Witness Encryption and NP-Hardness of Learning

Halley Goldberg
Simon Fraser University, Burnaby, Canada

Valentine Kabanets
Simon Fraser University, Burnaby, Canada

Abstract

We study connections between two fundamental questions from computer science theory. (1) Is *witness encryption* possible for NP [11]? That is, given an instance x of an NP-complete language L, can one encrypt a secret message with security contingent on the ability to provide a witness for $x \in L$? (2) Is *computational learning* (in the sense of [46, 30]) hard for NP? That is, is there a polynomial-time reduction from instances of L to instances of learning?

Our main contribution is that certain formulations of NP-hardness of learning characterize the existence of witness encryption for NP. More specifically, we show:

- witness encryption for a language $L \in \mathsf{NP}$ is equivalent to a half-Levin reduction from L to the Computational Gap Learning problem (denoted CGL [2]),

where a half-Levin reduction is the same as a Levin reduction but only required to preserve witnesses in one direction, and CGL formalizes agnostic learning as a decision problem. We show versions of the statement above for witness encryption secure against non-uniform and uniform adversaries. We also show that witness encryption for NP with ciphertexts of logarithmic length, along with a circuit lower bound for E, are together equivalent to NP-hardness of a generalized promise version of MCSP.

We complement the above with a number of unconditional NP-hardness results for agnostic PAC learning. Extending a result of [16] to the standard setting of boolean circuits, we show NP-hardness of "semi-proper" learning. Namely:

- for some polynomial s, it is NP-hard to agnostically learn circuits of size $s(n)$ by circuits of size $s(n) \cdot n^{1/(\log\log n)^{O(1)}}$.

Looking beyond the computational model of standard boolean circuits enables us to prove NP-hardness of improper learning (ie. without a restriction on the size of hypothesis returned by the learner). We obtain such results for:

- learning circuits with oracle access to a given randomly sampled string, and
- learning RAM programs.

In particular, we show that a variant of MINLT [31] for RAM programs is NP-hard with parameters corresponding to the setting of improper learning. We view these results as partial progress toward the ultimate goal of showing NP-hardness of learning boolean circuits in an improper setting.

Lastly, we give some consequences of NP-hardness of learning for private- and public-key cryptography. Improving a main result of [2], we show that if improper agnostic PAC learning is NP-hard under a randomized non-adaptive reduction (with some restrictions), then $\mathsf{NP} \not\subseteq \mathsf{BPP}$ implies the existence of i.o. one-way functions. In contrast, if CGL is NP-hard under a half-Levin reduction, then $\mathsf{NP} \not\subseteq \mathsf{BPP}$ implies the existence of i.o. public-key encryption.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases agnostic PAC learning, witness encryption, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.CCC.2025.34

1 Introduction and Background

It has long been recognized that learning and cryptography are two sides of the same coin. As Valiant already observed in his paper introducing PAC learning, its easiness would preclude the existence of pseudorandom functions [46]. In 1993, building on a work of Impagliazzo

40th Computational Complexity Conference (CCC 2025).
Editor: Srikanth Srinivasan; Article No. 34; pp. 34:1–34:43
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and Levin [24], Blum, Furst, Kearns, and Lipton sharpened Valiant's observation by showing that private-key cryptography can be broken if and only if PAC learning is easy in an average-case setting [6]. A number of more recent results have developed such connections further (eg. [41, 39, 18]).

In the present work, we turn to more structured kinds of hardness of learning (namely, NP-hardness) and cryptography (namely, witness encryption), which we show to be likewise equivalent.

NP-hardness of PAC learning

Learning theory asks whether an efficient entity can hope to learn general truths about reality from necessarily limited input from experience. Valiant formalized this idea in 1984 with his definition of Probably Approximately Correct (PAC) Learning [46]. In this framework, a learner for some representation class $\mathcal{C}$ gets access to labeled examples $(x, f(x))$ for some unknown concept $f \in \mathcal{C}$ and x randomly sampled according to an arbitrary distribution $\mathcal{D}$. The learner is asked to produce, with high probability, a hypothesis that approximates f well over $\mathcal{D}$. One can distinguish between "proper" and "improper" settings: in proper learning, the learner should produce a hypothesis that also belongs to $\mathcal{C}$, whereas in improper learning, there is no restriction on the complexity of the hypothesis. One can also consider various "semi-proper" settings, in which the hypothesis must belong to some specific concept class $\mathcal{C}'$ containing $\mathcal{C}$. Lastly, one can relax the model of learning to be "agnostic" in that the labeled examples (x, b) do not necessarily reflect the values of a function $f \in \mathcal{C}$, and the learner is only asked to do as well as the best function in $\mathcal{C}$ does [30].

A long line of work has established NP-hardness of PAC learning for various concept classes (eg. [43, 1, 16, 32]). Early results obtained NP-hardness of proper learning of restricted circuit classes. Over time, progress has pushed toward NP-hardness of "less proper" learning of less restricted circuit classes. However, NP-hardness of learning *unrestricted* boolean circuits in the *improper* setting remains elusive. By this, we refer to the problem of learning the concept class $\mathsf{SIZE}[s(n)]$ for some fixed polynomial s by a hypothesis of arbitrary polynomial size. Proving NP-hardness in this setting seems especially challenging in light of a 2008 result of Applebaum et al.:

▶ **Theorem 1** ([2]). *Suppose there is a randomized non-adaptive honest*[1] *reduction R and a polynomial $s : \mathbb{N} \to \mathbb{N}$ such that, for every constant $c \in \mathbb{N}$, R reduces* SAT *to agnostically PAC learning* $\mathsf{SIZE}[s(n)]$ *by* $\mathsf{SIZE}[s(n)^c]$*. Then, if* NP *is hard on average, infinitely-often one-way functions exist.*

Of course, there are various ways in which one could formulate reductions to PAC learning. The aforementioned paper [2] describes two (of many) possibilities. The statement of Theorem 1 refers to one kind of NP-hardness reduction described in which a randomized oracle machine on an input of length $n \in \mathbb{N}$ makes non-adaptive queries $\mathcal{E}$ to its oracle, with $\mathcal{E}$ being a $\mathsf{poly}(n)$-size circuit sampling example-label pairs $(x, b) \in \{0,1\}^{n^{\Omega(1)}} \times \{0,1\}$. The oracle returns, for each query, a polynomial-size circuit h having high agreement over $\mathcal{E}$ (i.e. $\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b]$ close to 1), which the reduction may use in any way.

In the other kind of NP-hardness reduction that [2] describes, the reduction only uses its learning oracle to *decide* if a small circuit consistent with $\mathcal{E}$ exists. The relevant decision problem introduced by [2] is the *Computational Gap Learning Problem* (CGL). The authors

[1] Roughly, "honest" means that the input length of the learning instance is polynomially related to the length of the input to the reduction. See Definition 28.

show that if CGL is NP-hard under a deterministic many-one reduction, then every language in NP has a statistical zero-knowledge argument.

▶ Remark 2. The results in [2] indicate it is a challenge to show NP-hardness of agnostic PAC learning, as it would prove breakthrough results in cryptography, which, although conjectured to be true, seem beyond our reach at the moment. However, we should note that the assumption in Theorem 1 is actually quite strong in its requirement that the *same* reduction must work for *every* constant $c \in \mathbb{N}$. Theorem 1 also requires that queries $\mathcal{E}$ made by the reduction sample pairs (x, b) with the length of x *polynomially* related to the length of the given SAT instance. It is possible that PAC learning is NP-hard in the improper setting without either of these conditions being met. For example, if a different reduction exists for each $c \in \mathbb{N}$ (perhaps taking longer time for larger c), then the easiness of improper learning would still imply the easiness of NP.

Witness encryption for NP

First introduced in a 2013 paper of Garg, Gentry, Sahai, and Waters [11], witness encryption (WE) is a cryptographic primitive in which security is contingent on the ability to provide a proof of an answer to a hard problem. As Garg et al. put it,

> What if we don't really care if [the receiver] knows a secret key, but we do care if he knows a solution to a crossword puzzle that we saw in the Times? [11]

More formally, the functionality of a witness encryption scheme is based on the structure of a particular language $L \in \mathsf{NP}$. An instance x of L is made public, and anyone can encrypt a secret message using the witness encryption scheme along with x. It is guaranteed that if x belongs to L, then anyone with a witness proving $x \in L$ can decrypt, but if x does not belong to L, then no one can do so.

Witness encryption is intimately connected to another cryptographic primitive known as indistinguishability obfuscation (iO) [3], which has been studied extensively (eg. [33, 10, 23, 37]) and has many candidate constructions (see [28]). iO is known to imply WE, though the converse is not known [10]. (We give a sketch of the proof that iO implies WE in Appendix A.) WE is also closely related to secret-sharing, another important cryptographic tool [4, 34]. In fact, the candidate construction of WE in [11] also yields a Rudich-type secret sharing scheme. As with much of cryptography, we have no unconditional proof that WE is possible.

WE alone does not imply the existence of one-way functions, since it is trivially possible if $\mathsf{NP} \subseteq \mathsf{BPP}$. However, WE *along with* one-way functions becomes very powerful, together yielding public key encryption, as demonstrated in [11]. By combining this result with two subsequent works, one obtains the following.

▶ **Theorem 3** ([11, 33, 19]; see [35])**.** *Assume* $\mathsf{NP} \not\subseteq \mathsf{io\text{-}P/poly}$*. If* $\mathsf{P/poly}$*-secure witness encryption exists for* NP*, then public-key encryption is possible.*

In this paper, we show that CGL is NP-hard if and only if witness encryption exists for NP. We prove statements along these lines for a few different settings: WE secure against PPT adversaries, WE secure against polynomial-size circuits, and WE with logarithmic-length ciphertexts; the latter roughly characterizes the NP-hardness of a promise version of the Minimum Circuit Size Problem [29].

In contrast to the aforementioned results of [2] (such as Theorem 1), not only do we show that proving NP-hardness of learning is a challenge as it would imply a breakthrough

cryptographic construction (namely, WE), but we also show the *converse*: progress in cryptography (showing that WE is possible) would imply that agnostic PAC learning is NP-hard. Hence, if one believes in cryptography such as iO or WE, then one should also believe in the NP-hardness of improper agnostic learning, even in the extremely restrictive sense of a deterministic, many-one, half-Levin reduction to CGL. We interpret our results as evidence that improper learning is likely NP-hard.

As a complement, we prove a number of *unconditional* NP-hardness results for agnostic PAC learning. For one, we obtain NP-hardness of agnostic learning for unrestricted boolean circuits in a semi-proper setting that improves on the previous state of the art. In other settings, we obtain NP-hardness of *improper* agnostic learning. Though we have made some partial progress toward showing NP-hardness unconditionally, we feel that it remains a promising direction for further research.

Lastly, we give some applications of our results for the possibility of private- and public-key cryptography. We improve Theorem 1 to conditionally obtain infinitely-often one-way functions from *worst-case* hardness of NP. In contrast, if learning is NP-hard in a more restricted sense (namely, a half-Levin reduction to CGL), then one obtains infinitely-often public-key encryption assuming $\mathsf{NP} \not\subseteq \mathsf{BPP}$. Along the way, we prove Theorem 3 under the weaker uniform assumption $\mathsf{NP} \not\subseteq \mathsf{BPP}$.[2]

1.1 Our Results

Connections between witness encryption and NP-hardness of learning

We first discuss a set of results that characterize witness encryption in terms of NP-hardness reductions to PAC learning.

Witness encryption for an NP language L consists of a pair of algorithms $(\mathsf{Enc}, \mathsf{Dec})$ such that, for an L-instance x and a secret bit $b \in \{0,1\}$, $\mathsf{Enc}(x,b)$ outputs a ciphertext string c using some randomness. It is guaranteed that, if $x \in L$ and w is a witness for the membership of x under a fixed witness relation for L, then $\mathsf{Dec}(c,x,w)$ deterministically recovers the bit b. On the other hand, if $x \notin L$, then $\mathsf{Enc}(x,0)$ and $\mathsf{Enc}(x,1)$ are indistinguishable (within a negligible difference) for polynomial-size circuits (if the witness encryption is P/poly-secure) or PPT algorithms (if the witness encryption is BPP-secure). See Definition 39.

Our first result characterizes P/poly-secure witness encryption in terms of a deterministic many-one reduction to a promise problem known as CGL, introduced in a work of Applebaum et al. [2]. Inputs to CGL are distributions $\mathcal{E}$ over string-label pairs $(x,b) \in \{0,1\}^n \times \{0,1\}$, where $\mathcal{E}$ is represented by a $\mathsf{poly}(n)$-size sampling circuit. For parameters $s_1 < s_2 \in \mathbb{N}$ and $0 < \varepsilon < 1/2$, a distribution $\mathcal{E}$ is a yes-instance of $\mathsf{CGL}_{s_1}^{s_2}[\varepsilon]$ if there is a circuit C of size at most s_1 such that, for $(x,b) \sim \mathcal{E}$, $\Pr[C(x) = b] = 1$; $\mathcal{E}$ is a no-instance of $\mathsf{CGL}_{s_1}^{s_2}[\varepsilon]$ if for every circuit C' of size at most s_2, for $(x,b) \sim \mathcal{E}$, $\Pr[C'(x) = b] < 1/2 + \varepsilon$.

Note that a many-one reduction to CGL is quite a restrictive notion of reducibility to agnostic PAC learning: the reduction only uses its learning oracle to *decide* if a small circuit exists for the given distribution. As discussed earlier, one can imagine other kinds of reduction to learning that make more robust use of a learner: for example, actually examining the circuits returned by the learning oracle.

[2] We believe that this last statement follows from a combination of techniques used in prior work ([11, 33, 19]; see [35]), but we have not seen the uniform version stated. In any case, we offer an alternative proof that does not rely on properties of statistical zero-knowledge arguments.

We further restrict the definition of a many-one reduction to CGL by requiring that it be a *half-Levin reduction*. This is a term that we introduce here for a kind of reduction that is intermediate between a many-one reduction and a Levin reduction. A half-Levin reduction (R, R_{wit}) from L_1 to L_2 is a pair of polynomial-time machines such that R is a many-one reduction from L_1 to L_2, and R_{wit} transforms witnesses for $x \in L_1$ into witnesses for $R(x) \in L_2$. In contrast to a standard Levin reduction, we do not require a third algorithm transforming L_2-witnesses into L_1-witnesses. We emphasize that the half-Levin reductions considered in this work are deterministic unless stated otherwise.

We call a reduction to CGL *honest* if, on inputs of length $n \in \mathbb{N}$, it outputs a distribution $\mathcal{E}$ supported over $\{0,1\}^{n^{\Omega(1)}} \times \{0,1\}$.

We are now ready to state our main result. In general, fixing parameters s_1, s_2, and ε, we show that a half-Levin reduction from a language L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ is equivalent to a witness encryption scheme for L with running time roughly s_1, security against circuits of size roughly s_2, and with advantage parameter roughly ε; see Lemmas 52 and 54. For the special case of P/poly-secure witness encryption, we get the following.

▶ **Theorem 4.** *Consider any language $L \in \mathsf{NP}$. The following are equivalent.*

1. *P/poly-secure witness encryption exists for L;*
2. *there exists a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2, ε^{-1}, (R, R_{wit}) is an honest half-Levin reduction from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$.*

We also note that the above characterization is related to a very recent result from Liu, Mazor, and Pass, which shows that P/poly-secure witness encryption for an NP-language L is equivalent to the existence of a laconic special-honest verifier zero-knowledge argument for L [36]. We refer to that paper for the definition of this kind of protocol, though we mention that the term "laconic" means that the total length of the prover's messages is bounded by $O(\log n)$. As a corollary of [36] and our Theorem 4 above, we obtain the following.

▶ **Corollary 5.** *Consider any language $L \in \mathsf{NP}$. The following are all equivalent.*

1. *P/poly-secure witness encryption exists for L;*
2. *there exists a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2, ε^{-1}, (R, R_{wit}) is an honest half-Levin reduction from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$;*
3. *there exists a laconic special-honest verifier zero-knowledge argument for L.*

One may also be interested in the possibility that NP has witness encryption schemes secure against PPT algorithms but not against polynomial-size circuits. We therefore present a kind of reduction to learning that is equivalent to BPP-secure witness encryption. This kind of reduction is intermediate between a half-Levin reduction to CGL and the more general kind of reduction described in [2] wherein the reduction may actually inspect the hypotheses returned by the learner and use them in any way.

