be taken to be a product of $\lambda^{\varepsilon/2}$ sufficiently large distinct primes. However, this construction only has $(\mathrm{poly}, \mathrm{negl})$-security.*

B. The Correlation Finding Problem

Definition VIII.4 (Correlation Finding Problem)**.** Let R be a relation and $s : \mathbb{N} \to \mathbb{N}$ be a function. $\mathsf{Search}_R[s]$ is the following search problem:

- **Given:** a circuit $C : \{0,1\}^{n(\lambda)} \to \mathbb{Z}_{p(\lambda)}^{m(\lambda)}$ of size at most $s(\lambda)$
- **Output:** an x such that $(x, C(x)) \in R$, provided one exists.

We observe that $\mathsf{Search}_R[s]$ is not necessarily a total search problem: some instances C may have no valid output. In the worst-case setting, we declare by convention that any output is a valid solution, although it will turn out that we work with total $\mathsf{Search}_R[s]$.

We now consider three notions of hardness for $\mathsf{Search}_R[s]$: mild worst-case hardness, strong worst-case hardness, and average-case hardness.

Definition VIII.5. We say that $\mathsf{Search}_R[s]$ is *mildly* worst-case hard if no non-uniform polynomial-sized (deterministic) circuit family solves $\mathsf{Search}_R[s]$ for infinitely many λ.

Definition VIII.6. We say that $\mathsf{Search}_R[s]$ is *strongly* worst-case hard if for every polynomial-sized family of *probabilistic* circuits A, there are infinitely many λ and circuits C_λ of size $s(\lambda)$ such that

$$\Pr\left[A(C_\lambda) \text{ outputs an } x \text{ where } (x, C_\lambda(x)) \in R\right] = \lambda^{-\omega(1)}.$$

Definition VIII.7 (Average-case Hardness of Search_R)**.** Let $\mathcal{D} = \{D_\lambda\}$ be a distribution on circuits with $n(\lambda)$-length inputs and $m(\lambda)$-length outputs. We say that Search_R is *hard on* $\mathcal{D}$ if for every non-uniform polynomial-time adversary A

$$\Pr_{C \leftarrow D_\lambda}\left[A(C) \text{ outputs an } x \text{ where } (x, C(x)) \in R\right] = \lambda^{-\omega(1)}.$$

Remark: in the average-case setting, we no longer count the case where there is no x with $(x, C(x)) \in R$ as a "success" for the algorithm.

a) Worst-Case Assumption: Correlation Finding is Worst-Case Hard: For any class $\mathcal{R}$ of sparse relations, one can conjecture the worst-case hardness of $\mathsf{Search}_R[s]$ for all $R \in \mathcal{R}$ such that $\mathsf{Search}_R[s]$ is a total search problem. Indeed, one can conjecture both mild and strong worst-case hardness.

Conjecture 1 ($\mathcal{R}$-Correlation Finding is Mildly Worst-Case Hard)**.** *For all $m, n = \mathrm{poly}(\lambda)$ there is an $s = \mathrm{poly}(\lambda)$ such that $\mathsf{Search}_R[s]$ is mildly worst-case hard for all relations $R \in \mathcal{R}$ for which $\mathsf{Search}_R[s]$ is total.*

Conjecture 2 ($\mathcal{R}$-Correlation Finding is Strongly Worst-Case Hard)**.** *For all $m, n = \mathrm{poly}(\lambda)$ there is an $s = \mathrm{poly}(\lambda)$ such that $\mathsf{Search}_R[s]$ is strongly worst-case hard for all relations $R \in \mathcal{R}$ for which $\mathsf{Search}_R[s]$ is total.*

We briefly give some remarks on these conjectures.

- Conjecture 2 implies Conjecture 1.
- Conjecture 2 is implied by the existence of an efficiently computable hash family that is CI for all $R \in \mathcal{R}$.
- Worst-case hardness may hold for even more choices of R, but this suffices for our purposes. Moreover, some assumption must be made on R to rule out trivial counterexamples where Search_R is efficiently solvable: e.g., the empty relation, or relations that are "ultra sparse" in that they only have support on a polynomial number of x.
- The order of quantifiers is important: there needs to be a single $s = \mathrm{poly}(\lambda)$ for which $\mathsf{Search}_R[s]$ is hard for *all* applicable R.

C. The worst-case to average-case reduction: theorem statements

We prove two worst-case to average-case reductions; one is specific to the relation class $\mathcal{R}_{\mathrm{poly}}$, while one also holds for various larger relation classes.

Theorem VIII.8. *Let $n(\lambda), m(\lambda), p(\lambda)$ be efficiently computable functions of a security parameter.*

Assume that the following are both true:

- *For all $s(\lambda) = \mathrm{poly}(\lambda)$, a family of SHSFs with input length $n(\lambda)$ and output $\mathbb{Z}_p^{m(\lambda)}$ exists supporting $\mathcal{C}_s$.*
- *Conjecture 1 holds for the class $\mathcal{R}_{\mathrm{poly}}$.*

Then, there exists a hash family that is CI for all $R \in \mathcal{R}_{\mathrm{poly}}$.

Theorem VIII.9. *Let $\mathcal{B}$ be a class of algorithms that is closed under polynomial-time composition, and let $\mathcal{R}$ denote the corresponding class of* sparse *relations that can be decided by some algorithm in $\mathcal{B}$. Let $n(\lambda), m(\lambda), p(\lambda)$ be efficiently computable functions of a security parameter.*

Assume that the following are both true:

- *For all $s(\lambda) = \mathrm{poly}(\lambda)$, a family of 2^{n+m}-secure SHSFs with input length $n(\lambda)$ and output $\mathbb{Z}_p^{m(\lambda)}$ exists supporting $\mathcal{C}_s$.*
- *Conjecture 2 holds for the class $\mathcal{R}$.*

Then, there exists a hash family that is CI for all $R \in \mathcal{R}$.

In particular, this gives new feasibility results for $\mathcal{R}_{\mathrm{Sparse}}$- and $\mathcal{R}_{\mathrm{poly}}$- correlation intractability.

In fact, since the existence of any CI function implies Conjecture 2 we get that our construction is *universal* in the following sense.

Corollary VIII.10. *[Informal] Assume the existence of SHSFs as in Theorem VIII.8 or Theorem VIII.9. Then, if* any *function family is correlation intractable for the appropriate relation class $\mathcal{R}$, then our explicit construction (depending on the SHSF) is correlation intractable for $\mathcal{R}$.*

This universality result is an improvement over the universality result of [15], which assumed the existence of $\mathcal{R}_{\mathrm{Sparse}}$-CI and only concluded $\mathcal{R}_{\mathrm{poly}}$-CI of the universal construction. Thus, this corollary is new for both $\mathcal{R} = \mathcal{R}_{\mathrm{poly}}$ and $\mathcal{R} = \mathcal{R}_{\mathrm{Sparse}}$.

D. Proof of Theorems VIII.8 and VIII.9

Our construction is identical to that of [51], although our assumptions, conclusions, and security proof are different. Given a SHSF family $(\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval}, \mathsf{SEval})$, the construction is as follows.