More specifically, in our case, the reduction makes a single query $\mathcal{E}$ to the *search* version of CGL, getting back a small circuit consistent with $\mathcal{E}$ if $\mathcal{E}$ is a yes-instance of CGL. We still require the reduction to be half-Levin. Furthermore, we require the reduction to be BPP-black-box;[3] see Definition 32.

▶ **Theorem 6.** *Consider any language $L \in \mathsf{NP}$. The following are equivalent.*

[3] The standard notion of a *class-specific black-box* reduction from a language L_1 to a language L_2 was defined by [13] as a way to formalize an *intermediate* case between treating L_2 as an oracle ("black box") and requiring a source code for an L_2 algorithm implemented in some complexity class ("white box"); see Definitions 30 and 31.

1. BPP-*secure witness encryption exists for* L*;*
2. *there exist a polynomial* s_1 *and a pair of machines* (R, R_{wit}) *such that, for all polynomials* s_2 *and* ε^{-1}*,* (R, R_{wit}) *is an honest* BPP-*black-box half-Levin reduction from* L *to* $\mathsf{search\text{-}CGL}^{s_2}_{s_1}[\varepsilon]$*.*

So far, we have only discussed learning in the "polynomial regime", where the distribution $\mathcal{E}$ over pairs (x, b) is in PSAMP/poly. However, the original PAC learning framework of Valiant makes no such restriction: a PAC learner should be able to learn over *arbitrary* distributions [46]. We thus consider reductions to learning that make further use of this capacity. In particular, imagine a polynomial-time reduction that, on input length n, queries a learner for a function on $O(\log n)$-length inputs. In this case, it will be possible to ask the learner to work on inefficiently samplable distributions, and a truth-table of the function in question can be given to the learner explicitly. In this particular setup, the distributions queried represent well-defined functions, which may not be the case in the setup above with CGL. In this sense, we get closer to standard PAC learning rather than agnostic PAC learning.

This notion of learning in the "exponential regime" is closely related to MCSP, which asks, given a truth-table T and a size threshold s, whether there exists a circuit of size s consistent with T. We generalize the problem to the context of learning by providing, along with T, a probability distribution μ represented as a table. We call this problem "Gap Distributional MCSP", denoted GapDistMCSP. Specifically, (T, μ) is a yes-instance of $\mathsf{GapDistMCSP}^{s_2}_{s_1}[\varepsilon]$ if there exists a circuit C of size at most s_1 such C is consistent with T, and (T, μ) is a no-instance if for every circuit C' of size at most s_2, the probability over $x \sim \mu$ that $C(x) = T(x)$ is at most $1/2 + \varepsilon$. Note that this definition naturally generalizes AveMCSP as defined in [45] (and in [7], denoted MACSP), where the definition is the same except the distribution fixed to uniform.

We show that NP-hardness of learning in this sense is equivalent to witness encryption for NP such that the ciphertexts output by Enc have only *logarithmic* length, along with a circuit lower bound for E. Below we use $\mathsf{SIZE}[s(n)]$ to denote the class of n-input boolean functions computable by circuits of size $s(n)$; for $0 \leq \alpha \leq 1$, we use $\mathsf{SIZE}^{\alpha}_{D}[s(n)]$ to denote the class of n-input boolean functions that can be computed by circuits of size $s(n)$ on average, with probability at least α over inputs $x \in \{0,1\}^n$ sampled according to the distribution D.

▶ **Theorem 7.** *Consider any language* $L \in \mathsf{NP}$*, constant* $c \in \mathbb{N}$*, polynomial* s_1*, and* $s_2, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$*. The following are equivalent.*

1. - $(\mathsf{SIZE}[\Omega(s_2(n))], O(\varepsilon(n)))$*-secure* $(c \log n + O(1))$*-length* $\widetilde{O}(s_1(n))$*-decryption-time witness encryption exists for* L*; and*
   - $\mathsf{E} \not\subseteq \mathsf{io\text{-}SIZE}^{\frac{1}{2}+O(\varepsilon(n))}_{D}[\Omega(s_2(n))]$ *for some* E*-computable distribution family* D*.*
2. *There is a half-Levin reduction* (R, R_{wit}) *mapping* n*-length instances of* L *to* $O(n^c)$*-length instances of* $\mathsf{GapDistMCSP}^{\Omega(s_2(n))}_{\widetilde{O}(s_1(n))}[O(\varepsilon)]$*, where* R_{wit} *runs in time at most* $\widetilde{O}(s_1(n))$*.*

Note that we obtain correspondences between the output length of the reduction R and the output length of Enc, the s_1 parameter of GapDistMCSP and the running time of Dec, the s_2 parameter of GapDistMCSP and the security of witness encryption, and the ε parameter of GapDistMCSP and the advantage parameter of witness encryption.

Unconditional NP-hardness of semi-proper learning for P/poly

As discussed in the introduction, NP-hardness of proper agnostic PAC-learning for polynomial-size circuits is known. However, NP-hardness of improper learning is a major open question

that remains elusive. In this section, we advance toward NP-hardness of "less proper" agnostic PAC learning for polynomial-size circuits.

First, we show that for polynomials s_1 and ε^{-1}, letting $s_2(n) = s_1(n) \cdot n^{1/(\log\log n)^{O(1)}}$, $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ is NP-hard under a randomized many-one reduction. In particular,

▶ **Theorem 8.** *For some constant $c \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ with $g(n) = n^{1/(\log\log n)^{O(1)}}$ for $n \in \mathbb{N}$, for any polynomials ε^{-1} and s_1 such that $s_1(n) \geq \varepsilon^{-c}(n)$ for $n \in \mathbb{N}$, there is a randomized many-one reduction from* SAT *to* $\mathsf{CGL}^{s_1 \cdot g}_{s_1}[\varepsilon]$.

As a straightforward consequence of Theorem 8, we obtain NP-hardness of agnostic PAC-learning in the semi-proper setting of learning size-$s(n)$ circuits by size-$s(n) \cdot n^{1/(\log\log n)^{O(1)}}$ circuits.

▶ **Corollary 9.** *For some polynomial s, it is* NP*-hard under a randomized one-query reduction to agnostically PAC learn* $\mathsf{SIZE}[s(n)]$ *by* $\mathsf{SIZE}[s(n) \cdot n^{1/(\log\log n)^{O(1)}}]$ *over a flat* P/poly*-samplable distribution.*

This improves on a prior work of Hirahara [16], which contains two main results. The first shows that, for polynomials s, it is NP-hard to agnostically learn *programs* of size $s(n)$ by *programs* of size $s(n) \cdot n^{1/(\log\log n)^{O(1)}}$. This is accomplished by reduction from the "minimum monotone satisfying assignment" problem, MMSA, which is known to be NP-hard to approximate by way of a PCP theorem [8, 9]. We note that Hirahara's proof makes use of efficient secret sharing schemes, a cryptographic primitive closely related to witness encryption and known to exist unconditionally (with statistical security) for access structures that are monotone formulae [27, 5] (see [4, 11]).

The second main result of [16] shows that MCSP^*, a partial function version of MCSP, is likewise NP-hard. This proof requires even more sophisticated techniques in order to produce a truth-table on logarithmic-length inputs. In particular, Hirahara reduces from a gap version of the "collective minimum (weight) monotone satisfying assignment problem" (CMMSA), which generalizes MMSA. Using the techniques of that proof and padding the inputs to have polynomial length, one can obtain NP-hardness of learning P/poly, but with a smaller gap.

▶ **Theorem 10** (Implicit in [16])**.** *For any constant $\beta > 0$, there exists a constant $\alpha > 0$ such that for some polynomial s, it is* NP*-hard under a randomized one-query reduction to agnostically PAC learn* $\mathsf{SIZE}[s(n)]$ *by* $\mathsf{SIZE}[s(n) \cdot 2^{\alpha(\log n)^{1-\beta}}]$ *over a flat* P/poly*-samplable distribution.*

However, those ideas do not extend to obtain the larger $s(n)$ vs. $s(n) \cdot n^{1/(\log\log n)^{O(1)}}$ gap for circuits rather than programs as in Corollary 9.

▶ Remark 11. We note that Theorem 8 is in fact proved by a *randomized half-Levin* reduction from GapMMSA to CGL! That is, there is a poly-time machine R_{wit} such that, with high probability over r, if $\mathcal{E}_r$ is the output of the reduction of Theorem 8 run on a GapMMSA instance φ with randomness r, and α is a witness for φ, then $R_{wit}(\varphi, \alpha, r)$ outputs a circuit certifying that $\mathcal{E}_r$ is a yes-instance of CGL. However, unfortunately, we do not know how to extend the results of the previous section to obtain non-trivial witness encryption from a *randomized* half-Levin reduction, particularly since the amount of randomness required is greater than the parameter s_2 (which translates to the security of the witness encryption scheme). In any case, our results show that if the reduction of Theorem 8 can be suitably derandomized (with the running time of R_{wit} less than the parameter s_2) then one would unconditionally obtain non-trivial witness encryption.

Unconditional NP-hardness of improper learning for other concept classes

Though in the case of P/poly we are unable to obtain NP-hardness with a gap greater than $n^{1/(\log\log n)^{O(1)}}$, we show in this section that for some related concept classes, *improper* learning is NP-hard. We demonstrate such results for learning oracle circuits and learning RAM programs.

In the following, for a fixed $\lambda : \mathbb{N} \to \mathbb{N}$ and distribution family $D = \{D_m\}_{m\in\mathbb{N}}$, where each D_m is supported over strings of length 2^m and samplable uniformly in time $\mathsf{poly}(2^m)$, D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ is a generalization of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ in which all circuits and samplers take access to an oracle $O \in \{0,1\}^{2^{\lambda(n)}}$ randomly sampled from $D_{\lambda(n)}$. $\mathcal{E}$ is a yes-instance of D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ if there exists a circuit C of size $s_1(n)$ such that for every O in the support of $D_{\lambda(n)}$,

$$\Pr_{(x,b)\sim\mathcal{E}^O}[C^O(x) = b] = 1,$$

and $\mathcal{E}$ is a no-instance of D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ if, with probability $1 - \varepsilon(n)$ over $O \sim D_{\lambda(n)}$, for any circuit C of size $s_2(n)$,

$$\Pr_{(x,b)\sim\mathcal{E}^O}[C^O(x) = b] < \frac{1}{2} + \varepsilon(n).$$

We prove the following NP-hardness of oracle-CGL, from which we will derive further corollaries.

▶ **Theorem 12.** *There exist a distribution family $D \in \mathsf{PSAMP}$, an NP-complete language L, and a polynomial s_1 such that, for all polynomials s_2, ε^{-1}, there is a function $\lambda : \mathbb{N} \to \mathbb{N}$ such that $\lambda(n) = O(\log n)$ for $n \in \mathbb{N}$ and L reduces to D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ under an honest half-Levin reduction.*

Theorem 12 is proved using a framework developed by Huang, Ilango, and Ren for "witness encryption in oracle worlds" [21]. More specifically, drawing on the candidate witness encryption scheme proposed in [11], the authors show that there exist an NP-complete language L and a distribution family $D \in \mathsf{PSAMP}$ such that, with high probability over oracles O sampled randomly from D, witness encryption for L exists with respect to O. That is, Enc, Dec, and adversaries all get oracle access to O.

We may then apply the techniques of one direction of our main equivalence, Theorem 4, to obtain NP-hardness of learning with respect to O. Crucially, in [21], an oracle O with truth-table length n gives rise to a witness encryption scheme secure against non-uniform adversaries of size polynomially related to n. So, to achieve NP-hardness of oracle-CGL with an arbitrary polynomial gap, one only requires oracle truth-tables of polynomial length.

As a consequence of Theorem 12, we show that agnostic learning is NP-hard in an "oracle-PAC" model that we introduce here, which generalizes standard agnostic PAC learning of P/poly. In this model, a learner for $\mathsf{SIZE}[s]$ on input length n is explicitly given the whole truth table of an oracle function $\mathcal{O} : \{0,1\}^{O(\log n)} \to \{0,1\}$ sampled randomly from some distribution $D \in \mathsf{PSAMP}$. The learner is asked to output an oracle circuit h such that $h^{\mathcal{O}}$ approximates the target function almost as well as the best size-s $\mathcal{O}$-oracle circuit; see Definition 23.

▶ **Corollary 13.** *For every constant $c \in \mathbb{N}$ and sufficiently large polynomial s, it is NP-hard under a randomized one-query reduction to agnostically oracle-PAC learn $\mathsf{SIZE}[s(n)]$ by $\mathsf{SIZE}[s(n)^c]$.*

We stress that the above holds under a standard polynomial-time reduction that does not itself require any oracle.

As a second consequence of the NP-hardness of oracle-CGL, we show that it is NP-hard to agnostically learn *RAM programs* in an improper setting. In particular, we focus on a decision problem based on the problem MINLT defined in [31]. In Ko's definition, the input to MINLT consists of pairs $(x_i, b_i)_{i\in[m]} \in (\{0,1\}^n \times \{0,1\})^m$ for some $m = \mathsf{poly}(n)$ along with size and running time parameters $s, t \in \mathbb{N}$ represented in unary. The input is a yes-instance if and only if there exists a program M of size at most s and running time at most t such that $M(x_i) = b_i$ for all $i \in [m]$. Ko exhibited an oracle relative to which MINLT is not NP-hard [31].

Hirahara, using non-relativizing techniques, showed that a certain generalization of MINLT (called GapLearn) is in fact NP-hard [16]. In this promise-problem, the pairs (x_i, b_i) are represented implicitly by a sampling circuit (as in the definition of CGL), and there is a gap, eg., in program size between yes-instances and no-instances. It is easy to see that NP-hardness of this problem (Theorem 1.1 in [16]) implies NP-hardness of Ko's MINLT.

We formulate our definition of GapRAM-MINLT analogously to the definition of GapLearn in [16] but considering programs that are granted random access to their inputs. Namely, the input to $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT is a sampler $\mathcal{E}$ over pairs (x, b) along with parameters $s, t \in \mathbb{N}$. The input $(\mathcal{E}, 1^s, 1^t)$ is a yes-instance if there exists a program of size at most s and running in time at most t such that $\Pr_{(x,b)\sim\mathcal{E}}[M^x = b] = 1$, and $(\mathcal{E}, 1^s, 1^t)$ is a no-instance if for any program of size at most $g(s)$ and running in time at most $g(t)$, $\Pr_{(x,b)\sim\mathcal{E}}[M^x = b] < \frac{1}{2} + \varepsilon(n)$. Here, M^x denotes the program M with random access to its input x.

We obtain the following NP-hardness of GapRAM-MINLT. Note that the parameters we obtain imply NP-hardness of learning RAM programs in an improper setting.

▶ **Corollary 14.** *For all polynomials g and ε^{-1}, $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT is NP-hard under a randomized many-one reduction.*

In contrast, [16] considers programs that do not require random access to their inputs; the gap in program size achieved there is small (namely, $s(n)$ vs. $s(n) \cdot n^{1/(\log\log n)^{O(1)}}$), though the no-instances extend to time-unbounded programs.

Applications for the possibility of cryptography

Finally, we prove a few results demonstrating what cryptography is possible assuming worst-case hardness of NP along with various assumptions concerning witness encryption and NP-hardness of learning.

Assuming a randomized non-adaptive reduction from SAT to improper agnostic learning as described in [2] (see Definition 28), we show that the *worst-case* assumption $\mathsf{NP} \not\subseteq \mathsf{BPP}$ is sufficient to imply one-way functions secure infinitely often against uniform PPT adversaries. This improves a main result of [2] (Theorem 1 in the introduction), which requires *average-case* hardness of NP. In terms of Impagliazzo's five worlds [22], the result of [2] conditionally excludes Pessiland. In contrast, we conditionally exclude both Heuristica and Pessiland.

▶ **Theorem 15.** *Suppose there is a randomized non-adaptive honest reduction R and a polynomial $s : \mathbb{N} \to \mathbb{N}$ such that, for every constant $c \in \mathbb{N}$, R reduces SAT to agnostically PAC learning $\mathsf{SIZE}[s(n)]$ by $\mathsf{SIZE}[s(n)^c]$. Then, unless $\mathsf{NP} \subseteq \mathsf{BPP}$, there exist infinitely-often one-way functions.*

As discussed in the introduction, Garg et al. show that WE for an NP-complete language along with a one-way function is sufficient for public-key encryption [11]. Combining this result, our main equivalence in Theorem 4, and Theorem 15, we obtain that witness encryption simultaneously excludes Heuristica, Pessiland, and Minicrypt.

▶ **Theorem 16.** *Suppose* $\mathsf{P/poly}$*-secure witness encryption exists for* NP*. Then, unless* $\mathsf{NP} \subseteq \mathsf{BPP}$*, there exists public-key encryption secure infinitely often against polynomial-time adversaries.*

We believe that this statement can also be proved by combining prior works [33, 19, 11] (see also [35]), though we have not seen the uniform version stated (ie. the version with the assumption $\mathsf{NP} \not\subseteq \mathsf{BPP}$ rather than $\mathsf{NP} \not\subseteq \mathsf{P/poly}$). In any case, we provide an alternative proof that does not rely on properties of zero-knowledge, such as the "SZK/OWF" characterization from [42].

Theorem 16 and our our main equivalence in Theorem 4 together imply that if learning is NP-hard under a half-Levin reduction to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ working for arbitrary polynomials s_2 and ε^{-1}, one can rule out Minicrypt in addition to Heuristica and Pessiland. Thus, we obtain a stark distinction between the two kinds of reduction to learning defined in [2]. Namely, assuming worst-case hardness of NP, a *randomized non-adaptive* reduction to agnostic learning over $\mathsf{PSAMP/poly}$ yields *one-way functions*, whereas a *deterministic many-one* reduction to CGL, provided it is *half-Levin*, yields *public-key encryption*.

▶ **Theorem 17.** *Suppose there exist a polynomial* s_1 *and a pair of machines* (R, R_{wit}) *such that, for all polynomials* s_2, ε^{-1}*,* (R, R_{wit}) *is an honest half-Levin reduction from* SAT *to* $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$*. Then, unless* $\mathsf{NP} \subseteq \mathsf{BPP}$*, there exists public-key encryption secure infinitely often against polynomial-time adversaries.*

1.2 Related Work

As mentioned above, a very recent work of Liu, Mazor, and Pass shows that witness encryption for NP is equivalent to "laconic special-honest verifier zero-knowledge arguments" for NP [36]. Our main result shows that half-Levin reductions from NP to CGL are equivalent to both. This sharpens the result of Applebaum et al. showing that NP-hardness of CGL under a many-one (not necessarily half-Levin) reduction implies $\mathsf{NP} \subseteq \mathsf{SZKA}$ [2].

Huang, Ilango, and Ren [21] used an assumption of the existence of witness encryption secure against subexponential-size circuits to conclude the NP-hardness of approximating the time-bounded conditional Kolmogorov complexity $\mathsf{K}^t(x \mid y)$, in the regime where $t > |y|$, with essentially optimal gap parameters. This is similar to one direction of our Theorem 4 showing an equivalence between the existence of witness encryption (secure against $\mathsf{P/poly}$) and NP-hardness of a learning problem CGL (under restricted reductions). We note that in our framework, proving such a connection between witness encryption and NP-hardness of learning is much simpler than in the setting considered by [21].

Our Theorem 12 relies on a framework for "witness encryption in oracle worlds" developed in [21] (building on ideas from [11]). The authors of [21] obtain NP-hardness of the Minimum *Oracle* Circuit Size Problem with a large circuit size gap [21]. This is somewhat like an "exponential regime" analogue of our oracle-PAC learning model, but we stress that, in [21], an algorithm for MOCSP needs to work in the worst case over oracle truth-tables. In our case, a learner only needs to work with high probability over oracle truth-tables sampled from some distribution in PSAMP.

An earlier example of an equivalence between NP-hardness of a meta-complexity problem and the existence of a secure cryptographic primitive is the result by Hirahara [17]. It shows an equivalence between the existence of one-way functions and NP-hardness of distributional time-bounded Kolmogorov complexity $d\mathsf{K}^t$ under a certain kind of restricted reductions.

An earlier example of an equivalence between hardness of PAC learning and the existence of a cryptographic primitive is due to Nanashima [39]. In that work, Nanashima shows that

auxiliary-input one-way functions exist if and only if a certain average-case, or "heuristic", variant of PAC learning is hard for efficient algorithms. In contrast, we focus on NP-*hardness* of worst-case agnostic PAC learning.

1.3 Overview of Techniques

Proof of Theorem 4

We start with an informal overview of the proof of our main equivalence, Theorem 4. In both directions, the constructions employed and the proofs of correctness are straightforward. We first show how to obtain a half-Levin reduction from L to CGL from a P/poly-secure witness encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ for L. In particular, given an L-instance z, the reduction outputs a distribution $\mathcal{E}$ as follows.

Sample $b \sim \mathcal{U}$. Let $x := \mathsf{Enc}(z, b)$. Output (x, b).

Note that we can assume that the output length of Enc is at least $n^{\Omega(1)}$ without loss of generality, so the reduction described here is honest.

In the case that $z \in L$ with witness w, $\mathcal{E}$ is a yes-instance of CGL, since the circuit defined by $\mathsf{Dec}(-, z, w)$ recovers the bit b from x with probability 1 over $\mathcal{E}$. This is true by the correctness of $(\mathsf{Enc}, \mathsf{Dec})$. Also note that, since Dec is a uniform polynomial-time algorithm, the circuit $\mathsf{Dec}(-, z, w)$ can be constructed in polynomial time given z and w. Thus, our reduction is half-Levin. In the case that $z \notin L$, we observe that if there were a circuit C of polynomial size such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(|x|)},$$

then by definition of $\mathcal{E}$,

$$\Pr[C(\mathsf{Enc}(z, 1)) = 1] - \Pr[C(\mathsf{Enc}(z, 0)) = 1] \geq \frac{1}{\mathsf{poly}(n)},$$

violating the security of Enc. So, $\mathcal{E}$ must be a no-instance of CGL.

Now we show how to obtain a P/poly-secure witness encryption scheme from a half-Levin reduction (R, R_{wit}) to CGL. We define a witness encryption scheme as follows. Let z be an instance of L, and if $z \in L$, let w be a witness. Let a be a secret bit to encrypt.

$\mathsf{Enc}(z, a)$ computes $\mathcal{E} = R(z)$, samples $(x, b) \sim \mathcal{E}$ and then outputs $(x, a \oplus b)$.

$\mathsf{Dec}((x, c), z, w)$ computes $A := R_{wit}(z, w)$ and then outputs $A(x) \oplus c$.

The correctness of $(\mathsf{Enc}, \mathsf{Dec})$ follows from the fact that, for $z \in L$, $R_{wit}(z, w)$ is a witness for $\mathcal{E} = R(z)$ being a yes-instance of CGL: that is, a circuit A of polynomial size such that

$$\Pr_{(x,b)\sim\mathcal{E}}[A(x) = b] = 1.$$

It follows that $A(x) \oplus c = b \oplus (a \oplus b) = a$. The security of Enc follows from Yao's argument that a distinguisher implies a next-bit predictor. Specifically, for $(x, b) \sim \mathcal{E}$, from a circuit of size $s(n)$ distinguishing $\mathsf{Enc}(z, 0) = (x, b)$ from $\mathsf{Enc}(z, 1) = (x, 1 \oplus b)$, we would obtain a circuit of size $O(s(n))$ predicting (with inverse polynomial advantage) b from x. By the honesty of the reduction, we have $O(s(n)) = \mathsf{poly}(|x|)$. This violates the fact that $\mathcal{E}$ is a no-instance of $\mathsf{CGL}^{s_2}_{s_1}$ for every polynomial s_2, a contradiction.

Proof of Theorem 6

To modify the argument above for the case of BPP-secure witness encryption, we note that, in the first direction discussed, the security of WE is only violated if the circuit C can be constructed uniformly in polynomial time. Therefore, we consider BPP-black-box reductions to *search*-CGL. That is, roughly, the reduction is only guaranteed to work when its oracle is a PPT algorithm, and we require the oracle to actually construct a small circuit with inverse polynomial advantage over the queried distribution $\mathcal{E}$ if one exists. For the other direction, we employ the observation that, from a uniform PPT distinguisher algorithm A, one can construct a next-bit predictor P^A uniformly.

Proof of Theorem 7

For the case of GapDistMCSP in Theorem 7, a challenge is that an arbitrary witness encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ may not yield a distribution $\mathcal{E}$ that represents a well-defined function. On one hand, if $z \in L$, then there is guaranteed to be a unique bit b_x such that (x, b_x) is in the support of $\mathcal{E}$, by the correctness of the witness encryption scheme. However, if $z \notin L$, this may not be true. Luckily, since $\mathcal{E}$ is supported over logarithmic-length strings, we can check in polynomial time whether each x has a unique label. If not, we use our assumption of a circuit lower bound for E to output a no-instance of GapDistMCSP. In the other direction, from a (deterministic) half-Levin reduction from SAT to GapDistMCSP, we obtain a circuit lower bound for E by applying the reduction to a trivial no-instance of SAT, as in [29].

Proof of Theorem 8 and corollary

Our proof of NP-hardness of CGL with parameters corresponding to semi-proper learning works within the framework developed in [16]. It differs from Theorem 7.2 in that paper mainly in the "completeness" argument. Referring to the notation in the proof of that theorem and of our Lemma 64, we need to construct a circuit of size not exceeding $O(\theta\lambda)$.

For one thing, in the proof of [16], using DP_k generators to encode the high-complexity truth-tables f_i, the number of input gates to the circuit alone would exceed the desired threshold. Therefore, we use Nisan-Wigderson generators rather than DP_k generators to allow for a short random seed. (A similar idea is used elsewhere in [16].)

More importantly, we apply a recent result of Holmgren and Rothblum showing that "multiselection" of t indices from a string of length $m \in \mathbb{N}$ can be done by a circuit of size at most $O(m + t \cdot \log^3(m))$ [20]. See Lemma 61. So, we can make $\mathsf{poly}(n)$ queries to θ truth-tables f_j, each of length λ, with total circuit size at most $O(\theta \cdot (\lambda + o(\lambda)))$. This allows us to compute values of the amplified functions $\hat{f}_j$ and recover the secret labels b without exceeding the desired size threshold. Note that Uhlig's Theorem as in [16] cannot accommodate $\mathsf{poly}(n)$ queries.

To obtain Corollary 9, which states that semi-proper agnostic learning is NP-hard, we give a simple randomized reduction from CGL to learning that works by empirical estimation of the success probability of the hypothesis produced by the learner. See Lemma 33.

Proof of Theorem 12 and corollaries

To obtain NP-hardness of oracle-CGL, we apply an argument analogous to the first direction of our main equivalence, along with the fact that, for some distribution family $D \in \mathsf{PSAMP}$, witness encryption for NP exists unconditionally (with high probability) with respect to oracles sampled from D. This was demonstrated in [21]. More specifically, the authors show the following.

▶ **Lemma 18** ([21] – informal; see Lemma 41 and definition 40)**.** *For some* NP*-complete language L, distribution $D \in$ PSAMP, and constant $\ell \in \mathbb{N}$, there is a witness encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *for L such that, for all $m \in \mathbb{N}$, with high probability over $O \sim D_m$ (with $O \in \{0,1\}^m$),* $(\mathsf{Enc}^O, \mathsf{Dec}^O)$ *is secure against O-oracle adversaries of size $m^{1/\ell}$.*

Our idea is to apply the foregoing "oracle witness encryption" along with our main technique for proving that witness encryption implies NP-hardness of CGL. We modify the definition of CGL accordingly so that all circuits and samplers have access to an oracle randomly sampled from D. See Definition 25. Crucially, to achieve NP-hardness of D-oracle-CGL$^{s(n)^c}_{s(n)}$ for a constant $c \in \mathbb{N}$, we require witness encryption with security against adversaries of size $s(n)^c$, and thus we require an oracle O with truth-table length only $s(n)^{c\cdot\ell} = \mathsf{poly}(n)$.

Corollary 13 follows from Theorem 12 analogously to Corollary 9 and Theorem 8, as discussed above: a straightforward reduction from oracle-CGL to oracle-PAC learning.

To prove Corollary 14, we give a randomized reduction from D-oracle-CGL to the problem GapRAM-MINLT. Given an instance $\mathcal{E}^{(-)}$ of D-oracle-CGL, the reduction starts by sampling a random truth-table $O \sim D$. Then, it defines an instance $\mathcal{E}'$ of GapRAM-MINLT which always outputs the entire truth-table O along with a string (x, b) sampled according to $\mathcal{E}^O$; one can think of $\mathcal{E}'$ as sampling pairs (x', b) with $x' = (x, O)$. Clearly, this results in a non-oracle instance of GapRAM-MINLT. In the proof of correctness, we also rely on the fact that a program of size and running time at most s' can be simulated by a circuit of size $\mathsf{poly}(s')$ (and vice versa). Since the oracles obtained from [21] require length polynomially greater than the size of adversary they are secure against, we need to consider machines running in time less than the length of their inputs. Given these observations and the definitions of oracle-CGL and GapRAM-MINLT, the corollary follows easily.

Proof of Theorem 15

We prove Theorem 15 with an argument from [2] as a starting point.[4] Let R be a randomized non-adaptive reduction from an NP-complete language L to agnostic learning over PSAMP/poly. Assume that i.o. one-way functions do not exist. Then, for any $D \in$ PSAMP and auxiliary-input function $\{f_z\}_{z\in\{0,1\}^*}$, there is a machine I that distributionally inverts f_z with high probability over auxiliary inputs $z \sim D$. We think of instances z of L as auxiliary inputs for the function f_z that, roughly, simulates $R(z)$ to obtain a sampling circuit $\mathcal{E}$ over string/label pairs and then samples from $\mathcal{E}$. Applebaum et al. show that there is a polynomial-time-computable hypothesis h, using a distributional inverter for f_z, having near-maximum agreement over $\mathcal{E}$. Running the reduction R and answering oracle queries with hypotheses h defined in this way, we obtain

$$\mathsf{DistNP} \subseteq \mathsf{HeurBPP},$$

where "Heur", informally, refers to average-case heuristics that may err by incorrectly outputting 1 or 0.

We next use the fact that the inversion of polynomial-time-computable functions is *verifiable*. That is, if a machine fails to invert on some instance, it can be defined to "realize this" in polynomial time. More concretely, the machine I above can be defined to output $\perp$ on input z if it fails to invert often on some particular f_z, using empirical estimation. From this observation, we further obtain

$$\mathsf{DistNP} \subseteq \mathsf{AvgBPP},$$

[4] Note that we include a proof below for the reader's convenience; see Lemma 72.

where Avg refers to heuristics that do not err but may output $\perp$ rather than 1 or 0 with inverse polynomial probability. Lastly, we apply a recent result of [12], which, from DistNP $\subseteq$ AvgBPP, yields *worst-case* easiness of agnostic PAC learning over distributions in PSAMP/poly. We then run the reduction R *for a second time*, answering oracle queries with the worst-case agnostic learner guaranteed by [12], to obtain NP $\subseteq$ BPP.