Construction VIII.11 (Candidate Correlation Intractable Family $CI[n, m, s]$)**.** Let $n = \mathrm{poly}(\lambda)$, $m = \mathrm{poly}(\lambda)$, and $s = s(\lambda)$ be arbitrary parameters. The construction $CI[n, m, s]$ is a distribution $\mathcal{D} = \{D_\lambda\}$ on circuits, where D_λ is given by

- Sample a master secret key $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.
- Sample a shifted key $\mathsf{sk}_Z \leftarrow \mathsf{Shift}(\mathsf{msk}, Z)$ for an all-zero circuit Z.
- The circuit is given by $x \mapsto \mathsf{SEval}(\mathsf{sk}_Z, x)$. The hash key is sk_Z.

Theorem VIII.12. *Under the assumptions of Theorem VIII.8 or Theorem VIII.9, we have that for all $n, m = \mathrm{poly}(\lambda)$, there exists an $s = \mathrm{poly}(\lambda)$ such that $CI[n, m, s]$ is $\mathcal{R}$-correlation intractable for the $\mathcal{R}$ specified in the theorem statements.*

In fact, we prove a stronger statement about *specific* relations R (rather than classes of relations) that implies Theorem VIII.12.

Theorem VIII.13. *Under the* cryptographic *assumptions of Theorem VIII.8 (respectively, Theorem VIII.9), we have that for all $n, m = \mathrm{poly}(\lambda)$ and all $R \subset \mathcal{R} \cap \bigcup_\lambda \{0,1\}^{n(\lambda)} \times \mathbb{Z}_p^{m(\lambda)}$, if $CI[n, m, s]$ is* not *R-correlation intractable, then there exists a non-trivial relation $R' \subset \mathcal{R} \cap \bigcup_\lambda \{0,1\}^{n(\lambda)} \times \mathbb{Z}_p^{m(\lambda)}$ such that $\mathsf{Search}_{R'}[s]$ is not mildly (respectively, strongly) worst-case hard.*

Proof. The two main ideas are:

1) for any circuit C^* with the appropriate size and input/output length, an adversary for $CI[n, m, s]$ must still find R-correlations for $H_{\mathsf{sk}_{C^*}}$. This produces an R_{msk}-correlation for C^* with non-negligible probability, where R_{msk} is some msk-dependent relation.
2) The problem with Item 1 is that any given adversary may only succeed on an inverse polynomial fraction of msk, and the subset of "good msk" may depend on the adversary. However, a variant of Adleman's trick can

be used to construct a relation R' that is *not* adversary-dependent, on which the adversary violates correlation intractability.

Now, we give the formal proof. Assume that the hash family is not R-correlation intractable; this means that for some efficient adversary A, we have that

$$\Pr_{\substack{\mathsf{msk}\leftarrow\mathsf{Gen}(1^\lambda)\\ \mathsf{sk}_Z\leftarrow\mathsf{Shift}(\mathsf{msk},Z)}}\Big[A(\mathsf{sk}_Z) \text{ outputs an } x \text{ satisfying } (x, H_{\mathsf{sk}_Z}(x)) \in R\Big] \geq \frac{1}{\lambda^{O(1)}} \tag{10}$$

for infinitely many λ. Without loss of generality (by an averaging argument), we assume A is deterministic.

We now begin to build our worst-case algorithm for correlation-finding (although we have yet to specify R'). For a worst-case circuit $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$, let $\mathsf{sk}_{C^*} \leftarrow \mathsf{Shift}(\mathsf{msk}, C^*)$ for $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.

Claim VIII.14. *For infinitely many λ, we have that for all circuits $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$*

$$\Pr_{\mathsf{sk}_{C^*}}\Big[A(\mathsf{sk}_{C^*}) \textit{ outputs an } x \textit{ satisfying } (x, H_{\mathsf{sk}_{C^*}}(x)) \in R\Big] \geq \frac{1}{\lambda^{O(1)}} \tag{11}$$

This follows immediately from the shift-hiding property of the SHSF. (Either polynomial or exponential security is required depending on whether R is polynomial-time decidable.)

Claim VIII.15. *For infinitely many λ, we have that for all circuits $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$*

$$\Pr_{\mathsf{sk}_{C^*}}[A(\mathsf{sk}_{C^*}) \textit{ outputs an } x \textit{ satisfying } (x, F_{\mathsf{msk}}(x) + C^*(x) \pmod{p(\lambda)}) \in R] \geq \frac{1}{\lambda^{O(1)}} \tag{12}$$

This follows immediately from the computational correctness property of the SHSF. (Again, either polynomial or exponential security is required depending on whether R is polynomial-time decidable.)

The failure probability in (12) is $1 - \frac{1}{\lambda^{O(1)}}$. We will now repeat this sampling procedure independently polynomially many times (using fresh msk each time), so that the failure probability becomes exponentially small. In more detail, let $q = \mathrm{poly}(\lambda)$ be a sufficiently large polynomial. For a circuit $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$, consider sampling $\mathsf{sk}_{C^*,1}, \ldots, \mathsf{sk}_{C^*,q}$ *independent* shifted keys. We then get that for infinitely many λ, for all circuits $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$ that, we have

$$\Pr_{\mathsf{sk}_{C^*,1},\ldots,\mathsf{sk}_{C^*,q}}\Big[\text{for some } i \in [q],\ A(\mathsf{sk}_{C^*,i}) \to x \text{ with } (x, C^\star(x) + F_{\mathsf{msk}_i}(x) \pmod{p_\lambda}) \in R\Big] \geq 1 - O(2^{-s^2}).$$

Then, since the number of circuits $C^\star$ of size at most s is $2^{O(s \log s)} = o(2^{s^2})$, an averaging argument implies that for each input length λ we can fix a choice of $\mathsf{msk}_1, \ldots, \mathsf{msk}_q$ as well as sampling randomness for the $\mathsf{Shift}(\mathsf{msk}_i, \cdot)$ operations to obtain a non-uniform polynomial-time *deterministic* algorithm $D(C^*) \to (\mathsf{msk}_1, \mathsf{sk}_{C^*,1}, \mathsf{msk}_2, \mathsf{sk}_{C^*,2}, \ldots, \mathsf{msk}_q, \mathsf{sk}_{C^*,q})$ such that the following holds: for infinitely many λ, for all circuits $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$, there exists an i such that $A(\mathsf{sk}_{C^*,i})$ outputs an x with $(x, C^\star(x) + F_{\mathsf{msk}_i}(x) \pmod{p_\lambda}) \in R$.

This analysis gives us our desired relation $R' = R'_{\mathsf{msk}_1,\ldots,\mathsf{msk}_q}$:

$$R_{\mathsf{msk}_1,\ldots,\mathsf{msk}_q} = \{(x,y) : (x, y + F_{\mathsf{msk}_i}(x) \pmod{p_\lambda}) \in R \text{ for some } i \in [q(\lambda)]\}.$$

By construction, we have that R' is sparse: its sparsity is at most $q(\lambda)$ times that of R. Moreover, we claim that $R' \in \mathcal{R}$. To see this, we recall that $\mathcal{R}$ is the class of all relations decidable by algorithms in $\mathcal{B}$, and $\mathcal{B}$ is closed under polynomial composition. By construction, whether $(x,y) \in R'$ can be decided in polynomial time given oracle access to an algorithm deciding membership in R, so we conclude that $R' \in \mathcal{R}$ for all $R \in \mathcal{R}$.