Proofs of Theorem 16 and Theorem 17

To obtain Theorem 16, start by assuming the existence of witness encryption for an NP-complete language. Then, apply the characterization given by Theorem 4 to obtain a half-Levin reduction from NP to CGL. Since CGL reduces to agnostic PAC learning, we may apply Theorem 15 as described above to exclude Heuristica and Pessiland. To summarize, informally,

$$\begin{aligned} \text{WE} &\implies \mathsf{NP} \leq_{hl} \mathsf{CGL} && \text{(Theorem 4)} \\ &\implies \mathsf{NP} \leq_{tt} \text{ agnostic PAC-learning over } \mathsf{PSAMP/poly} \\ &\implies (\mathsf{NP} \not\subseteq \mathsf{BPP} \implies \text{i.o. OWF}), && \text{(Theorem 15)} \end{aligned}$$

where $\leq_{hl}$ refers to a half-Levin reduction and $\leq_{tt}$ refers to a randomized non-adaptive (ie. "truth-table") reduction. Furthermore, Garg et al. show that witness encryption for NP, together with the existence of a one-way function, yields public-key encryption [11]. This, together with the above, conditionally yields public-key encryption. To summarize,

$$\begin{aligned} \text{WE and } \mathsf{NP} \not\subseteq \mathsf{BPP} &\implies \text{WE and i.o. OWF} \\ &\implies \text{i.o. PKE} && \text{([11])} \end{aligned}$$

Theorem 17 is immediate from the reasoning above together with another application of the characterization in Theorem 4. That is,

$$\begin{aligned} \mathsf{NP} \leq_{hl} \mathsf{CGL} &\implies \text{WE} && \text{(Theorem 4)} \\ &\implies (\mathsf{NP} \not\subseteq \mathsf{BPP} \implies \text{i.o. PKE}) && \text{(Theorem 16)} \end{aligned}$$

2 Preliminaries

▶ **Definition 19.** *We say that a distribution is* flat *if it is equal to the uniform distribution over its support.*

▶ **Definition 20.** *For a distribution family $D = \{D_n\}_{n\in\mathbb{N}}$, $s : \mathbb{N} \to \mathbb{N}$, and $\delta : \mathbb{N} \to [0,1]$, $\mathsf{SIZE}^{\delta}_{D}[s]$ denotes the class of function families $f = \{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$ such that, for all sufficiently large $n \in \mathbb{N}$, there exists a circuit C_n of size at most $s(n)$ such that*

$$\Pr_{x\sim D_n}[C_n(x) = f_n(x)] \geq \delta(n).$$

2.1 PAC Learning Problems

▶ **Definition 21** (Optimal hypotheses)**.** *For $n \in \mathbb{N}$, a function $s : \mathbb{N} \to \mathbb{N}$, a distribution family $\mathcal{E}_n$ supported over $\{0,1\}^n \times \{0,1\}$, a parameter $\varepsilon > 0$, and a hypothesis $h : \{0,1\}^n \to \{0,1\}$, we say that h is* (s,ε)-optimal *for $\mathcal{E}_n$ if*

$$\Pr_{(x,b)\sim\mathcal{E}_n}[h(x) = b] \geq \max_{f\in\mathsf{SIZE}[s(n)]}\left\{\Pr_{(x,b)\sim\mathcal{E}_n}[f(x) = b]\right\} - \varepsilon.$$

*For $\lambda \in \mathbb{N}$, an oracle function $O : \{0,1\}^\lambda \to \{0,1\}$, and a hypothesis $g^{(-)} : \{0,1\}^n \to \{0,1\}$, we say that g is (s, ε, O)-*optimal *for $\mathcal{E}_n$ if*

$$\Pr_{(x,b)\sim\mathcal{E}_n}[g^O(x) = b] \geq \max_{f \in \mathsf{SIZE}[s(n)]} \left\{ \Pr_{(x,b)\sim\mathcal{E}_n}[f^O(x) = b] \right\} - \varepsilon.$$

▶ **Definition 22** (Agnostic PAC learning circuits)**.** *Consider $s_1, s_2 : \mathbb{N} \to \mathbb{N}$. We say that an algorithm A* agnostically PAC learns $\mathsf{SIZE}[s_1(n)]$ *by* $\mathsf{SIZE}[s_2(n)]$ *if, for every $n \in \mathbb{N}$, $\delta > 0$, $\varepsilon > 0$, and joint distribution family $\mathcal{E} = \{\mathcal{E}_n\}_{n\in\mathbb{N}}$ where each $\mathcal{E}_n$ is supported over $\{0,1\}^n \times \{0,1\}$, the following holds. With probability at least $1 - \delta$, $A^{\mathcal{E}_n}$ outputs an (s_1, ε)-optimal hypothesis h for $\mathcal{E}_n$ of size at most $s_2(n)$.*

A is given the parameters $n, 1/\varepsilon$, and $1/\delta$ in unary.

We say that A agnostically PAC learns $\mathsf{SIZE}[s_1]$ in the improper setting *if the above is true except there is no restriction on the complexity of h (except that it cannot exceed size the run time of A).*

▶ **Definition 23** (Agnostic Oracle-PAC learning circuits)**.** *The definition is analogous to Definition 22, except we also consider distribution families $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ for sampling oracle truth-tables, where D_λ is supported over $\{0,1\}^{2^\lambda}$ and uniformly samplable in time* $\mathsf{poly}(2^\lambda)$*. We say that an algorithm A* agnostically oracle-PAC learns $\mathsf{SIZE}[s_1]$ *by* $\mathsf{SIZE}[s_2]$ *if, for any such D, for every $\varepsilon, \delta > 0$ and $n, \lambda \in \mathbb{N}$, the following holds with probability at least $1 - \delta$ over $O \sim D_\lambda$.*

For any joint distribution family $\mathcal{E} = \{\mathcal{E}_n\}_{n\in\mathbb{N}}$ where each $\mathcal{E}_n$ is supported over $\{0,1\}^n \times \{0,1\}$, with probability at least $1 - \delta$ over the randomness of A, $A^{\mathcal{E}_n}(\mathsf{tt}(O))$ outputs an (s_1, ε, O)-optimal hypothesis h for $\mathcal{E}_n$ of circuit size at most $s_2(n)$.

A is also given the parameters $n, 1/\varepsilon$, and $1/\delta$ in unary.

Applebaum et al. define the following decision version of the PAC learning problem for circuits.

▶ **Definition 24** (Computational gap-learning problem (CGL) [2])**.** *The* computational gap-learning *problem with size parameters $s_1, s_2 : \mathbb{N} \to \mathbb{N}$ and security parameter $\varepsilon : \mathbb{N} \to [0,1]$, denoted $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, is the promise problem defined as follows.*[5] *The input consists of a distribution $\mathcal{E}$ over pairs $(x, b) \in \{0,1\}^n \times \{0,1\}$ represented as a circuit of size* $\mathsf{poly}(n)$*.*[6]

- *$\mathcal{E}$ is a yes-instance if there exists a circuit C of size at most $s_1(n)$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1.$$

- *$\mathcal{E}$ is a no-instance if for every circuit C of size at most $s_2(n)$,*

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < \frac{1}{2} + \varepsilon(n).$$

▶ **Definition 25** (Oracle computational gap-learning problem (oracle-CGL))**.** *Let $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ be a polynomial-time-samplable distribution family such that each D_λ is supported over truth-tables of oracle functions $O_\lambda : \{0,1\}^\lambda \to \{0,1\}$. The D-*oracle computational gap-learning *problem with size parameters $s_1, s_2 : \mathbb{N} \to \mathbb{N}$, security parameter $\varepsilon : \mathbb{N} \to [0,1]$, and oracle*

[5] We note that [2] define CGL for a given completeness parameter $\alpha : \mathbb{N} \to [0,1]$ (ie. a yes-instance has C such that $\Pr[C(x) = b] \geq \alpha(n)$). In this work, we only consider $\alpha = 1$.

[6] In this work, we abuse notation by taking $\mathcal{E}$ to refer both to the distribution and the circuit sampling it.

parameter $\lambda : \mathbb{N} \to \mathbb{N}$*, denoted* D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$*, is defined analogously to* $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ *except all circuits (including the sampler* $\mathcal{E}$*) have access to an oracle randomly drawn from* $D_{\lambda(n)}$*. More specifically,*

- $\mathcal{E}^{(-)}$ *is a yes-instance if there exists an oracle circuit* $C^{(-)}$ *of size at most* $s_1(n)$ *such that, for every oracle* O *in the support of* $D_{\lambda(n)}$*,*

$$\Pr_{(x,b)\sim\mathcal{E}^O}\left[C^O(x) = b\right] = 1.$$

- $\mathcal{E}^{(-)}$ *is a no-instance if, with probability at least* $1-\varepsilon(n)$ *over* $O \sim D_{\lambda(n)}$*, for every oracle circuit* $C^{(-)}$ *of size at most* $s_2(n)$*,*

$$\Pr_{(x,b)\sim\mathcal{E}^O}\left[C^O(x) = b\right] < \frac{1}{2} + \varepsilon(n).$$

▶ **Definition 26** ($\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT)**.** GapRAM-MINLT *with gap parameter* $g : \mathbb{N} \to \mathbb{N}$ *and soundness parameter* $\varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$*, denoted* $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT*, is the decision problem defined as follows. The input consists of a distribution* $\mathcal{E}$ *over pairs* $(x, b) \in \{0,1\}^n \times \{0,1\}$ *represented as a circuit of size* $\mathsf{poly}(n)$*. The input also includes a threshold parameter* $s \in \mathbb{N}$ *and a running time parameter* $t \in \mathbb{N}$ *represented in unary.*

- $(\mathcal{E}, 1^s, 1^t)$ *is a yes-instance of* $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT *if there exists a* t*-time program* M *of size at most* s *such that* $M^x = b$ *for all* (x, b) *in the support of* $\mathcal{E}$*.*
- $(\mathcal{E}, 1^s, 1^t)$ *is a no-instance of* $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT *if for every* $g(t)$*-time program* M *of size at most* $g(s)$*,*

$$\Pr_{(x,b)\sim\mathcal{E}}[M^x = b] < \frac{1}{2} + \varepsilon(n)$$

2.2 Reductions to Learning

Here we define a form of reduction that is intermediate between the standard notions of a Karp reduction and a Levin reduction.

▶ **Definition 27** (Half-Levin reduction)**.** *Consider languages* $L_1, L_2 \in \mathsf{NP}$ *with witness relations* V_{L_1}, V_{L_2}*. A* half-Levin reduction *from* L_1 *to* L_2 *consists of polynomial-time machines* R *and* R_{wit} *with the following properties.*

- *For any* L_1*-instance* x*,* $x \in L_1 \iff R(x) \in L_2$*;*
- *For any* $(x, w) \in V_{L_1}$*, it holds that* $(R(x), R_{wit}(x, w)) \in V_{L_2}$*.*

A number of the results in this paper involve half-Levin reductions to CGL, which belongs to (a promise version of) MA but not necessarily NP. Note that the definition of a half-Levin reduction can be extended to CGL in the natural way. Specifically, in a half-Levin reduction from $L \in \mathsf{NP}$ to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, the machine R_{wit}, given some $z \in L$ and witness w, is required to produce a circuit C of size at most s_1 such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1,$$

where $\mathcal{E}$ is the output of $R(z)$. In addition, we say that a reduction to CGL is *honest* if, on an input z of length n, it outputs a distribution $\mathcal{E}$ supported over $\{0,1\}^{n^{\Omega(1)}} \times \{0,1\}$.

▶ **Definition 28** (Reduction to agnostic PAC learning circuits [2])**.** *For polynomials* s_1 *and* s_2 *and* $\delta : \mathbb{N} \to [0,1]$*, a reduction from a language* L *to agnostic PAC learning* $\mathsf{SIZE}[s_1]$ *by* $\mathsf{SIZE}[s_2]$ *over* $\mathsf{PSAMP}/\mathsf{poly}$ *with failure probability* δ *consists of a PPT machine* R*, which on*

input $z \in \{0,1\}^n$, makes queries that are joint distributions $\mathcal{E}^{(i)}$ over $\{0,1\}^{m(n)} \times \{0,1\}$ for some $m : \mathbb{N} \to \mathbb{N}$, each represented by a sampling circuit. The oracle provides a hypothesis h_i, represented as a poly(n)*-size circuit, as a response to each query. Based on the hypotheses h_i and its own computation, R accepts or rejects the input z.*

It is guaranteed that, for some inverse polynomial ε, if the oracle returns an (s_1, ε)-optimal hypothesis h of size $s_2(m(n))$ for each query $\mathcal{E}^{(i)}$, then with probability at least $1 - \delta(n)$, $R(z) = L(z)$.

If the failure probability is omitted, we take it to be $1/3$.

If $m(n) \geq n^{\Omega(1)}$, we call the reduction honest.

▶ **Definition 29** (Reduction to agnostic oracle-PAC learning circuits)**.** *We define reductions to agnostic oracle-PAC learning analogously with Definition 28. Specifically, the reduction samples random $O^{(i)} \sim D_{\lambda(n)}$ and then queries pairs $(\mathcal{E}^{(i)}, \mathsf{tt}(O^{(i)}))$ to its oracle. The guarantee of the reduction holds if the oracle responds with an $(s_1, \varepsilon, O^{(i)})$-optimal hypothesis of size at most $s_2(m(n))$ for each query $\mathcal{E}^{(i)}$.*

For our results concerning BPP-secure witness encryption, we will require non-black-box notions of reduction. We start by recalling the standard definition of a class-specific black-box reduction.

▶ **Definition 30** (Class-specific black-box reduction [13])**.** *For a complexity class C and languages L_1 and L_2, a PPT oracle machine R is a C-*black-box reduction *from L_1 to L_2 if, for any oracle $\mathcal{O} \in C$ deciding L_2, $R^{\mathcal{O}}$ decides L_1.*

We will actually require a slightly less strict kind of reduction: rather than insisting $L_2 \in C$ everywhere, to be correct on a given input z, the reduction will only require that an oracle $\mathcal{O} \in C$ agree with L_2 on the queries actually made given that input z. For our purposes, we only need to consider the case of a reduction that generates its queries deterministically.

▶ **Definition 31** (Instance-wise class-specific black-box reduction)**.** *Consider a complexity class C, languages L_1 and L_2, and a non-adaptive oracle machine R that generates its queries deterministically but may use randomness elsewhere. R is an* instance-wise C-black-box reduction *from L_1 to L_2 if the following holds. For any oracle $\mathcal{O} \in C$ and instance z of L_1, if $q_1, ..., q_m$ are the oracle queries made by R on input z and $\mathcal{O}(q_i) = L_2(q_i)$ for all $i \in [m]$, then $R^{\mathcal{O}}(z)$ correctly decides $L_1(z)$.*

We now explicitly define our required notion of an instance-wise BPP-black-box half-Levin reduction to the search version of CGL. Note that we may drop the qualifier "instance-wise" elsewhere in the presentation. Also note that the definition is extended from those above to the case of a reduction to a search problem.

▶ **Definition 32** (Instance-wise BPP-black-box half-Levin reduction to search-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$)**.** *For polynomials s_1 and s_2 and a language L, an* instance-wise BPP-black-box half-Levin reduction from L to search-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ *is an oracle machine R as follows. On an input $z \in \{0,1\}^*$, R deterministically makes one query, $\mathcal{E}_z$: a joint distribution supported on $\{0,1\}^{m(|z|)} \times \{0,1\}$ for some polynomial m, represented by a sampling circuit.*

(Instance-wise BPP-black-box) *Consider an input $z \in \{0,1\}^n$ for some $n \in \mathbb{N}$, and suppose the oracle A is a PPT algorithm meeting the following condition:*

> *Given the query $(z, \mathcal{E}_z)$ made by R on input z, if $\mathcal{E}_z$ is not a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, then with probability at least $1 - 2^{-\Omega(n)}$ over its internal randomness, A returns a hypothesis h of size at most $s_2(m(n))$ such that*
>
> $$\Pr_{(x,b)\sim\mathcal{E}_z}[h(x) = b] \geq \frac{1}{2} + \frac{\varepsilon(m(n))}{2}.$$

Then, the following are guaranteed for such z.

- *if $z \in L$, $\mathcal{E}_z$ is a yes-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, and $R^A(z)$ accepts with probability $1 - 2^{-\Omega(n)}$;*
- *if $z \notin L$, $\mathcal{E}_z$ is a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, and $R^A(z)$ rejects with probability at least $n^{-O(1)}$.*

(Half-Levin) *Further, there exists a polynomial-time computable function R_{wit} such that, for any input $z' \in \{0,1\}^*$, if z' is a yes-instance of L with witness w, then $R_{wit}(z', w)$ outputs a witness for $\mathcal{E}_{z'} \in \mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$: that is, a circuit C of size at most $s_1(m(n))$ such that $\Pr_{(x,b)\sim\mathcal{E}_{z'}}[C(x) = b] = 1$.*

The lemma below shows that a reduction to CGL implies a reduction to agnostic PAC learning as in Definition 28.

▶ **Lemma 33.** *For a language L, $\delta : \mathbb{N} \to [0,1]$, and polynomials $s_1, s_2, \varepsilon^{-1}$ with $\varepsilon(n) \leq 1/8$ for $n \in \mathbb{N}$, suppose there is a randomized many-one reduction from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ with failure probability δ. Then, there is a randomized poly-time reduction from L to agnostic PAC learning $\mathsf{SIZE}[s_1]$ by $\mathsf{SIZE}[s_2]$ with failure probability δ', where $\delta'(n) := \delta(n) + 2^{-\Omega(n)}$ for $n \in \mathbb{N}$.*

Proof. Let R be the assumed reduction to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$. Define a reduction R' as follows:

> On input an instance $z \in \{0,1\}^n$ of L and a choice of randomness r, simulate $R(z, r)$ to produce a polynomial-size sampling circuit $\mathcal{E}$ supported over $\{0,1\}^m \times \{0,1\}$. Query $\mathcal{E}$ to the oracle, obtaining a hypothesis h in return. If the size of h is greater than $s_2(m)$, reject. By sampling $\mathsf{poly}(\varepsilon^{-1}(m) \cdot n)$ times from $\mathcal{E}$, obtain an empirical estimate of $\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b]$. If the value obtained is less than $1 - 2 \cdot \varepsilon(m)$, reject. Otherwise, accept.

We will show that R' constitutes a reduction to agnostic PAC learning $\mathsf{SIZE}[s_1]$ by $\mathsf{SIZE}[s_2]$.

Consider an L-instance z. First suppose $z \in L$. Then, with probability at least $1 - \delta(n)$ over r, $\mathcal{E}$ is a yes-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$. In this case,

$$\max_{C \in \mathsf{SIZE}[s_1(m)]} \left\{ \Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] \right\} = 1.$$

Assume that the oracle A returns an $(s_1, \varepsilon(m))$-optimal hypothesis h of size $s_2(m)$: that is,

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \geq 1 - \varepsilon(m).$$

Then, by a Chernoff bound, with probability at least $1 - 2^{-\Omega(n)}$, R' accepts. Overall, for $z \in L$, R' accepts with probability greater than $1 - (\delta(n) + 2^{-\Omega(n)})$.

Now suppose z is a no-instance of L. In this case, with probability at least $1 - \delta(n)$ over r, $\mathcal{E}$ is a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$: that is, for every circuit C of size at most $s_2(m)$,

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < \frac{1}{2} + \varepsilon(m).$$

Thus, A *cannot* return a hypothesis $h \in \mathsf{SIZE}[s_2(m)]$ with success better than $(1/2) + \varepsilon(m) < 1 - 2 \cdot \varepsilon(m)$ over $\mathcal{E}$. So R' rejects with probability at least $1 - 2^{-\Omega(n)}$. Overall, for $z \notin L$, R' accepts with probability at most $\delta(n) + 2^{-\Omega(n)}$, as desired. ◀

By a very similar argument, we obtain the following for the oracle-PAC setting. We omit the proof here.

▶ **Lemma 34.** *For a language L, $\delta : \mathbb{N} \to [0,1]$, $D \in \mathsf{PSAMP}$, $\lambda : \mathbb{N} \to \mathbb{N}$ with $\lambda(n) = O(\log n)$ for $n \in \mathbb{N}$, and polynomials $s_1, s_2, \varepsilon^{-1}$ with $\varepsilon(n) \leq 1/8$ for $n \in \mathbb{N}$, suppose there is a randomized many-one reduction from L to D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ with failure probability δ. Then, there is a randomized poly-time reduction from L to agnostic oracle-PAC learning $\mathsf{SIZE}[s_1]$ by $\mathsf{SIZE}[s_2]$ with failure probability δ', where $\delta'(n) := \delta(n) + \varepsilon(n) + 2^{-\Omega(n)}$ for $n \in \mathbb{N}$.*

2.3 Meta-complexity

The following is a natural generalization of MCSP to the distributional version, where in addition to the truth table of a function $f : \{0,1\}^n \to \{0,1\}$, one is also given an explicit distribution μ over $\{0,1\}^n$ (as a table of probabilities), and is asked to distinguish "easy" f's that can be well approximated by small boolean circuits from "hard" f's that are almost impossible to approximate even by much larger boolean circuits.

▶ **Definition 35** (GapDistMCSP)**.** *The* gap Distributional Minimum Circuit Size problem *with size parameters $s_1, s_2 : \mathbb{N} \to \mathbb{N}$ and advantage parameter $\varepsilon : \mathbb{N} \to [0,1]$, denoted $\mathsf{GapDistMCSP}^{s_2}_{s_1}[\varepsilon]$, is the decision problem defined as follows. The input consists of the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ (represented as a truth-table of length 2^n bits) and a probability distribution $\mu : \{0,1\}^n \to [0,1]$ (represented as a table of length $2^{O(n)}$ bits).*

- *(f, μ) is a yes-instance if there exists a circuit C of size at most $s_1(n)$ such that*

$$\Pr_{x \sim \mu}[C(x) = f(x)] = 1.$$

- *(f, μ) is a no-instance if for every circuit C of size at most $s_2(n)$,*

$$\Pr_{x \sim \mu}[C(x) = f(x)] < \frac{1}{2} + \varepsilon(n).$$

We note that, for an appropriate choice of the approximation parameters and distribution μ, GapDistMCSP generalizes other variants of MCSP, e.g., MCSP^* of [16] and an average-case version AveMCSP of [45].

2.4 Secret Sharing

See [4] for a survey.

▶ **Definition 36** (Access Structures Represented by Monotone Formulae)**.** *An* access structure *is a monotone collection $\mathcal{A} \subseteq 2^{[n]}$ of non-empty subsets of $[n]$. For $T \subseteq [n]$, we say that T is* authorized *if it belongs to $\mathcal{A}$ and* unauthorized *otherwise. We say that a monotone formula $\varphi : \{0,1\}^n \to \{0,1\}$* represents *$\mathcal{A}$ if $\varphi(\chi_T) = 1$ exactly when $T \in \mathcal{A}$, where χ_T is the characteristic vector of T.*

▶ **Definition 37** (Efficient Secret Sharing Scheme)**.** *An efficient secret sharing scheme for an access structure $\mathcal{A} \subseteq 2^{[n]}$ represented by a monotone formula φ consists of a pair of poly-time algorithms Share and Rec with the following properties. Share takes three inputs; the first is the formula $\varphi : \{0,1\}^n \to \{0,1\}$, the second is a "secret" $b \in \{0,1\}$, and the third is a choice of "randomness" $r \in \{0,1\}^{|\varphi|}$. Rec also takes three inputs: the formula φ, a subset of "shares" $v_T = (v_i)_{i \in T} \in \left(\{0,1\}^{|\varphi|}\right)^{|T|}$, and the set $T \subseteq [n]$.*

- ***Correctness:*** *Authorized parties can recover the secret. Consider any $b \in \{0,1\}$ and $r \in \{0,1\}^{|\varphi|}$, let $(v_1, ..., v_n)$ be the output of $Share(\varphi, b; r)$, and let T be an authorized set. Then $Rec(\varphi, v_T, T) = b$.*

- ***Privacy:*** *Unauthorized parties cannot learn anything about the secret, in an information theoretic sense. Let T be an unauthorized set. Then, for $b \sim \{0,1\}$ and $r \sim \{0,1\}^{|\varphi|}$, the random variable $Share(\varphi, b; r)_T$ is statistically independent from the random variable b.*

▶ **Lemma 38** ([27, 5]). *For every access structure $\mathcal{A}$, there exists an efficient secret sharing scheme as in Definition 37. The circuit size required to simulate Rec is at most $O(|v_T|^2)$.*

Proof sketch. We will describe the construction given in [5]. Let $\varphi : \{0,1\}^n \to \{0,1\}$ be a monotone formula representing $\mathcal{A}$. Starting at the root of the tree representation of φ, the *Share* algorithm assigns values to the nodes in a recursive manner as follows. For a secret $b \in \{0,1\}$, assign b to the root, and instantiate all shares v_i for $i \in [n]$ to be empty. Then:

- If the current gate is $\vee$ and the current value is s, assign s to both children.
- if the current gate is $\wedge$ and the current value is s, randomly choose $t \sim \{0,1\}$. Assign t to one child and $t \oplus s$ to the other.
- If the current gate is a variable $i \in [n]$ and the current value is s, include s in the share v_i.

Note that the amount of randomness required is at most $|\varphi|$. The *Rec* algorithm proceeds from the leaves of the tree back to the root. Start by assigning shares $(v_i)_{i \in T}$ to leaves using the set T. Then:

- If the current gate is $\vee$ and both children have value s, assign s to the parent.
- If the current gate is $\wedge$ and the children have values s and t, assign $t \oplus s$ to the parent.
- If the current gate is the root, return the value assigned. ◀

2.5 Witness Encryption

▶ **Definition 39** (Witness Encryption [11]). *Consider a language $L \in \mathsf{NP}$ with witness relation R_L, a class of function families Γ, a security function $\varepsilon : \mathbb{N} \to [0,1]$, and a length function $\ell : \mathbb{N} \to \mathbb{N}$. A (Γ, ε)-*secure *ℓ-*length witness encryption scheme for L *consists of polynomial-time algorithms* Enc *and* Dec *as follows.*

- *Given an L-instance $x \in \{0,1\}^n$, a message bit $b \in \{0,1\}$, and randomness $r \sim \mathcal{U}_{\mathsf{poly}(n)}$, $\mathsf{Enc}(x, b; r)$ outputs a ciphertext c of length at most $\ell(n)$.*
- *Given a ciphertext c and an instance/witness pair $(x, w) \in R$, $\mathsf{Dec}(c, x, w)$ deterministically outputs a bit $b \in \{0,1\}$.*

Moreover, Enc *and* Dec *have the properties below.*

- ***Correctness:*** *For all sufficiently large $n \in \mathbb{N}$, any $b \in \{0,1\}$, $x \in \{0,1\}^n \cap L$, and w such that $(x, w) \in R_L$,*

$$\Pr_r \left[\mathsf{Dec}(\mathsf{Enc}(x, b; r), x, w) = b\right] = 1.$$

- ***Security:*** *For any $A = \{A_n\}_{n \in \mathbb{N}} \in \Gamma$, sufficiently large $n \in \mathbb{N}$, and $x \in \{0,1\}^n \backslash L$,*

$$|\Pr[A_n(x, \mathsf{Enc}(x, 1; r)) = 1] - \Pr[A_n(x, \mathsf{Enc}(x, 0; r)) = 1]| < \varepsilon(n).$$

If ε is omitted, we assume it is some negligible function. If ℓ is omitted, we allow the output length of Enc *to be any polynomial.*

▶ **Definition 40** (Witness Encryption in Oracle Worlds [21]). *Consider a language $L \in \mathsf{NP}$ with witness relation R_L, a class of function families Γ, a security function $\varepsilon : \mathbb{N} \to [0,1]$, and a family of distributions $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$, where each D_λ is supported over $\{0,1\}^{2^\lambda}$. A (Γ, ε)-*secure witness encryption scheme for L with respect to D *is defined as in Definition 39, except the algorithms* Enc *and* Dec *both take an oracle, and they satisfy the following properties.*

- ***Correctness:*** *For all sufficiently large $n, \lambda \in \mathbb{N}$, O in the support of D_λ, $b \in \{0,1\}$, $x \in \{0,1\}^n \cap L$, and w such that $(x,w) \in R_L$,*

$$\Pr_r\left[\mathsf{Dec}^O(1^\lambda, \mathsf{Enc}^O(1^\lambda, x, b; r), x, w) = b\right] = 1.$$

- ***Security:*** *For all sufficiently large $n, \lambda \in \mathbb{N}$, with probability $1 - \varepsilon(\lambda)$ over $O \sim D_\lambda$, for any $A = \{A_{n,\lambda}\}_{n,\lambda\in\mathbb{N}} \in \Gamma$ and any $x \in \{0,1\}^n \backslash L$,*

$$\left|\Pr_r[A^O_{n,\lambda}(\mathsf{Enc}^O(1^\lambda, x, 1; r)) = 1] - \Pr_r[A^O_{n,\lambda}(\mathsf{Enc}^O(1^\lambda, x, 0; r)) = 1]\right| < \varepsilon(\lambda).$$

The following result from Huang, Ilango, and Ren was originally stated with security against oracle *programs* of polynomial size and query complexity; we observe that the same holds for polynomial-size circuits.

▶ **Lemma 41** ([21]). *There exist an* NP*-complete language L, a constant $\ell \in \mathbb{N}$, and a family of distributions $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$, where each D_λ is supported over $\{0,1\}^{2^\lambda}$ and samplable in time* $\mathsf{poly}(2^\lambda)$*, such that there is a* $(\mathsf{SIZE}[2^{\lambda/\ell}], 2^{-\lambda/\ell})$*-secure witness encryption scheme for L with respect to D.*

2.6 Hardness of Approximation

▶ **Definition 42** (MMSA). *For $g, \theta, \Delta : \mathbb{N} \to \mathbb{N}$,* $\mathsf{Gap}_g\mathsf{MMSA}^\Delta_\theta$ *denotes the following promise problem.*

The input consists of a monotone formula φ over variables $[n]$ and of length at most $\Delta(n)$ as a binary string. Let $\mathsf{MMSA}(\varphi)$ *denote* $\min\left\{ \sum_{i\in[n]} \alpha(i) \;\middle|\; \alpha : [n] \to \{0,1\} \text{ satisfies } \varphi \right\}$.

- $\Pi_Y = \{\varphi \mid \mathsf{MMSA}(\varphi) \le \theta(n)\}$
- $\Pi_N = \{\varphi \mid \mathsf{MMSA}(\varphi) > g(n) \cdot \theta(n)\}$

▶ **Lemma 43** ([8, 9, 16]). *There exist $g, \theta, \Delta : \mathbb{N} \to \mathbb{N}$ with $g(n) = n^{(\log\log n)^{-O(1)}}$, $\theta(n) \le n$, and Δ a polynomial with $\Delta(n) \ge n$ for $n \in \mathbb{N}$ as follows. For every language $L \in$ NP, there is a half-Levin reduction from L to* $\mathsf{Gap}_g\mathsf{MMSA}^\Delta_\theta$.

▶ **Definition 44** (CMMSA [16]). *For $g, \varepsilon^{-1}, \theta, \Delta : \mathbb{N} \to \mathbb{N}$,* $\mathsf{Gap}_{g,\varepsilon}\mathsf{CMMSA}^\Delta_\theta$ *denotes the following promise problem. The input consists of a collection $\Phi = \{\varphi_1, ..., \varphi_m\}$ of monotone formulas over variables $[n]$, where each φ_i has length at most $\Delta(n)$ as a binary string, along with a weight function $w : [n] \to \mathbb{N}$. For an assignment $\alpha : [n] \to \{0,1\}$, let $w(\alpha)$ denote $\sum_{i\in[n]} \alpha(i) \cdot w(i)$.*

- $\Pi_Y = \{(\Phi, w) \mid$ *there exists $\alpha : [n] \to \{0,1\}$ such that $w(\alpha) \le \theta(n)$ and $\Pr_{\varphi\sim\Phi}[\varphi(\alpha) = 1] = 1$*$\}$
- $\Pi_N = \{(\Phi, w) \mid$ *for all $\alpha : [n] \to \{0,1\}$ such that $w(\alpha) \le \theta(n) \cdot g(n)$, $\Pr_{\varphi\sim\Phi}[\varphi(\alpha) = 1] < \varepsilon$*$\}$

▶ **Lemma 45** ([9, 16]). *There exists a constant $\gamma > 0$ and a function $\theta : \mathbb{N} \to \mathbb{N}$ such that for all $\Delta : \mathbb{N} \to \mathbb{N}$ with $\Delta(n) = O(\log n)$, for every language $L \in$ NP, there is a half-Levin reduction from L to* $\mathsf{Gap}_{\Delta^\gamma, \Delta^{-\gamma}}\mathsf{CMMSA}^\Delta_\theta$.