Finally, we have to specify an algorithm solving $\mathsf{Search}_{R'}[s]$. The algorithm is as follows:

- Hard-coded: the keys $\mathsf{msk}_1, \ldots, \mathsf{msk}_q$, and randomness $r_1, \ldots, r_q$ for key generation.
- Input: a circuit C^* of size at most s.
- For all $i \in [q]$, construct the key $\mathsf{sk}_{C^*,i} = \mathsf{Shift}(\mathsf{msk}_i, C^*; r_i)$.
- For all $i \in [q]$, run $A(\mathsf{sk}_{C^*,i})$ to obtain input string x_i.
- **For $\mathcal{R}_{\mathrm{poly}}$:** for all i, check if $(x_i, C^*(x_i)) \in R'$ and output the first x_i for which this is true.
- **For general $\mathcal{R}$:** sample a random i and output x_i.

In both cases, our algorithm runs in polynomial time. Moreover, our earlier analysis shows that for infinitely many λ, for all circuits $C^\star : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ of size at most $s(\lambda)$ the algorithm will violate weak (respectively, strong) worst-case hardness[6] of $\mathsf{Search}_{R'}[s]$, as claimed. □

E. Open Problems

We end this section by stating two interesting open research directions related to Theorems VIII.8 and VIII.9.

- Are there interesting $\mathcal{R}$ for which one can base Conjecture 1 or Conjecture 2 on a standard complexity assumption? One possible avenue for this is to try to show *NP-hardness* of $\mathsf{Search}_R[s]$, or even to show *coNP-hardness* of $\mathsf{Search}_R[s]$ under non-deterministic reductions.
- Can Theorems VIII.8 and VIII.9 be extended to forms of *multi-input* correlation intractability [51]?

We hope that further work in these directions will result in better feasibility results for powerful forms of correlation-intractability.

[6]Observe that we have implicitly proved that $\mathsf{Search}_{R'}[s]$ is total, as our algorithm finds valid solutions on every circuit C^* of size at most s.

IX. Worst-Case to Average-Case Reduction for Lossy Functions

We begin by recalling the definition of a lossy function.

Definition IX.1 (Lossy Functions)**.** A (cryptographic) $\ell = \ell(n)$-lossy function L consists of a pair of probabilistic polynomial-time algorithms $(L.\mathsf{inj}, L.\mathsf{lossy})$ with the following properties:

- **Injective Functionality:**

$$\Pr_{C \leftarrow L.\mathsf{inj}(n)}[C \text{ is an } n\text{-input injective circuit}] \geq 1 - n^{-\omega(1)}.$$

- **Lossy Functionality:**

$$\Pr_{C \leftarrow L.\mathsf{lossy}(n)}\left[\begin{array}{r} C \text{ is an } n\text{-input circuit} \\ \text{whose range has size at most } 2^{\ell} \end{array}\right] \geq 1 - n^{-\omega(1)}$$

- **Indistinguishability:** $L.\mathsf{inj}(n) \approx_{n^{-\omega(1)}} L.\mathsf{lossy}(n)$.

Next, we describe the construction we will work with in this section. For a circuit C, let $\mathsf{Pad}_s(C)$ denote Padding C to size s if $|C| < s$ and the identity map otherwise.

Construction IX.2. For a function $\ell = \ell(n)$ and a polynomial $s = s(n) \geq n$ and a subexponentially-secure indistinguishability obfuscator $i\mathcal{O}$, we let $L = L_{\ell,s}$ be the following construction.

- $L.\mathsf{inj}(n) \to i\mathcal{O}(\mathsf{Pad}_s(F))$ where $F : \{0,1\}^n \to \{0,1\}^{4n}$ is a subexponentially-secure puncturable PRF and where we use security parameter $\lambda = s$ for both the $i\mathcal{O}$ and the puncturable PRF.
- $L.\mathsf{lossy}(n) \to i\mathcal{O}(\mathsf{Pad}_s(G \circ H))$ where $H : \{0,1\}^n \to \{0,1\}^{\ell}$ and $G : \{0,1\}^{\ell} \to \{0,1\}^{4n}$ are subexponentially-secure puncturable PRFs and where we use security parameter $\lambda = s$ for both the $i\mathcal{O}$ and the puncturable PRFs.

It is easy to see that this construction has injective functionality (because a random function length quadrupling function is injective with very high probability) and lossy functionality (by construction). Formally, we have the following proposition.

Proposition IX.3. *The lossy function L in Construction IX.2 with parameters ℓ and s has both the injective functionality and the lossy functionality properties (as defined in Definition IX.1) of being an ℓ-lossy function .*

Proof. Lossy functionality is by construction. In more detail, $L.\mathsf{lossy}(n) \to i\mathcal{O}(\mathsf{Pad}_s(G \circ H))$, where $H : \{0,1\}^n \to \{0,1\}^{\ell}$. Hence, $L.\mathsf{lossy}(n)$ always outputs a circuit with a range of size at most 2^{ℓ}.

Next, we show injective functionality. Recall $L.\mathsf{inj}(n) \to i\mathcal{O}(\mathsf{Pad}_s(F))$ where $F : \{0,1\}^n \to \{0,1\}^{4n}$ is a subexponentially-secure puncturable PRF. If F were a truly random function, then it is injective with probability at least $1 - 2^{-4n}2^{2n} = 1 - 2^{-2n}$, by union bounding over the event that two inputs collide. Hence, by the subexponential security of F, we get that injective functionality holds with probability at least $2^{-2n} + 2^{-n^{\Omega(1)}} = n^{-\omega(1)}$, as desired. □

Thus, to show that Construction IX.2 is a lossy function, we just need to show the indistinguishability property (i.e., $L.\mathsf{inj}(n) \approx_{n^{-\omega(1)}} L.\mathsf{lossy}(n)$). We will show that it holds under the assumption that the Lossy problem (defined next) is worst-case hard. Roughly, Lossy asks one to determine if a given circuit is injective or has a small range.

Definition IX.4 (Lossy Problem)**.** Let $\ell = \ell(n) < n$ and $s = s(n)$ be functions of n. Then $\mathsf{Lossy}[\ell, s]$ is the following problem:

- **Given:** an n-input circuit C of size at most s
- **Accept:** if C is injective
- **Reject:** if the range of C has size at most 2^{ℓ}

. One can easily solve $\mathsf{Lossy}[\ell, s]$ in $\mathrm{poly}(2^{\ell}, s)$ time.

Proposition IX.5. *For every ℓ and s, there is a circuit for $\mathsf{Lossy}[\ell, s]$ of size $\mathrm{poly}(2^{\ell}, s)$.*

Proof. Consider the following algorithm: evaluate the given C on $2^{\ell(n)} + 1$ different inputs, and look at whether all the outputs are pairwise distinct. If C is injective, then all the outputs will be pairwise distinct. If C has a range of size at most 2^{ℓ}, then two outputs must be the same (by the pigeonhole principle). □

We are now ready to prove the main theorem of this section: that Construction IX.2 is indeed a lossy function if Lossy is worst-case hard.

Theorem IX.6 (Lossy Functions from $i\mathcal{O}$ and Worst-Case Hardness)**.** *If $\mathsf{Lossy}[\ell, s']$ is not in $\mathsf{P/poly}$ infinitely often for some polynomial $s' = \mathrm{poly}(n)$, then, for some $s = \mathrm{poly}(n)$, Construction IX.2 with parameters 4ℓ and s is a 4ℓ-lossy function.*

Proof. Let s be a sufficiently large polynomial we set later. Let L denote Construction IX.2 with parameters 4ℓ and s. We already showed injective and lossy functionality in Proposition IX.3. So, we just need to show indistinguishability.

We do this by the contrapositive. Let $C : \{0,1\}^n \to \{0,1\}^m$ be an arbitrary n-input circuit of size at most s'. Let F, G, H, G', H' be subexponentially secure puncturable PRFs with security parameter $\lambda = s$ and with $F : \{0,1\}^n \to \{0,1\}^{4n}$, $G : \{0,1\}^{4\ell} \to \{0,1\}^{4n}$, $H : \{0,1\}^n \to \{0,1\}^{4\ell}$, $H' : \{0,1\}^m \to \{0,1\}^{4\ell}$ and $F' : \{0,1\}^m \to \{0,1\}^{4n}$. Recall that $L.\mathsf{inj} \equiv i\mathcal{O} \circ \mathsf{Pad}_s(F)$ and $L.\mathsf{lossy} \equiv i\mathcal{O} \circ \mathsf{Pad}_s(G \circ H)$.

Define $L_C.\mathsf{inj} \equiv i\mathcal{O}(\mathsf{Pad}_s(F' \circ C))$ and $L_C.\mathsf{lossy} \equiv i\mathcal{O}(\mathsf{Pad}_s(G \circ H' \circ C))$. The reason for our naming of $L_C.\mathsf{inj}$ and $L_C.\mathsf{lossy}$ is because we will show that the following claim holds.

Claim IX.7. *Assume C is injective. Then $L_C.\mathsf{inj} \approx_{2^{-s^{\Omega(1)}}} L.\mathsf{inj}$ and $L_C.\mathsf{lossy} \approx_{2^{-s^{\Omega(1)}}} L.\mathsf{lossy}$.*

On the other hand, when C is lossy, we will show that $L_C.\mathsf{lossy}$ and $L_C.\mathsf{inj}$ are indistinguishable.

Claim IX.8. *Assume C has a range of size at most 2^{ℓ}. Then $L_C.\mathsf{inj} \approx_{n^{-\omega(1)}} L_C.\mathsf{lossy}$. $L_C.\mathsf{inj} \approx_{2^{-n^{\Omega(1)}} + 2^{-2\ell}} L_C.\mathsf{lossy}$.*

Together, the claims mean that if an adversary A distinguishes $L.\mathsf{lossy}$ from $L.\mathsf{inj}$ with polynomial advantage, then one can use A to solve $\mathsf{Lossy}[\ell, s']$. This is because if C is injective, A distinguishes $L_C.\mathsf{lossy}$ from $L_C.\mathsf{inj}$ (by Claim IX.7), but, when C is lossy, A does not distinguish $L_C.\mathsf{lossy}$ from $L_C.\mathsf{inj}$ (by Claim IX.8). Thus, we can use A to solve $\mathsf{Lossy}[\ell, s']$.

It remains to show the two claims. First, we prove Claim IX.7.

Proof of Claim IX.7. First we show that $L_C.\mathsf{inj} \approx_{2^{-s^{\Omega(1)}}} L.\mathsf{inj}$. We do this by an input-by-input puncturing argument. Define

$$E^i(x) = i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right),$$

where we interpret $C(x) \in [2^{s'}]$ in the natural way (note that C has size at most s' so it has at most s' output bits). Our goal is to show that $L_C.\mathsf{inj} \equiv E^0$ is computationally indistinguishable from $E^{2^{s'}} \equiv L_C.\mathsf{inj}$. We do this by showing that E^i is computationally indistinguishable from E^{i+1} for an arbitrary i.

Indeed, letting $r \leftarrow \{0,1\}^{4n}$, we have that

$$\begin{aligned}
&E^i(x) \\
\equiv &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\approx_{2^{-s^{\Omega(1)}}} &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i \\ F' \circ C(x), & \text{if } C(x) = i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\approx_{2^{-s^{\Omega(1)}}} &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i \\ r, & \text{if } C(x) = i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\approx_{2^{-s^{\Omega(1)}}} &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\equiv &E^{i+1}(x).
\end{aligned}$$

The first line is by definition; the second is by functional equivalence (and setting s sufficiently large for padding); the third is by the puncturable PRF security of F' (and setting s sufficiently large for padding); the fourth is by the puncturable PRF security of F and C being injective (and setting s sufficiently large for padding); the last is by definition.

Furthermore, note that if $\{i+1, i+2, \ldots, i+k\}$ does not intersect the range of C, we have that

$$\begin{aligned}
&E^i(x) \\
\equiv &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\approx_{2^{-s^{\Omega(1)}}} &i\mathcal{O} \circ \mathsf{Pad}_s \left(x \mapsto \begin{cases} F(x), & \text{if } C(x) \le i+k \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right) \\
\equiv &E^{i+k}
\end{aligned}$$

by functional equivalence (and setting s sufficiently large for padding). Hence, since there are at most 2^n strings in the range of C, we get that $L_C.\mathsf{inj} \approx_\varepsilon L.\mathsf{inj}$, where $\varepsilon = 2^{-s^{\Omega(1)}+n} = 2^{-s^{\Omega(1)}}$ by setting s to be a sufficiently large polynomial. This completes our proof that $L_C.\mathsf{inj} \approx_{2^{-s^{\Omega(1)}}} L.\mathsf{inj}$.

The proof that $L_C.\mathsf{lossy} \approx_{2^{-s^{\Omega(1)}}} L.\mathsf{lossy}$ is similar. □

Next, we establish a claim that will be helpful in the proof of Claim IX.8.

Claim IX.9. *If C has a range of size at most 2^ℓ, then*

$$\Pr_{H'}[H' \textit{is injective on the range of } C] \ge 1 - 2^{-2\ell} - 2^{-s^{\Omega(1)}}.$$

Proof. If H' were a truly random function, then the probability that H' is not injective on the range of C is at most $2^{2\ell} 2^{-4\ell} = 2^{-2\ell}$, by a union bound argument. The claim then follows by the subexponential security of the PRF. □

Finally, we prove Claim IX.8. Actually, we will prove a more general version of Claim IX.8, which will be useful in the next subsection. After we prove the more general version, we will show that it indeed does imply Claim IX.8.

Claim IX.10. *Assume C has a range of size at most 2^ℓ. Then $L_C.\mathsf{inj} \approx_{2^{-n^{\Omega(1)}} + 2^{-2\ell}} L_C.\mathsf{lossy}$.*

Proof. We do this by an input-by-input puncturing argument. Define

$$E^i(x) = i\mathcal{O} \left(x \mapsto \begin{cases} G \circ H' \circ C(x), & \text{if } H' \circ C(x) \le i \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right),$$

where we interpret $H' \circ C(x) \in [2^{4\ell}]$ in the natural way. Our goal is to show that $E^0 \equiv L_C.\mathsf{inj}$ is computationally indistinguishable from $E^{2^{4\ell}} \equiv L_C.\mathsf{lossy}$. We do this by arguing that E^i and E^{i+1} are computationally indistinguishable for an arbitrary i.