2.7 One-way functions and Inversion

▶ **Definition 46** (δ-invertible (auxiliary-input) functions). *For $n, m \in \mathbb{N}$, a function $f : \{0,1\}^n \to \{0,1\}^m$, a PPT algorithm A, and $\delta : \mathbb{N} \to [0,1]$, we say that A is a* δ-inverter *for f if*

$$\Pr_{x\sim\mathcal{U}_n}[A(1^n, f(x)) \in f^{-1}(f(x))] \ge 1 - \delta(n).$$

In the case of an auxiliary-input function $f_z : \{0,1\}^\ell \to \{0,1\}^m$*, where* $\ell, m \in \mathbb{N}$*, we say that* A *is a* δ-inverter *for* f_z *if*

$$\Pr_{x \sim \mathcal{U}_n}[A(z, f_z(x)) \in f_z^{-1}(f_z(x))] \geq 1 - \delta(|z|).$$

▶ **Definition 47** (Infinitely-often one-way function)**.** *A polynomial-time computable function family* $f = \{f_n : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{n \in \mathbb{N}}$ *is an* infinitely-often one-way function *if, for every PPT algorithm* A*, there exists an inverse polynomial* δ *such that, for infinitely many* $n \in \mathbb{N}$*,* A *is not a* δ-inverter *for* f_n*.*[7]

▶ **Definition 48** (Auxiliary-input one-way function)**.** *A polynomial-time computable auxiliary-input function family* $f = \{f_z : \{0,1\}^{\ell(|z|)} \to \{0,1\}^{m(|z|)}\}_{z \in \{0,1\}^*}$ *is an* auxiliary-input one-way function *if, for every PPT algorithm* A*, there exists an inverse polynomial* δ *such that, for infinitely many* $z \in \{0,1\}^*$*,* A *is not a* δ-inverter *for* f_z*.*

▶ **Definition 49** (Statistical Distance)**.** *The* statistical distance *between two probability distributions* $D^{(1)}$ *and* $D^{(2)}$ *over domain* $\{0,1\}^n$*, denoted* $\Delta(D^{(1)}, D^{(2)})$*, is defined as*

$$\frac{1}{2} \sum_{x \in \{0,1\}^n} \left| D^{(1)}(x) - D^{(2)}(x) \right|,$$

or equivalently,

$$\max_{T \subseteq \{0,1\}^n} \left| \Pr_{x \sim D^{(1)}}[x \in T] - \Pr_{x \sim D^{(2)}}[x \in T] \right|.$$

▶ **Definition 50** (Distributionally invertible auxiliary-input functions)**.** *Consider an auxiliary input function* f_z *computable uniformly in polynomial time given* $z \in \{0,1\}^n$ *and an input* $x \in \{0,1\}^{\ell(n)}$*. The function* f_z *is said to be* distributionally invertible *if for every constant* $b > 0$ *there is a PPT algorithm* I *such that*

$$\Delta((x, f_z(x)), (I(z, f_z(x)), f_z(x))) \leq 1/n^b,$$

where $x \sim \mathcal{U}_{\ell(n)}$*. We refer to the machine* I *as an* n^{-b}-distributional inverter *for* f_z*.*

▶ **Lemma 51** (Strong inversion implies distributional inversion [25])**.** *For every polynomial-time computable auxiliary-input function* f_z *and constant* $b \in \mathbb{N}$*, there is a constant* $c \in \mathbb{N}$ *such that the following holds. If there exists an* n^{-c}*-inverter for* f_z*, then there exists an* n^{-b}*-distributional inverter for* f_z*.*

3 Proofs of Main Results

3.1 Connections between WE and learning

3.1.1 Proof of Theorem 4

▶ **Lemma 52.** *Consider a language* $L \in \mathsf{NP}$*. For* $s_2, \varepsilon^{-1}, \ell : \mathbb{N} \to \mathbb{N}$*, suppose a* $(\mathsf{SIZE}[s_2], \varepsilon)$*-secure* ℓ*-length witness encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *exists for* L*. Let the polynomial* s_1 *be such that* $s_1(n)$ *upper bounds the running times of* Enc *and* Dec *on* L*-instances of length* n*. Let* $s_1'(\ell(n)) = s_1(n)^2$*,* $s_2'(\ell(n)) = s_2(n)$*, and* $2 \cdot \varepsilon'(\ell(n)) = \varepsilon(n)$*.*

Then, there is a half-Levin reduction from L *to* $\mathsf{CGL}^{s_2'}_{s_1'}[\varepsilon']$ *mapping* L*-instances of length* n *to distributions supported over* $\{0,1\}^{\ell(n)} \times \{0,1\}$*.*

[7] It is known that a weak one-way function is equivalent to a standard (strong) one-way function [47], so we will not distinguish between the two in this work.

Proof. Given an L-instance $z \in \{0,1\}^n$, the reduction to $\mathsf{CGL}^{s_2'}_{s_1'}[\varepsilon']$ will output a distribution $\mathcal{E}$ as follows.

Sample $b \sim \mathcal{U}$. Let $x := \mathsf{Enc}(z,b) \in \{0,1\}^{\ell(n)}$. Output (x,b).

First consider the case that $z \in L$. Let w be a witness for z. Let C be the circuit that, with z and w hard-wired, given x as input, outputs $b := \mathsf{Dec}(x,z,w)$. Note that the size of C is less than $s_1(n)^2 = s_1'(\ell(n))$. Moreover, since C is efficiently computable given z and w, the reduction is a half-Levin reduction. By the correctness of the witness encryption scheme, we have

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1.$$

Thus, $\mathcal{E}$ is a yes-instance of $\mathsf{CGL}^{s_2'}_{s_1'}[\varepsilon']$.

Now consider the case that $z \notin L$. Toward a contradiction, suppose there exists a circuit C of size at most $s_2'(\ell(n)) = s_2(n)$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] \geq \frac{1}{2} + \varepsilon'(\ell(n)) = \frac{1}{2} + \frac{\varepsilon(n)}{2}.$$

Then,

$$\begin{aligned}\frac{1}{2} + \frac{\varepsilon(n)}{2} &\leq \frac{1}{2} \cdot \Pr_{\mathsf{Enc},b}[C(x) = 0 \mid b = 0] + \frac{1}{2} \cdot \Pr_{\mathsf{Enc},b}[C(x) = 1 \mid b = 1] \\ &= \frac{1}{2} + \frac{1}{2} \cdot \left(\Pr[C(x) = 1 \mid b = 1] - \Pr[C(x) = 1 \mid b = 0]\right),\end{aligned}$$

which implies that

$$|\Pr[C(\mathsf{Enc}(z,1)) = 1] - \Pr[C(\mathsf{Enc}(z,0)) = 1]| \geq \varepsilon(n),$$

contradicting the security of the witness encryption scheme. Thus, $\mathcal{E}$ is a no-instance of $\mathsf{CGL}^{s_2'}_{s_1'}[\varepsilon']$.

This completes the proof of the lemma. ◀

Observe that a $\mathsf{P/poly}$-secure encryption scheme is $(\mathsf{SIZE}[s],\varepsilon)$-secure for all polynomials s and ε^{-1}. Thus, we obtain the corollary below, which gives the first direction of Theorem 4.

▶ **Corollary 53.** *Consider any language $L \in \mathsf{NP}$, and suppose a $\mathsf{P/poly}$-secure witness encryption scheme $(\mathsf{Enc},\mathsf{Dec})$ exists for L. Then, there exist a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2, ε^{-1}, (R, R_{wit}) is an honest half-Levin reduction from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$.*

▶ **Lemma 54.** *Consider any polynomials $s_1, m : \mathbb{N} \to \mathbb{N}$, any $s_2, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, and any language $L \in \mathsf{NP}$. Suppose there is a half-Levin reduction (R, R_{wit}) from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, where on L-instances of length $n \in \mathbb{N}$, R outputs distributions supported over $\{0,1\}^{m(n)} \times \{0,1\}$ in time $t_R(n)$, and R_{wit} runs in time $t_{R_{wit}}(n)$.*

Then, there is a $(\mathsf{SIZE}[s_2'],\varepsilon')$-secure $(m(n)+1)$-length witness encryption scheme for L, where $s_2'(n) = s_2(m(n))/2$, $\varepsilon'(n) = 2 \cdot \varepsilon(m(n))$, Enc runs in time $O(t_R(n))$, and Dec runs in time $O(t_{R_{wit}}(n))$.

Proof. We define a witness encryption scheme for L as follows.

For $z \in \{0,1\}^n$ and $a \in \{0,1\}$, $\mathsf{Enc}(z,a)$ computes $\mathcal{E} = R(z)$, samples $(x,b) \sim \mathcal{E}$ and then outputs $(x, a \oplus b)$.

$\mathsf{Dec}((x,c),z,w)$ computes $A := R_{wit}(z,w)$ and then outputs $A(x) \oplus c$.

We first show the correctness of $(\mathsf{Enc}, \mathsf{Dec})$. Suppose z is a yes-instance of L with witness w. Then, by the correctness of the half-Levin reduction, $A = R_{wit}(z,w)$ is a witness for $\mathcal{E} = R(z) \in \mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$; that is, A is a circuit of size at most $s_1(m(n))$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[A(x) = b] = 1.$$

Then, for any $a \in \{0,1\}$ and any random choice of (x,b) made by Enc, it holds that $A(x) = b$, so $\mathsf{Dec}((x, a \oplus b), z, w)$ correctly outputs $a = b \oplus (a \oplus b)$.

We now argue for security. Suppose z is a no-instance of L. Then $\mathcal{E} = R(z)$ is a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$; that is, for any circuit A of size at most $s_2(m(n))$,

$$\Pr_{(x,b)\sim\mathcal{E}}[A(x) = b] < \frac{1}{2} + \varepsilon(m(n)).$$

Let A' be any circuit of size at most $s'_2(n) = s_2(m(n))/2$. Toward a contradiction, suppose[8]

$$\Pr[A'(\mathsf{Enc}(z,0)) = 1] - \Pr[A'(\mathsf{Enc}(z,1)) = 1] \geq \varepsilon'(n) = 2 \cdot \varepsilon(m(n)).$$

By definition of Enc, the above can be written as

$$\Pr_{(x,b)\sim\mathcal{E}}[A'(x,b) = 1] - \Pr_{(x,b)\sim\mathcal{E}}[A'(x, 1 \oplus b) = 1] \geq 2 \cdot \varepsilon(m(n)).$$

Let $P^{A'}$ denote Yao's next-bit predictor applied to A'. In particular,

On input $x \in \{0,1\}^n$, $P^{A'}$ samples a random bit $c \sim \mathcal{U}$ and outputs c iff $A'(x,c) = 1$.

By a standard argument, it holds for some fixed choice of $c \in \{0,1\}$ that

$$\Pr_{(x,b)\sim\mathcal{E}}[P^{A'}(x) = b] \geq \frac{1}{2} + \varepsilon(m(n)).$$

Finally, observe that $P^{A'}$ with randomness fixed as above can be implemented as a circuit of size at most $s_2(m(n))$. This contradicts the assumption that $\mathcal{E}$ is a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$. We conclude that Enc is $(\mathsf{SIZE}[s'_2], \varepsilon')$-secure.

This completes the proof of the lemma. ◀

Since a witness encryption scheme that is $(\mathsf{SIZE}[s], \varepsilon)$-secure for all polynomials s and ε^{-1} is $\mathsf{P/poly}$-secure, we obtain the corollary below, which gives the second direction of Theorem 4.

▶ **Corollary 55.** *Consider any language $L \in \mathsf{NP}$. Suppose there exist a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2, ε^{-1}, (R, R_{wit}) is an honest half-Levin reduction from L to $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$. Then, there is a $\mathsf{P/poly}$-secure witness encryption scheme for L.*

[8] We have removed the absolute value signs without loss of generality: a similar argument can be applied if the left-hand side is at most $-\varepsilon'(n)$.

3.1.2 Proof of Theorem 6

▶ **Lemma 56.** *Consider any language $L \in \mathsf{NP}$, and suppose a BPP-secure witness encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ exists for L.*

Then, there exist a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2 and ε^{-1}, (R, R_{wit}) is an honest BPP-black-box half-Levin reduction from L to $\mathsf{search\text{-}CGL}^{s_2}_{s_1}[\varepsilon]$

Proof. Consider a language $L \in \mathsf{NP}$ with witness relation $R_L \subseteq \{0,1\}^n \times \{0,1\}^{n^c}$ for some constant $c \in \mathbb{N}$ and an L-instance $z \in \{0,1\}^n$. Let $(\mathsf{Enc}, \mathsf{Dec})$ be a BPP-secure ℓ-length witness encryption scheme for L. Let the polynomial s_1 be such that $s_1(\ell(n))$ upper bounds both n^c and the running time of Dec on L-instances of length n. As in Lemma 52, we consider the following distribution $\mathcal{E}$.

Sample $b \sim \mathcal{U}$. Let $x := \mathsf{Enc}(z, b) \in \{0,1\}^{\ell(n)}$ and output (x, b).

Our reduction R will make one query to its oracle: namely, $(z, \mathcal{E})$. After receiving a hypothesis h in return, R will sample from $\mathcal{E}$ and simulate h to obtain an empirical estimate of $\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b]$. R will accept iff the estimate of this value is greater than $1/2 + \varepsilon(\ell(n))/3$.

Let A be a PPT algorithm, $z \in \{0,1\}^n$ an instance of L, and s_2, ε^{-1} arbitrary polynomials. Assume that A meets the condition in Definition 32 on z: namely, letting $(z, \mathcal{E})$ be the query made by $R(z)$, if $\mathcal{E}$ is not a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$, then with probability at least $1 - 2^{-\Omega(n)}$ over its internal randomness, A returns a hypothesis h with agreement at least $1/2 + \varepsilon(\ell(n))/2$ over $\mathcal{E}$.

First consider the case that $z \in L$. Let $w \in \{0,1\}^{n^c}$ be a witness for z. Let C be the circuit that, with z and w hard-wired, given x as input, outputs $b := \mathsf{Dec}(x, z, w)$. Note that the size of C is at most $s_1(\ell(n))^2$. Without loss of generality, assume $s_2(\ell(n)) \geq s_1(\ell(n))^2$. By the correctness of the witness encryption scheme, we have

$$\begin{aligned}\max_{C' \in \mathsf{SIZE}[s_2(\ell(n))]} \left\{ \Pr_{(x,b)\sim\mathcal{E}}[C'(x) = b] \right\} &\geq \max_{C' \in \mathsf{SIZE}[s_1(\ell(n))^2]} \left\{ \Pr_{(x,b)\sim\mathcal{E}}[C'(x) = b] \right\} \\ &\geq \Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] \\ &= 1.\end{aligned}$$

This implies that $\mathcal{E}$ is a yes-instance of $\mathsf{CGL}^{s_2}_{(s_1)^2}[\varepsilon]$. Moreover, by the assumption on A, A outputs a hypothesis with agreement at least $1/2 + \varepsilon(\ell(n))/2$ over $\mathcal{E}$ with probability at least $1 - 2^{-\Omega(n)}$, so the reduction accepts with probability at least $1 - 2^{-\Omega(n)}$ overall.

Now consider the case that $z \notin L$. Toward a contradiction, suppose that with probability at least $1 - \varepsilon(\ell(n))/4$, A returns a hypothesis h such that

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \geq \frac{1}{2} + \frac{\varepsilon(\ell(n))}{4}.$$

Let A' be the algorithm that, on input (z, x), constructs the distribution $\mathcal{E}$ from z as above, simulates A on $(z, \mathcal{E})$ to obtain a hypothesis h, and then outputs $h(x)$. Then,

$$\Pr_{A\,;\,(x,b)\sim\mathcal{E}}[A'(z, x) = b] \geq \left(1 - \frac{\varepsilon(\ell(n))}{4}\right) \cdot \left(\frac{1}{2} + \frac{\varepsilon(\ell(n))}{4}\right) > \frac{1}{2} + \frac{\varepsilon(\ell(n))}{16}.$$

By the reasoning in Lemma 52 and the definition of $\mathcal{E}$, this implies that

$$|\Pr[A'(z, \mathsf{Enc}(z, 1)) = 1] - \Pr[A'(z, \mathsf{Enc}(z, 0)) = 1]| \geq \frac{\varepsilon(\ell(n))}{8},$$

contradicting the security of the witness encryption scheme on input z. So, with probability greater than $\varepsilon(\ell(n))/4$, A returns a hypothesis h with agreement over $\mathcal{E}$ less than $1/2 + \varepsilon(\ell(n))/4$, in which case the reduction rejects with probability $1 - 2^{-\Omega(n)}$. Overall, the reduction rejects with probability greater than $\varepsilon(\ell(n))/8 = n^{-O(1)}$, as desired. Finally, by the assumption on A, $\mathcal{E}$ must be a no-instance of $\mathsf{CGL}^{s_2}_{(s_1)^2}[\varepsilon]$.

Also note: it is easy to see that the reduction is half-Levin, since a circuit C simulating Dec can always be constructed from an instance z' and witness w' in polynomial time.