Indeed, letting $r \leftarrow \{0,1\}^{4n}$, we have that E^i is $\equiv$ to

$$i\mathcal{O} \left(x \mapsto \begin{cases} G \circ H' \circ C(x), & \text{if } H' \circ C(x) \le i \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right)$$

is $\approx_{2^{-s^{\Omega(1)}}}$ to

$$i\mathcal{O} \left(x \mapsto \begin{cases} G \circ H' \circ C(x), & \text{if } H' \circ C(x) \le i \\ F'(C(x)), & \text{if } H' \circ C(x) = i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right)$$

is $\approx_{2^{-s^{\Omega(1)}} + 2^{-2\ell}}$ to

$$i\mathcal{O} \left(x \mapsto \begin{cases} G \circ H' \circ C(x), & \text{if } H' \circ C(x) \le i \\ r, & \text{if } H' \circ C(x) = i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right)$$

is $\approx_{2^{-s^{\Omega(1)}}}$ to

$$i\mathcal{O} \left(x \mapsto \begin{cases} G \circ H' \circ C(x), & \text{if } H' \circ C(x) \le i+1 \\ F' \circ C(x), & \text{otherwise.} \end{cases} \right)$$

is $\equiv$ to E^{i+1}, where the first is by definition, the second is by functional equivalence (and setting s sufficiently large), the third is by puncturable PRF security of F' combined with the fact that H' is injective on the range of C (Claim IX.9), the fourth is by puncturable PRF security of G, and the last is by definition. Putting this all together, we get that $L_C.\mathsf{inj} \approx_{2^{-s^{\Omega(1)}}+2^{-2\ell}} L_C.\mathsf{lossy}$. □

Finally, we need to show that Claim IX.10 implies Claim IX.8. Because $\mathsf{Lossy}[\ell, s']$ is not in $\mathsf{P/poly}$ infinitely often and because of the trivial upper bound (Proposition IX.5) on $\mathsf{Lossy}[\ell, s']$, it must be that $\ell = \omega(\log n)$. Thus, $2^{-2\ell} = n^{-\omega(1)}$. Plugging this into Claim IX.10 and setting s sufficiently large, we get that $L_C.\mathsf{inj} \approx_{n^{-\omega(1)}} L_C.\mathsf{lossy}$, which proves Claim IX.8.

Having proved Claim IX.7 and Claim IX.8, we have completed the proof of Theorem IX.6. □

In light of Theorem IX.6, it would be nice to connect the (worst-case) hardness of Lossy to the (worst-case) hardness of some standard complexity class. Intruigingly, Lossy does seem somewhat related to the following problem, which is complete for NISZK (the complexity class for non-interactive statistical zero-knowledge).

Theorem IX.11 ([37] [38, Lemma 13])**.** *There is an $\varepsilon > 0$ such that the following problem is complete for* NISZK*:*

- Given: *a circuit $C : \{0,1\}^m \to \{0,1\}^n$*
- Accept: *if C samples a distribution with entropy at least $n-2$*
- Reject: *if the range of C has size at most 2^{n-n^ε}*

We leave connections between NISZK and Lossy (or variants of either) as an interesting direction for future work. One concrete direction is to study variants of NISZK with perfect completeness.

A. ELFs from worst-case extreme lossiness

Our techniques for lossy function also extend to extremely lossy functions (ELFs).

Definition IX.12 (Extremely Lossy Functions [16])**.** An extremely lossy function (ELF) L consists of a pair of probabilistic polynomial-time algorithms $(L.\mathsf{inj}, L.\mathsf{lossy})$ and a polynomial $p(\cdot,\cdot)$ with the following properties:

- **Injective Functionality:**

$$\Pr_{C \leftarrow L.\mathsf{inj}(n)}[C \text{ is an } n\text{-input injective circuit}] \geq 1 - n^{-\omega(1)}.$$

- **Lossy Functionality:**

$$\Pr_{C \leftarrow L.\mathsf{lossy}(n,T,\varepsilon)}\begin{bmatrix} C \text{ is an } n\text{-input circuit} \\ \text{whose range has size} \\ \text{at most } p(T, 1/\varepsilon) \end{bmatrix} \geq 1 - n^{-\omega(1)}.$$

- **Indistinguishability:** For all polynomial functions $T(n,\lambda), \varepsilon(n,\lambda)$, the injective and (T,ε)-lossy modes are ε-indistinguishable to time T adversaries:

$$L.\mathsf{inj}(n) \approx_{T,\varepsilon} L.\mathsf{lossy}(n,T,\varepsilon).$$

In particular, we observe that Claim IX.7 and Claim IX.10 can also be used to prove a worst-case to average-case reduction for ELFs. We make use of the following worst-case computational hardness assumption.

Assumption IX.13 (Worst-case extreme lossiness)**.** There exists polynomials $s(\cdot)$ and $p(\cdot,\cdot)$ such that for every polynomial $T(\cdot)$ and every T-time randomized algorithm A that is given an n-input s-size circuit C of size s and outputs a bit b, there exists a pair of n-input s-size circuits $(C_{\mathsf{inj}}, C_{\mathsf{lossy}})$ with the following properties:

- C_{inj} is injective.
- C_{lossy} has a range of size at most $p(n, T(n))$, and
- $\mathbb{E}[A(C_{\mathsf{inj}})] \approx_{1/T} \mathbb{E}[A(C_{\mathsf{lossy}})]$.

Using this assumption, we get the following theorem.

Theorem IX.14. *Assuming sub-exponentially secure $i\mathcal{O}$, sub-exponentially secure OWFs, and Assumption IX.13, there exist ELFs.*

Proof. The proof is similar to the proof of Theorem IX.6. In detail, let s' and p be the polynomials for which Assumption IX.13 holds. As before, let s be a sufficiently large polynomial we choose later. Our ELF construction is as follows. $L.\mathsf{inj}$ is the same as the $L.\mathsf{inj}$ in Construction IX.2 with parameter s. We construct $L.\mathsf{lossy}$ so that $L.\mathsf{lossy}(n,T,\varepsilon)$ is the same as $L.\mathsf{lossy}(n)$ from Construction IX.2 but with parameter settings s and $\ell = p(T, 1/\varepsilon)$. This construction satisfies the injective and lossy functionality parts of the ELF definition by the same arguments as before.

It remains to show the indistinguishability part of the ELF definition holds for all polynomials T and ε. To do so, we fix an arbitrary T and ε. Now that T and ε have been fixed, observe that $L.\mathsf{lossy}(n,T,\varepsilon)$ is exactly the old $L.\mathsf{lossy}(n)$ from Construction IX.2 with parameter settings s and $\ell = p(T, 1/\varepsilon)$. This means we can reuse our analysis from before. In particular, Claim IX.7 and Claim IX.10 both hold. Finally, similar to before, Claim IX.7 and Claim IX.10 imply that an attack on ELF security can be used as a distinguisher for the worst-case extreme lossiness problem. □

X. Proofs of Quantumness from the White-Box Simon Problem

We consider search and decision variants of the white-box Simon problem.

Definition X.1 (White-Box Simon Problem)**.** Let $m, t : \mathbb{N} \to \mathbb{N}$ denote a efficiently computable unary functions. We define the search promise problem $\mathsf{Search\text{-}Simon} = \mathsf{Search\text{-}Simon}_{m(\cdot),t(\cdot)}$ as follows:

- Input: a circuit $C : \{0,1\}^n \to \{0,1\}^{m(n)}$ of size $t(n)$.
- Promise: there exists a nonzero string $s \in \{0,1\}^n$ such that $C(x) = C(y)$ if and only if $y \in \{x, x \oplus s\}$.
- Output: the (unique) string s guaranteed to exist by the promise.

We define the *decisional* promise problem $\mathsf{Simon} = \mathsf{Simon}_{m(\cdot),t(\cdot)}$ as follows:

- Input: a circuit $C : \{0,1\}^n \to \{0,1\}^m$ of size t.
- Promise: there exists a string $s \in \{0,1\}^n$ such that $C(x) = C(y)$ if and only if $y \in \{x, x \oplus s\}$.
- Output: 0 if $s = 0^n$ and 1 otherwise.

We prove the following worst-case to average-case reductions for Search-Simon and Simon.

Theorem X.2. *Assume the existence of sub-exponentially secure $i\mathcal{O}$ and sub-exponentially secure one-way functions. Then, there is a polynomial $p(n)$ such that if* $\mathsf{Search\text{-}Simon}_{m(\cdot),t(\cdot)}$ *(respectively,* $\mathsf{Simon}_{m(\cdot),t(\cdot)}$*) is hard for* BPP *or* P/poly *in the worst case, there is an efficiently sampleable distribution of inputs on which* $\mathsf{Search\text{-}Simon}_{3n,t(n)\cdot p(n)}$ *(respectively,* $\mathsf{Simon}_{3n,t(n)\cdot p(n)}$*) is hard on average for* BPP *or* P/poly.

As a corollary, this implies a non-interactive proof of quantumness assuming sub-exponential $i\mathcal{O}$, sub-exponential OWFs, and the worst-case hardness of Search-Simon:

- The verifier samples a circuit C from the specified distribution and sends C to the prover.
- The honest quantum prover runs Simon's algorithm to obtain s.
- The verifier checks that $C(s) = C(0^n)$.

Notably, the $i\mathcal{O}$ and OWF assumptions are agnostic to the adversary's computational model; the worst-case hardness of Search-Simon is the only "quantum advantage assumption" made in our protocol.

Proof of Theorem X.2. Let $i\mathcal{O}$ denote a 2^{-n}-secure indistinguishability obfuscator, and let $\{F_{\mathsf{sk}} : \{0,1\}^n \to \{0,1\}^{3n}\}$ denote a 2^{-4n}-secure puncturable PRF family. We then define the circuit family

$$C^*_{s,\mathsf{sk}}(x) = F_{\mathsf{sk}}(\min(x, s \oplus x))$$

where $\min$ is the efficient operation choosing the lexicographically first string out of an input pair. The circuit $C^*_{s,\mathsf{sk}}$ will be padded to a sufficiently large $p(n) \cdot t(n)$ size to facilitate the below security proof. Our hard Search-Simon distribution is given by

$$\widetilde{C}^* \leftarrow i\mathcal{O}(C^*_{s,\mathsf{sk}})$$

for uniformly random $s \leftarrow \{0,1\}^n \setminus \{0^n\}$ and sk sampled according to the PRF scheme.

We claim that if an efficient adversary $A(\widetilde{C}^*)$ outputs s with non-negligible probability, then Search-Simon has a BPP (or P/Poly, if A is non-uniform) algorithm. To prove this, let $C : \{0,1\}^n \to \{0,1\}^m$ denote an arbitrary circuit satisfying the Search-Simon promise. Let $M : \{0,1\}^n \to \{0,1\}^n$ denote a random full-rank matrix in $\mathbb{Z}_2^{n \times n}$. Let $F'_{\mathsf{sk}'} : \{0,1\}^m \to \{0,1\}^{3n}$ denote another 2^{-4n}-secure puncturable PRF family, and define the following distribution on obfuscated circuits:

$$\widetilde{C} \leftarrow i\mathcal{O}(C_{\mathsf{sk}',M})$$

for

$$C_{\mathsf{sk}',M} = F'_{\mathsf{sk}'} \circ C \circ M$$

We are now ready to state the main claim.

Claim X.3. $\widetilde{C}^* \approx_c \widetilde{C}$.

To prove the claim, we first observe that for every fixed circuit C, $C \circ M$ satisfies the Search-Simon promise with respect to a *uniformly random* $s \leftarrow \{0,1\}^n \setminus \{0^n\}$. We now claim that for every matrix M such that $C \circ M$ has period s,

$$i\mathcal{O}(C^*_{\mathsf{sk},s}) \approx_c i\mathcal{O}(C_{\mathsf{sk}',M})$$

for uniformly sampled $\mathsf{sk}, \mathsf{sk}'$. To prove this, we consider the following intermediate circuits

$$C_{\mathsf{sk},\mathsf{sk}',s,M,i}(x) = \begin{cases} F_{\mathsf{sk}}(\min(x, x \oplus s)), & \text{if } \min(x, x \oplus s) \le i \\ F_{\mathsf{sk}'} \circ C \circ M(x), & \text{otherwise.} \end{cases}$$

as well as the following hybrid distributions over obfuscated circuits:

$$\widetilde{C}_i \leftarrow i\mathcal{O}(C_{\mathsf{sk},\mathsf{sk}',s,M,i})$$

for uniformly random $\mathsf{sk}, \mathsf{sk}'$. The indistinguishability $\widetilde{C}_i \approx_c \widetilde{C}_{i+1}$ follows by a standard puncturing argument. Thus, if the $i\mathcal{O}$ and puncturable PRFs are $2^{-n} \cdot \mathrm{negl}(n)$-secure, the claim follows.

As a result, if $A(\widetilde{C}^*)$ outputs s such that $\widetilde{C}^*(s) = \widetilde{C}^*(0)$ with non-negligible probability, then the same must be true for $\widetilde{C}$. Now, let s_C denote the Search-Simon solution for C. By the security of $F_{\mathsf{sk}'}$, we have that with overwhelming probability, $\widetilde{C}(0)$ only collides with $\widetilde{C}(s)$ for $s = M^{-1} \cdot s_C$. Thus, we have obtained a polynomial-time algorithm for Search-Simon: run $A(\widetilde{C}) \to s$ and output $M \cdot s$.

For the decision problem Simon, the argument is almost identical: the worst-case hardness of Simon implies that $\widetilde{C}^*$ is computationally indistinguishable from the distribution $i\mathcal{O}(x \mapsto F_{\mathsf{sk}}(x))$, which is injective with overwhelming probability. □

A. Generalized Simon Problem

One benefit of the worst-case to average-case reduction in Theorem X.2 is that it allows for composition with simple worst-case reductions between different forms of Simon's problem. As a result, we obtain proofs of quantumness making use of an even weaker worst-case assumption.

Definition X.4 (Generalized White-Box Simon Problem)**.** Let $m, t : \mathbb{N} \to \mathbb{N}$ denote a efficiently computable unary functions. We define the search promise problem Search-GenSimon $=$ $\mathsf{Search\text{-}GenSimon}_{m(\cdot),t(\cdot)}$ as follows:

- Input: a circuit $C : \{0,1\}^n \to \{0,1\}^{m(n)}$ of size $t(n)$.
- Promise: there exists a subgroup $\mathcal{L} \subset \mathbb{Z}_2^n$ such that $C(x) = C(y)$ if and only if $y - x \in \mathcal{L}$.
- Output: any nonzero string $s \in \mathcal{L}$ (equivalently, a nonzero s such that $C(s) = C(0)$).