The correctness of BPP-black-box half-Levin reduction follows. This completes the proof of the lemma. ◀

▶ **Lemma 57.** *Consider any language $L \in \mathsf{NP}$. Suppose there exist a polynomial s_1 and a pair of machines (R, R_{wit}) such that, for all polynomials s_2, ε, (R, R_{wit}) is a BPP-black-box half-Levin reduction from L to $\mathsf{search\text{-}CGL}^{s_2}_{s_1}[\varepsilon]$. Then, BPP-secure witness encryption exists for L.*

Proof. Let (R, R_{wit}) be a BPP-black-box half-Levin reduction from L to $\mathsf{search\text{-}CGL}$, and let the polynomial $m : \mathbb{N} \to \mathbb{N}$ be such that R maps L-instances of length n to distributions $\mathcal{E}$ supported over $\{0,1\}^{m(n)} \times \{0,1\}$. We define a witness encryption scheme for L as in Lemma 54. In particular:

For $z \in \{0,1\}^n$ and $a \in \{0,1\}$, $\mathsf{Enc}(z, a)$ computes the query $\mathcal{E}$ made by $R(z)$, samples $(x, b) \sim \mathcal{E}$ and then outputs $(x, a \oplus b) \in \{0,1\}^{m(n)+1}$.

$\mathsf{Dec}((x, b'), z, w)$ computes $C := R_{wit}(z, w)$ and then outputs $C(x) \oplus b'$.

We will first show the correctness of $(\mathsf{Enc}, \mathsf{Dec})$. Suppose $z \in \{0,1\}^n$ is a yes-instance of L with witness w. By the definition of a half-Levin reduction to $\mathsf{search\text{-}CGL}$, $C = R_{wit}(z, w)$ is a circuit of size at most $\{0,1\}^{s_1(m(n))}$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1.$$

Then, for any $a \in \{0,1\}$,

$$\Pr_{(x,b)\sim\mathcal{E}}[\mathsf{Dec}((x, a \oplus b), z, w) = a] = 1,$$

as desired.

We will now show the security of $(\mathsf{Enc}, \mathsf{Dec})$. Toward a contradiction, suppose that for some PPT algorithm A and polynomial q, for infinitely many $n \in \mathbb{N}$ and *no*-instances $z \in \{0,1\}^n$ of L,

$$\Pr[A(z, \mathsf{Enc}(z, 0)) = 1] - \Pr[A(z, \mathsf{Enc}(z, 1)) = 1] \geq \frac{1}{q(n)}.$$

Let the polynomial s_2 be such that $s_2(m(n))^{1/2}$ upper bounds the running time of A on inputs of length $m(n) + 1 + n$, and let $\varepsilon(m(n)) := 1/(2 \cdot q(n))$.

Consider the PPT algorithm A' that, on input $(z, \mathcal{E})$, outputs a circuit describing P^A as in Lemma 54, fixing the bit c that maximizes success probability over $\mathcal{E}$, which A' determines by sampling from $\mathcal{E}$ and taking empirical estimates. By the same reasoning as in Lemma 54, with probability at least $1 - 2^{-\Omega(n)}$, A' outputs a hypothesis P^A of size at most $s_2(m(n))$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[P^A(x) = b] \geq \frac{1}{2} + \varepsilon(m(n)).$$

Thus, for infinitely many inputs z, A' meets the condition on the oracle of an instance-wise BPP-black-box reduction to search-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$: namely, given the query $(z, \mathcal{E})$ made by $R(z)$, A' returns a hypothesis with non-trivial success probability over $\mathcal{E}$. However, $\mathcal{E}$ is *not* a no-instance of $\mathsf{CGL}^{s_2}_{s_1}[\varepsilon]$ for the no-instance z of L. This contradicts the correctness of R. ◀

3.1.3 Proof of Theorem 7

▶ **Lemma 58.** *Consider a language $L \in \mathsf{NP}$. For $s_2, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$, assume that $(\mathsf{SIZE}[s_2], \varepsilon)$-secure $c \log n$-length witness encryption exists for L. Let the polynomial s_1 be such that $s_1(n)$ upper bounds the running time of Dec on L-instances of length n. Let $s'_1(c \log n) = 4 \cdot s_1(n) \cdot \log n$, $s'_2(c \log n) = s_2(n)$, and $2 \cdot \varepsilon'(c \log n) = \varepsilon(n)$. Assume, for some E-computable distribution family D, that $\mathsf{E} \not\subseteq \mathsf{io}\text{-}\mathsf{SIZE}^{\frac{1}{2}+\varepsilon'(n)}_{D}[s'_2(n)]$.*

Then, there is a half-Levin reduction (R, R_{wit}) from L to $\mathsf{GapDistMCSP}^{s'_2}_{s'_1}[\varepsilon']$ mapping L-instances of length n to truth-tables of length n^c, where R_{wit} runs in time at most $s'_1(c \log n)$.

Proof. The reduction R will first define a distribution $\mathcal{E}$ as in the proof of Lemma 52: namely,

Sample $b \sim \mathcal{U}$. Let $x := \mathsf{Enc}(z, b) \in \{0,1\}^{c \log n}$. Output (x, b).

Let $\mathcal{E}'$ denote the marginal distribution of $\mathcal{E}$ over $x \in \{0,1\}^{c \log n}$.

The reduction R then proceeds as follows. If, for all $x \in \mathsf{supp}(\mathcal{E}')$, there is a unique $b_x \in \{0,1\}$ such that $(x, b_x) \in \mathsf{supp}(\mathcal{E})$, R constructs a truth-table $f : \{0,1\}^{c \log n} \to \{0,1\}$ such that

$$f(x) = \begin{cases} b_x & x \in \mathsf{supp}(\mathcal{E}') \\ 0 & \text{otherwise} \end{cases}$$

and outputs $(f, \mathcal{E}')$. If, for any x, both $(x, 0)$ and $(x, 1)$ are in $\mathsf{supp}(\mathcal{E})$, then R outputs $(f'_{c \log n}, D_{c \log n})$, where $f' \in \mathsf{E}$ and $f' \notin \mathsf{SIZE}^{\frac{1}{2}+\varepsilon'(n)}_{D}[s'_2(n)]$. This function exists by our assumption, and both it and a table representing $D_{c \log n}$ can be constructed uniformly in time $\mathsf{poly}(n)$. Note that $(f'_{c \log n}, D_{c \log n})$ is a no-instance of $\mathsf{GapDistMCSP}^{s'_2}_{s'_1}[\varepsilon']$.

In the case that $z \in L$, for all $x \in \mathsf{supp}(\mathcal{E}')$, there is a unique $b_x \in \{0,1\}$ such that $(x, b_x) \in \mathsf{supp}(\mathcal{E})$; this follows from the correctness of the witness encryption scheme. Thus, using a circuit for Dec as in the proof of Lemma 52, we get that $(f, \mathcal{E}')$ is a yes-instance of $\mathsf{GapDistMCSP}^{s'_2}_{s'_1}[\varepsilon']$. Let R_{wit} be the function that, given z and a witness w for $z \in L$, outputs a circuit describing $\mathsf{Dec}(-, z, w)$. It follows that (R, R_{wit}) is a half-Levin reduction. Note that R_{wit} runs in time at most $4 \cdot s_1(n) \cdot \log n$.

In the case that $z \notin L$, if both $(x, 0), (x, 1) \in \mathsf{supp}(\mathcal{E})$, then R correctly outputs a no-instance of $\mathsf{GapDistMCSP}$. Now supposing that the labels associated with strings x are unique, let C be any circuit of size $s'_2(c \log n) = s_2(n)$. Toward a contradiction, suppose

$$\Pr_{(x,b_x)\sim\mathcal{E}}[C(x) = b_x] = \Pr_{x\sim\mathcal{E}'}[C(x) = f(x)] \geq \frac{1}{2} + \varepsilon'(c \log n).$$

As in Lemma 52, this implies

$$\left|\Pr_r[C(\mathsf{Enc}(z, 1, r)) = 1] - \Pr_r[C(\mathsf{Enc}(z, 0, r)) = 1]\right| \geq \varepsilon(n),$$

contradicting the security of the witness encryption scheme. Thus, $(f, \mathcal{E})$ is a no-instance of $\mathsf{GapDistMCSP}^{s'_2}_{s'_1}[\varepsilon']$.

This completes the proof of the lemma. ◀

▶ **Lemma 59.** *Consider a constant $c \in \mathbb{N}$ and $s_1, s_2, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$. Suppose that there is a half-Levin reduction (R, R_{wit}) from SAT to $\mathsf{GapDistMCSP}^{s_2}_{s_1}[\varepsilon]$, where R maps SAT-instances of length $n \in \mathbb{N}$ to truth-tables of length n^c, and R_{wit} runs in time at most $\widetilde{O}(s_1(c \log n))$ on L-instances of length n. Let $s'_1(n) = s_1(c \log n)$, $s'_2(n) = s_2(c \log n)/2$, and $\varepsilon'(n) = 2 \cdot \varepsilon(c \log n)$.*

Then, there is a $(\mathsf{SIZE}[s'_2], \varepsilon')$-secure $\widetilde{O}(s'_1(n))$-decryption-time $(c \log n + 1)$-length witness encryption scheme for SAT. Moreover, there exists an E-computable distribution family D such that $\mathsf{E} \not\subseteq \mathsf{io}\text{-}\mathsf{SIZE}^{\frac{1}{2}+\varepsilon(n)}_{D}[s_2(n)]$.

Proof. We argue as in the proof of Lemma 54. In particular, the witness encryption scheme for SAT is defined as follows.

For $\varphi \in \{0,1\}^n$ and $a \in \{0,1\}$, $\mathsf{Enc}(\varphi, a)$ computes $(f, D) = R(\varphi)$, samples $x \sim D$ and then outputs $(x, a \oplus f(x))$.

$\mathsf{Dec}((x, c), \varphi, w)$ computes $A := R_{wit}(\varphi, w)$ and then outputs $A(x) \oplus c$.

Note that, the output length of Enc is $c \log n + 1$ and that the running time of Dec is at most $\widetilde{O}(s'_1(n))$.

For $\varphi \in \mathsf{SAT}$, $A = R_{wit}(\varphi, w)$ is a circuit of size at most $s_1(c \log n)$ such that

$$\Pr_{x \sim D}[A(x) = f(x)] = 1.$$

The correctness of $(\mathsf{Enc}, \mathsf{Dec})$ follows as in Lemma 54.

For $\varphi \notin \mathsf{SAT}$, by the same reasoning as in Lemma 54, we have that Enc is $(\mathsf{SIZE}[s'_2], \varepsilon')$-secure.

To see the second conclusion of the lemma, let $\varphi \in \{0,1\}^n$ be a trivial no-instance of SAT, constructible uniformly in time $O(n)$. Let $(f, D) = R(\varphi)$. By the correctness of the reduction (R, R_{wit}) and the definition of $\mathsf{GapDistMCSP}$, we have that, for all circuits C of size $s_2(c \log n)$,

$$\Pr_{x \sim D}[C(x) = f(x)] \leq \frac{1}{2} + \varepsilon(c \log n).$$

Also note that tables representing both f and D are uniformly constructible in time $2^{O(c \log n)}$, as desired. ◀

3.2 NP-hardness of learning polynomial-size circuits

We start with the following auxiliary lemmas.

▶ **Lemma 60** ([40, 44])**.** *For all sufficiently large $\ell, m \in \mathbb{N}$ with $m < 2^\ell$, for some $d = O(\ell^2 \log m)$ there are sets $\mathcal{S}_1, ..., \mathcal{S}_m \subseteq [d]$ constructible in time $\mathsf{poly}(m, d)$ such that $|\mathcal{S}_i| = \ell$ and*

$$\sum_{j<i} 2^{|S_i \cap S_j|} \leq m - 1$$

for all $i \in [m]$.

▶ **Lemma 61** ([20])**.** *For $n, t \in \mathbb{N}$, consider the "multiselector" function $Sel^{n \to t}$ that, given a string $x \in \{0,1\}^n$ and indices $i_1, ..., i_t \in [n]$, outputs $(x_{i_1}, ..., x_{i_t})$.*

For all $n, t \in \mathbb{N}$, there is a logspace-uniform boolean circuit of size $O(n + t \cdot \log^3(n))$ computing $Sel^{n \to t}$.

▶ **Lemma 62** ([26]; see [16] and [15]). *There exist constants $a, c \in \mathbb{N}$ as follows. Consider a function $f : \{0,1\}^n \to \{0,1\}$ and a parameter $\varepsilon = o(1)$. There is a function $\hat{f} : \{0,1\}^{a \cdot n} \to \{0,1\}$ such that the truth-table of $\hat{f}$ can be computed from that of f in time $\mathsf{poly}(2^n/\varepsilon)$ given ε, and $\hat{f}$ has the following properties.*

- **Local encodability:** *There is a non-adaptive f-oracle circuit of size at most $(n/\varepsilon)^c$ computing $\hat{f}$, making at most $1/\varepsilon^2$ queries to f.*
- **Decodability:** *For any oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}$ and sufficiently large time bound $t \in \mathbb{N}$,*

$$\mathsf{K}^{(2^n \cdot t/\varepsilon)^c,\, \mathcal{O}}(f) \leq \mathsf{K}^{t,\, \mathcal{O}}_{\frac{1}{2}+\varepsilon}(\hat{f}) + 2^{n/2} \cdot (1/\varepsilon)^c + o(2^n).$$

The following is the "algorithmic information extraction" lemma of [16] modified to give a set B defined in terms of $\mathsf{K}^{\mathsf{poly}}$ rather than K. It is easy to see from the proof in [16] that B can be defined in this way.

▶ **Lemma 63.** *Consider an algorithm D running in time t_D, parameters $n, \Delta, \varepsilon^{-1}, \ell, d \in \mathbb{N}$, and $g_i : \{0,1\}^\ell \to \{0,1\}$ for $i \in [n]$. Let*

$$\left(\left(\mathcal{S}_1^{(1)}, ..., \mathcal{S}_\Delta^{(1)}\right), ..., \left(\mathcal{S}_1^{(n)}, ..., \mathcal{S}_\Delta^{(n)}\right)\right)$$

be a family of ℓ-sized subsets of $[d]$. Define

$$G(z) = \left(z, w^{(1)}(z), ..., w^{(n)}(z)\right),$$

where for $z \in \{0,1\}^d$,

$$w^{(i)}(z) = \left(g_i\left(z|_{\mathcal{S}_1^{(i)}}\right), ..., g_i\left(z|_{\mathcal{S}_\Delta^{(i)}}\right)\right) \in \{0,1\}^\Delta.$$

Define a set

$$B := \left\{ i \in [n] \;\middle|\; \mathsf{K}^{4 \cdot 2^\ell \cdot t_D,\, D}_{\frac{1}{2}+\varepsilon}(g_i) \leq 4 \cdot (\Delta n) \cdot \sum_{j<i} 2^{|S_i \cap S_j|} \right\}.$$

Then, for any advice string $\beta \in \{0,1\}^{O(\Delta n)}$,

$$|\Pr\left[D(\beta, G(z))\right] - \Pr\left[D(\beta, G(z)|_B)\right]| < 2\Delta n \cdot \varepsilon,$$

where $G(z)|_B = (z, u^{(1)}, ..., u^{(n)})$ is such that $u^{(i)} = w^{(i)}(z)$ for $i \in B$ and $u^{(i)} \sim \mathcal{U}_\Delta$ for $i \notin B$. The probabilities above are over $z \sim \mathcal{U}_d$ and $u^{(i)} \sim \mathcal{U}_\Delta$.

▶ **Lemma 64.** *For some polynomial p and constant $k \in \mathbb{N}$, for any polynomials $g, \Delta, \theta, \gamma^{-1}$ with $\theta(n) \leq n$, $\Delta(n) \geq n$, and $\gamma(n) \leq 4n^{-4}$ for $n \in \mathbb{N}$, there exists a polynomial-time machine R satisfying the following. For a polynomial s_1 with $s_1(n) \geq \gamma^{-1}(\Delta(n))^k$ for $n \in \mathbb{N}$, let*

$$\lambda(n) = \frac{s_1(n)}{\theta(n) \cdot \log n},$$

and consider an advice string $f = (f_i \in \{0,1\}^{\lambda(n)})_{i \in [n]}$ such that, for every subset $B \subseteq [n]$,

$$\mathsf{K}^{p(\lambda(n))}(f_B) \geq |B| \cdot \lambda(n) - 2n \cdot \log n.$$

Then, given as input f and an instance φ of $\mathsf{Gap}_g\mathsf{MMSA}^\Delta_\theta$ over n variables, R outputs a circuit $\mathcal{E}$ sampling a flat distribution supported over $\{0,1\}^{2\Delta(n) \cdot n} \times \{0,1\}$ as follows.