We observe that in the worst case, the generalized Simon problem is no harder than Search-Simon.

Claim X.5. *There is a polynomial-time randomized reduction from* Search-GenSimon *to* Search-Simon*.*

Proof. Let A be an algorithm for Search-Simon, and let C be an arbitrary circuit satisfying the promise of Search-GenSimon. To find a period of C, one can do the following:

- sample a uniformly random index $i \leq n$,
- sample a uniformly random full-rank matrix $M \leftarrow \mathbb{Z}_2^{n\times(n-i+1)}$,
- run $A(C \circ M) \to s$ on the circuit $C \circ M$,
- output $M \cdot s$ if $s \neq 0$ and $C(M \cdot s) = C(0)$, otherwise repeat.

This algorithm works (and only has polynomial overhead over A) because the index i will be equal to the dimension of $\mathcal{L} = \mathcal{L}_C$ with probability $1/n$, and conditioned on this event, the circuit $C \circ M$ will satisfy the promise of Search-Simon with constant probability by the following analysis:

- $C \circ M$ always satisfies the Search-GenSimon promise with respect to subgroup $M^{-1}(\mathcal{L}_C)$
- $M^{-1}(\mathcal{L}_C) = M^{-1}(\mathcal{L}_C \cap \mathrm{im}(M))$, and this subspace always has dimension equal to $\dim(\mathcal{L}_C \cap \mathrm{im}(M))$ since M is full rank.
- The probability that a random dimension $n - i + 1$ subspace of $\mathbb{Z}_2^n$ intersects a fixed i-dimensional subspace on exactly a line is $\Omega(1)$.

Since the success condition is efficiently checkable, this procedure can indeed be repeated, completing the analysis. □

Using this worst-case hardness reduction, Theorem X.2 implies the following.

Theorem X.6. *Assume the existence of sub-exponentially secure $i\mathcal{O}$ and sub-exponentially secure one-way functions. Then, there is a polynomial $p(n)$ such that if* Search-GenSimon$_{m(\cdot),t(\cdot)}$ *is hard for* BPP *or* P/poly *in the worst case, there is an efficiently sampleable distribution of inputs on which* Search-Simon$_{3n,t(n)\cdot p(n)}$ *is hard on average for* BPP *or* P/poly.

References

[1] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *37th ACM STOC*, H. N. Gabow and R. Fagin, Eds. ACM Press, May 2005, pp. 84–93.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[3] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," 1979.

[4] W. DIFFIE and M. E. HELLMAN, "New directions in cryptography," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 22, no. 6, 1976.

[5] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1364–1396, 1999.

[6] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or A completeness theorem for protocols with honest majority," in *19th ACM STOC*, A. Aho, Ed. ACM Press, May 1987, pp. 218–229.

[7] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more," in *46th ACM STOC*, D. B. Shmoys, Ed. ACM Press, May / Jun. 2014, pp. 475–484.

[8] A. Jain, H. Lin, and A. Sahai, "Indistinguishability obfuscation from well-founded assumptions," in *53rd ACM STOC*, S. Khuller and V. V. Williams, Eds. ACM Press, Jun. 2021, pp. 60–73.

[9] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *CRYPTO 2001*, ser. LNCS, J. Kilian, Ed., vol. 2139. Springer, Berlin, Heidelberg, Aug. 2001, pp. 1–18.

[10] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *54th FOCS*. IEEE Computer Society Press, Oct. 2013, pp. 40–49.

[11] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *ACM CCS 93*, D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, Eds. ACM Press, Nov. 1993, pp. 62–73.

[12] J. Holmgren and A. Lombardi, "Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications)," in *59th FOCS*, M. Thorup, Ed. IEEE Computer Society Press, Oct. 2018, pp. 850–858.

[13] R. Canetti, Y. Chen, and L. Reyzin, "On the correlation intractability of obfuscated pseudorandom functions," in *TCC 2016-A, Part I*, ser. LNCS, E. Kushilevitz and T. Malkin, Eds., vol. 9562. Springer, Berlin, Heidelberg, Jan. 2016, pp. 389–415.

[14] Y. T. Kalai, G. N. Rothblum, and R. D. Rothblum, "From obfuscation to the security of Fiat-Shamir for proofs," in *CRYPTO 2017, Part II*, ser. LNCS, J. Katz and H. Shacham, Eds., vol. 10402. Springer, Cham, Aug. 2017, pp. 224–251.

[15] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs, "Fiat-Shamir: from practice to theory," in *51st ACM STOC*, M. Charikar and E. Cohen, Eds. ACM Press, Jun. 2019, pp. 1082–1090.

[16] M. Zhandry, "The magic of ELFs," in *CRYPTO 2016, Part I*, ser. LNCS, M. Robshaw and J. Katz, Eds., vol. 9814. Springer, Berlin, Heidelberg, Aug. 2016, pp. 479–508.

[17] G. Asharov and G. Segev, "Limits on the power of indistinguishability obfuscation and functional encryption," in *56th FOCS*, V. Guruswami, Ed. IEEE Computer Society Press, Oct. 2015, pp. 191–209.

[18] N. Bitansky, A. Degwekar, and V. Vaikuntanathan, "Structure vs. hardness through the obfuscation lens," in *CRYPTO 2017, Part I*, ser. LNCS, J. Katz and H. Shacham, Eds., vol. 10401. Springer, Cham, Aug. 2017, pp. 696–723.

[19] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *ASIACRYPT 2011*, ser. LNCS, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, Berlin, Heidelberg, Dec. 2011, pp. 344–371.

[20] S. Hirahara, R. Ilango, and R. R. Williams, "Beating brute force for compression problems," in *56th ACM STOC*, B. Mohar, I. Shinkar, and R. O'Donnell, Eds. ACM Press, Jun. 2024, pp. 659–670.

[21] N. Mazor and R. Pass, "The non-uniform perebor conjecture for time-bounded kolmogorov complexity is false," in *ITCS 2024*, V. Guruswami, Ed., vol. 287. LIPIcs, Jan. / Feb. 2024, pp. 80:1–80:20.

[22] Y. Dodis, A. Jain, T. Moran, and D. Wichs, "Counterexamples to hardness amplification beyond negligible," in *TCC 2012*, ser. LNCS, R. Cramer, Ed., vol. 7194. Springer, Berlin, Heidelberg, Mar. 2012, pp. 476–493.

[23] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited (preliminary version)," in *30th ACM STOC*. ACM Press, May 1998, pp. 209–218.

[24] R. Canetti, Y. Chen, L. Reyzin, and R. D. Rothblum, "Fiat-Shamir and correlation intractability from strong KDM-secure encryption," in *EUROCRYPT 2018, Part I*, ser. LNCS, J. B. Nielsen and V. Rijmen, Eds., vol. 10820. Springer, Cham, Apr. / May 2018, pp. 91–122.

[25] R. Jawale, Y. T. Kalai, D. Khurana, and R. Y. Zhang, "SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE," in *53rd ACM STOC*, S. Khuller and V. V. Williams, Eds. ACM Press, Jun. 2021, pp. 708–721.

[26] A. R. Choudhuri, A. Jain, and Z. Jin, "SNARGs for $\mathcal{P}$ from LWE," in *62nd FOCS*. IEEE Computer Society Press, Feb. 2022, pp. 68–79.

[27] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev, "One-way functions and (im)perfect obfuscation," in *55th FOCS*. IEEE Computer Society Press, Oct. 2014, pp. 374–383.

[28] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, M. A. Harrison, R. B. Banerji, and J. D. Ullman, Eds. ACM, 1971, pp. 151–158. [Online]. Available: https://doi.org/10.1145/800157.805047

[29] L. A. Levin, "Universal search problems," *Problems of Information Transmission*, vol. 9, no. 3, pp. 115–116, 1973, originally published in Russian.

[30] R. Williams, "Improving exhaustive search implies superpolynomial lower bounds," *SIAM J. Comput.*, vol. 42, no. 3, pp. 1218–1244, 2013. [Online]. Available: https://doi.org/10.1137/10080703X

[31] S. Hirahara, "Average-case hardness of NP from exponential worst-case hardness assumptions," in *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, S. Khuller and V. V. Williams, Eds. ACM, 2021, pp. 292–302. [Online]. Available: https://doi.org/10.1145/3406325.3451065

[32] R. Impagliazzo, "A personal view of average-case complexity," in *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*. IEEE Computer Society, 1995, pp. 134–147. [Online]. Available: https://doi.org/10.1109/SCT.1995.514853

[33] N. Vyas and R. Williams, "On super strong eth," *Journal of Artificial Intelligence Research*, vol. 70, pp. 473–495, 2021.

[34] A. Drucker, "Nondeterministic direct product reductions and the success probability of SAT solvers," in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. IEEE Computer Society, 2013, pp. 736–745. [Online]. Available: https://doi.org/10.1109/FOCS.2013.84

[35] B. Dubrov and Y. Ishai, "On the randomness complexity of efficient sampling," in *38th ACM STOC*, J. M. Kleinberg, Ed. ACM Press, May 2006, pp. 711–720.

[36] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *40th ACM STOC*, R. E. Ladner and C. Dwork, Eds. ACM Press, May 2008, pp. 187–196.

[37] O. Goldreich, A. Sahai, and S. P. Vadhan, "Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK," in *CRYPTO'99*, ser. LNCS, M. J. Wiener, Ed., vol. 1666. Springer, Berlin, Heidelberg, Aug. 1999, pp. 467–484.

[38] E. Allender, S. Hirahara, and H. Tirumala, "Kolmogorov complexity characterizes statistical zero knowledge," in *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, ser. LIPIcs, Y. T. Kalai, Ed., vol. 251. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 3:1–3:19. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2023.3

[39] C. Mu, S. Nassar, R. D. Rothblum, and P. N. Vasudevan, "Strong batching for non-interactive statistical zero-knowledge," in *EUROCRYPT 2024, Part VI*, ser. LNCS, M. Joye and G. Leander, Eds., vol. 14656. Springer, Cham, May 2024, pp. 241–270.

[40] D. R. Simon, "On the power of quantum computation," *SIAM journal on computing*, vol. 26, no. 5, pp. 1474–1483, 1997.

[41] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, "Exponential algorithmic speedup by a quantum walk," in *35th ACM STOC*. ACM Press, Jun. 2003, pp. 59–68.

[42] T. Yamakawa and M. Zhandry, "Verifiable quantum advantage without structure," in *63rd FOCS*. IEEE Computer Society Press, Oct. / Nov. 2022, pp. 69–74.

[43] B. Barak, S. J. Ong, and S. P. Vadhan, "Derandomization in cryptography," *SIAM J. Comput.*, vol. 37, no. 2, pp. 380–400, 2007. [Online]. Available: https://doi.org/10.1137/050641958

[44] N. Bitansky, S. Goldwasser, A. Jain, O. Paneth, V. Vaikuntanathan, and B. Waters, "Time-lock puzzles from randomized encodings," in *ITCS 2016*, M. Sudan, Ed. ACM, Jan. 2016, pp. 345–356.

[45] R. Ilango, J. Li, and R. R. Williams, "Indistinguishability obfuscation, range avoidance, and bounded arithmetic," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, B. Saha and R. A. Servedio, Eds. ACM, 2023, pp. 1076–1089. [Online]. Available: https://doi.org/10.1145/3564246.3585187

[46] R. Kleinberg, O. Korten, D. Mitropolsky, and C. H. Papadimitriou, "Total functions in the polynomial hierarchy," in *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, ser. LIPIcs, J. R. Lee, Ed., vol. 185. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 44:1–44:18. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2021.44

[47] L. G. Valiant and V. V. Vazirani, "NP is as easy as detecting unique solutions," *Theor. Comput. Sci.*, vol. 47, no. 3, pp. 85–93, 1986. [Online]. Available: https://doi.org/10.1016/0304-3975(86)90135-0

[48] S. Goldwasser and M. Sipser, "Private coins versus public coins in interactive proof systems," *Adv. Comput. Res.*, vol. 5, pp. 73–90, 1989.

[49] L. Adleman, "Two theorems on random polynomial time," in *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, 1978, pp. 75–83.

[50] D. E. Muller and F. P. Preparata, "Bounds to complexities of networks for sorting and for switching," *J. ACM*, vol. 22, no. 2, pp. 195–201, 1975. [Online]. Available: https://doi.org/10.1145/321879.321882

[51] A. Lombardi and V. Vaikuntanathan, "Correlation-intractable hash functions via shift-hiding," in *ITCS 2022*, M. Braverman, Ed., vol. 215. LIPIcs, Jan. / Feb. 2022, pp. 102:1–102:16.

[52] D. Boneh and B. Waters, "Constrained pseudorandom functions and their applications," in *ASIACRYPT 2013, Part II*, ser. LNCS, K. Sako and P. Sarkar, Eds., vol. 8270. Springer, Berlin, Heidelberg, Dec. 2013, pp. 280–300.

[53] E. Boyle, S. Goldwasser, and I. Ivan, "Functional signatures and pseudorandom functions," in *PKC 2014*, ser. LNCS, H. Krawczyk, Ed., vol. 8383. Springer, Berlin, Heidelberg, Mar. 2014, pp. 501–519.

[54] A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias, "Delegatable pseudorandom functions and applications," in *ACM CCS 2013*, A.-R. Sadeghi, V. D. Gligor, and M. Yung, Eds. ACM Press, Nov. 2013, pp. 669–684.

[55] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions (extended abstract)," in *25th FOCS*. IEEE Computer Society Press, Oct. 1984, pp. 464–479.

[56] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Contemporary Mathematics*, vol. 305, pp. 53–74, 2002.

[57] N. Bitansky, I. Haitner, I. Komargodski, and E. Yogev, "Distributional collision resistance beyond one-way functions," in *EUROCRYPT 2019, Part III*, ser. LNCS, Y. Ishai and V. Rijmen, Eds., vol. 11478. Springer, Cham, May 2019, pp. 667–695.

[58] C. Peikert and S. Shiehian, "Privately constraining and programming PRFs, the LWE way," in *PKC 2018, Part II*, ser. LNCS, M. Abdalla and R. Dahab, Eds., vol. 10770. Springer, Cham, Mar. 2018, pp. 675–701.