- *If φ is a yes-instance of $\mathsf{Gap}_g\mathsf{MMSA}^{\Delta}_{\theta}$, then there exists a circuit C of size at most $s_1(n)$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x)=b]=1.$$

- *If φ is a no-instance of $\mathsf{Gap}_g\mathsf{MMSA}^{\Delta}_{\theta}$, then for every circuit C' of size at most*

$$s_2(n) := s_1(n)\cdot g(n)/(\log n)^3,$$

it holds that

$$\Pr_{(x,b)\sim\mathcal{E}}[C'(x)=b] < \frac{1}{2}+\gamma(n).$$

Proof. Let φ be a given instance of $\mathsf{Gap}_g\mathsf{MMSA}^{\Delta}_{\theta}$, where φ is over n variables. Let $a, c \in \mathbb{N}$ be as in Lemma 62. Consider an arbitrary polynomial s_1 such that $s_1(n) \geq \gamma(\Delta)^{-8c}$. Define $\ell := a\log(\lambda)$. Let $f = (f_i \in \{0,1\}^{\lambda})_{i\in[n]}$ satisfy the condition stated in the lemma.

The reduction R. Identifying each f_i with a function $f_i : \{0,1\}^{\log\lambda} \to \{0,1\}$, let $\hat{f}_i : \{0,1\}^{\ell} \to \{0,1\}$ be the encoded function as in Lemma 62 with $\varepsilon = \gamma(\Delta)/(2\Delta^2)$. Let $\left(\mathcal{S}_1^{(1)},...,\mathcal{S}_{\Delta}^{(1)}\right),...,\ \left(\mathcal{S}_1^{(n)},...,\mathcal{S}_{\Delta}^{(n)}\right)$ be the sets of size ℓ guaranteed by Lemma 60 with $d = O(\ell^2\cdot\log(\Delta n)) = o(\Delta n)$.

For any choice of $z \in \{0,1\}^d$, $r \in \{0,1\}^{\Delta}$, and $b \in \{0,1\}$, define $(v_1,...,v_n) = \mathit{Share}(\varphi, b; r) \in (\{0,1\}^{\Delta})^n$, and define the string $x(z,r,b)$ as

$$\left(z\ ,\ \left(\hat{f}_1\left(z|_{\mathcal{S}_1^{(1)}}\right),...,\hat{f}_1\left(z|_{\mathcal{S}_{\Delta}^{(1)}}\right)\right)\oplus v_1\ ,\ \ldots\ ,\ \left(\hat{f}_n\left(z|_{\mathcal{S}_1^{(n)}}\right),...,\hat{f}_n\left(z|_{\mathcal{S}_{\Delta}^{(n)}}\right)\right)\oplus v_n\ \right),$$

padded to have length $2\Delta n$. The output of the reduction is a circuit sampling the following distribution $\mathcal{E}$:

For uniformly random $z\sim\mathcal{U}_d$, $r\sim\mathcal{U}_{\Delta}$, and $b\sim\{0,1\}$, output $(x(z,r,b),b)$.

Note that $\mathcal{E}$ has a $\mathsf{poly}(n)$-size sampling circuit and that the marginal distribution of $\mathcal{E}$ over x is flat.

Completeness. To see the completeness of the reduction, suppose $\mathsf{MMSA}(\varphi)\leq\theta$, and let $T_\alpha = \{j\in[n] \mid \alpha(j)=1\}$ where $\alpha : [n]\to\{0,1\}$ is an assignment of Hamming weight at most θ satisfying φ.

We will construct a circuit C of size at most $s_1(n)$ such that $C(x)=b$ for all (x,b) in the support of $\mathcal{E}$. C takes as input a string $x=(z,w_1,...,w_n)$ with $z\in\{0,1\}^d$ and $w_i\in\{0,1\}^{\Delta}$ for $i\in[n]$, padded to have length $2n\Delta$. C uses $\theta\Delta$ encoding circuits as in Lemma 62 to compute the values

$$\left\{\ \hat{f}_j\left(z|_{\mathcal{S}_m^{(j)}}\right)\ \middle|\ j\in T_\alpha\ ,\ m\in[\Delta]\ \right\}$$

by making at most $\theta\Delta\cdot(1/\varepsilon)^2 < \gamma^{-4}(\Delta)$ total queries to f_{T_α}. C answers these queries with the string f_{T_α} hard-wired along with a multiselector circuit as in Lemma 61. C then XORs the outputs of the encoding circuits with the appropriate inputs w_j to obtain an authorized set of shares $(v_j)_{j\in T_\alpha}$. Finally, C applies Rec with φ and T_α hard-wired to obtain the secret b.

C consists of $2n\Delta$ input gates, $\theta\lambda$ gates hard-wiring f_{T_α}, $O(\theta\lambda+\gamma^{-4}(\Delta)\cdot\log^3(\theta\lambda)) = O(\theta\lambda)$ gates for multiselection, $\theta\Delta$ encoding circuits each of size $(\log\lambda/\varepsilon)^c = o(\lambda)$, $O(\theta\Delta)$ gates to XOR, and $O((\theta\Delta)^2)$ gates for Rec. This amounts to at most $O(\theta\lambda) < s_1(n)$ gates in total.

Thus, there exists a circuit C of size at most $s_1(n)$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x)=b]=1.$$

Soundness. We now argue for the soundness of the reduction. Suppose $\mathsf{MMSA}(\varphi) > \theta \cdot g(n)$. For a fixed circuit C' of size $s_2(n) = s_1(n) \cdot g(n)/(\log n)^3$, define a program D as follows. D takes as advice a string $\beta(b,r) := (b, v_1, ..., v_n) \in \{0,1\}^{O(n\cdot\Delta)}$, where $(v_1, ..., v_n) = Share(\varphi, b; r)$, and accepts its input $(z, w_1, ..., w_n)$ if and only if

$$C'(z, w_1 \oplus v_1, ..., w_n \oplus v_n) = b.$$

Note that D runs in time at most $t_D := 4 \cdot s_2(n) \cdot \log(s_2(n))$. Now, define a set B as in Lemma 63, applying the lemma with $g_i = \hat{f}_i$ for $i \in [n]$. In particular,

$$B := \left\{ i \in [n] \;\middle|\; \mathsf{K}^{4\cdot\lambda^a\cdot t_D,\, D}_{\frac{1}{2}+\varepsilon}(\hat{f}_i) \le (\Delta n)^2 \right\}.$$

By Lemma 63,

$$\begin{aligned} |\Pr[D(\beta, G(z))] - \Pr[D(\beta, G(z)|_B)]| &< 2\Delta^2 \cdot \varepsilon \\ &= \gamma(\Delta), \end{aligned} \tag{1}$$

where G and $G(z)|_B$ are as defined in that lemma.

We claim that B is not too large. Note that for $i \in B$, by the decodability statement of Lemma 62, for some polynomial q,

$$\begin{aligned} \mathsf{K}^{q(\lambda),\, D}(f_i) &\le \mathsf{K}^{4\cdot\lambda^a\cdot t_D,\, D}_{\frac{1}{2}+\varepsilon}(\hat{f}_i) + \lambda^{1/2} \cdot (1/\varepsilon)^c + o(\lambda) \\ &< o(\lambda). && \text{(definition of } B\text{)} \end{aligned}$$

Thus, for $p(\lambda) := n^2 \cdot q(\lambda) \cdot t_D$,

$$\begin{aligned} |f_B| = |B| \cdot \lambda &\le \mathsf{K}^{p(\lambda)}(f_B) + 2n \cdot \log n \\ &\le \mathsf{K}^{n^2\cdot q(\lambda),\, D}(f_B) + |D| + 2n \cdot \log n \\ &\le \sum_{i\in B} \left(\mathsf{K}^{q(\lambda),\, D}(f_i) \right) + O(|C'| \cdot \log |C'|) + |\beta| + O(n \cdot \log n) \\ &< \frac{s_1(n) \cdot g(n)}{2 \log n} + o(|B| \cdot \lambda) \\ &= \frac{\lambda \cdot \theta \cdot g(n)}{2} + o(|B| \cdot \lambda). \end{aligned}$$

It follows that

$$\begin{aligned} |B| &\le \frac{\theta \cdot g(n)}{2 \cdot (1 - o(1))} \\ &< \theta \cdot g(n), \end{aligned}$$

so B is not an authorized set.

Now observe that

$$\begin{aligned} \Pr_{z,r,b}[C'(x(z,r,b)) = b] &= \Pr[D(\beta(b,r), G(z)) = 1] && \text{(definition of } D\text{)} \\ &< \Pr[D(\beta(b,r), G(z)|_B) = 1] + \gamma(\Delta) && \text{(Eq. (1))} \\ &\le \frac{1}{2} + \gamma(\Delta), \end{aligned}$$

where the last line follows from the security of $Share$. We conclude that for any circuit C' of size at most $s_2(n)$,

$$\Pr_{(x,b)\sim\mathcal{E}}[C'(x) = b] < \frac{1}{2} + \gamma(\Delta) \le \frac{1}{2} + \gamma(n).$$

This completes the proof of the lemma. ◀

▶ **Lemma 65** (Claim 7.3 of [16]). *For a uniformly random string $f = (f_i)_{i\in[n]} \sim (\mathcal{U}_\lambda)^n$, it holds with probability $1 - 2^{-n}$ that for all subsets $B \subseteq [n]$,*

$$\mathsf{K}(f_B) \geq |B| \cdot \lambda - 2n \cdot \log n.$$

▶ **Theorem 66** (Restatement of Theorem 8). *For some constant $c \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ with $g(n) = n^{1/(\log\log n)^{O(1)}}$ for $n \in \mathbb{N}$, for any polynomials γ^{-1}, s_1 such that $s_1(n) \geq \gamma^{-c}(n)$ for $n \in \mathbb{N}$, there is a randomized many-one reduction from SAT to $\mathsf{CGL}^{s_1\cdot g}_{s_1}[\gamma]$.*

Proof. By Lemma 43, there is a reduction R_0 from SAT to $\mathsf{Gap}_{g_0}\mathsf{MMSA}^{\Delta}_{\theta}$ for some

$$g_0(n) = n^{\frac{1}{(\log\log n)^{O(1)}}},$$

polynomial $\Delta(n) \geq n$, and $\theta(n) \leq n$. Let $k \in \mathbb{N}$ be the constant of Lemma 64, and define the constant $c \in \mathbb{N}$ such that $n^c > \Delta(n)^k$. For arbitrary polynomials γ^{-1}, s_1, define $\gamma'(n) := \gamma(2n \cdot \Delta(n))$ and $s_1'(n) := s_1(2n \cdot \Delta(n))$ for $n \in \mathbb{N}$. We will apply Lemma 64 with parameters s_1', γ', g_0, Δ, and θ.

More specifically, our reduction will first apply R_0 to obtain an instance φ of $\mathsf{Gap}_{g_0}\mathsf{MMSA}^{\Delta}_{\theta}$ on n variables, and then it will sample a random string $f = (f_i)_{i\in[n]} \sim (\mathcal{U}_\lambda)^n$. It will then apply R as in Lemma 64 to obtain an instance $\mathcal{E}$ of $\mathsf{CGL}^{s_2}_{s_1}[\gamma]$ supported over $\{0,1\}^{2n\cdot\Delta(n)} \times \{0,1\}$, where

$$\begin{aligned} s_2(n \cdot \Delta(n)) &= \frac{s_1(2n \cdot \Delta(n)) \cdot g_0(n)}{(\log n)^3} \\ &= s_1(2n \cdot \Delta(n)) \cdot g(2n \cdot \Delta(n)), \end{aligned}$$

where we define $g(2n \cdot \Delta(n)) := g_0(n)/(\log n)^3$.

By Lemma 65, with probability $1 - 2^{-n}$ over the choice of f, f satisfies the condition of Lemma 64. The desired statement is then immediate from the correctness of R and the definition of CGL. ◀

Corollary 9 now follows by combining Theorem 8 with Lemma 33.

3.3 NP-hardness of improper learning in other settings

▶ **Theorem 67** (Restatement of Theorem 12). *There exist a distribution $D \in \mathsf{PSAMP}$, an NP-complete language L, and a polynomial s_1 such that, for all polynomials s_2, ε^{-1}, there is a function $\lambda : \mathbb{N} \to \mathbb{N}$ such that $\lambda(n) = O(\log n)$ for $n \in \mathbb{N}$ and L reduces to D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$ under an honest half-Levin reduction.*

Proof. Let $D = \{D_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{PSAMP}$ be the distribution family and L the NP-complete language of Lemma 41 with witness relation V_L. That is, each D_λ is supported over truth-tables of functions $O : \{0,1\}^\lambda \to \{0,1\}$, and there is an oracle witness encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ for L with respect to D secure against programs of size and query complexity $2^{\lambda/\ell}$ and with advantage $2^{-\lambda/\ell}$ for some constant $\ell \in \mathbb{N}$.

Let $d \in \mathbb{N}$ be such that n^d upper-bounds the running time of $\mathsf{Dec}^{(-)}$ on L-instances of length n, and define $s_1(n) = n^{2d}$. Let s_2 and ε be given, and define $\lambda(n) = \ell \cdot \log(s_2(n) \cdot \varepsilon(n)^{-1}) = O(\log n)$ for $n \in \mathbb{N}$.

We now give a reduction from L to D-oracle-$\mathsf{CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$. Given an L-instance $z \in \{0,1\}^n$ for some $n \in \mathbb{N}$, the reduction outputs $\mathcal{E}^{(-)}$ as follows.

> Given access to an oracle $O : \{0,1\}^{\lambda(n)} \to \{0,1\}$, sample $b \sim \mathcal{U}$, and let $x = \mathsf{Enc}^O(1^{\lambda(n)}, z, b)$. Output (x, b).

First suppose $z \in L$. Let w be such that $(z, w) \in V_L$. Consider the oracle circuit $C^{(-)}$ that, with (z, w) hard-wired, given input x and oracle O, simulates $\mathsf{Dec}^O(1^{\lambda(n)}, x, z, w)$ and returns its output b. Note that $C^{(-)}$ has size at most $s_1(n)$. By the correctness guarantee of $(\mathsf{Enc}, \mathsf{Dec})$, for every O in the support of $D_{\lambda(n)}$ and every (x, b) in the support of $\mathcal{E}$, it holds that $C^O(x) = b$. Therefore, in this case, the reduction outputs a yes-instance of $D\text{-oracle-CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$.

Now suppose $z \notin L$, and let $C^{(-)}$ be any oracle circuit of size at most $s_2(n)$. Note that

$$2^{\lambda/\ell} = s_2(n) \cdot \varepsilon(n)^{-1} > s_2(n),$$

so $(\mathsf{Enc}, \mathsf{Dec})$ is secure against C. Suppose for some oracle O that

$$\Pr_{(x,b)\sim\mathcal{E}}[C^O(x) = b] \geq \frac{1}{2} + \varepsilon(n).$$

By the same reasoning as in Lemma 52,

$$\left|\Pr[C^O(\mathsf{Enc}^O(1^{\lambda(n)}, z, 1)) = 1] - \Pr[C^O(\mathsf{Enc}^O(1^{\lambda(n)}, z, 0)) = 1]\right| \geq \varepsilon(n) > \frac{1}{2^{\lambda(n)/\ell}}.$$

By the security of the witness encryption scheme, this occurs with probability at most $2^{-\lambda(n)/\ell} < \varepsilon(n)$ over $O \sim D_{\lambda(n)}$. Thus, in this case, the reduction outputs a no-instance of $D\text{-oracle-CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$. ◀

By combining Theorem 12 with Lemma 34, we obtain the following.

▶ **Corollary 68** (Restatement of Corollary 13)**.** *For some polynomial s, for any constant $c \in \mathbb{N}$, it is* NP*-hard under a randomized one-query reduction to agnostically oracle-PAC learn* $\mathsf{SIZE}[s(n)]$ *by* $\mathsf{SIZE}[s(n)^c]$*.*

Theorem 12 also implies NP-hardness of learning RAMs in the improper setting, as indicated below.

▶ **Corollary 69** (Restatement of Corollary 14)**.** *For all polynomials g and ε^{-1}, the problem* $\mathsf{Gap}_{g,\varepsilon}\mathsf{RAM\text{-}MINLT}$ *is* NP*-hard under a randomized many-one reduction.*

Proof. We show that $D\text{-oracle-CGL}$ as in Theorem 12 reduces to $\mathsf{GapRAM\text{-}MINLT}$. The main idea is to include the whole truth-table of an oracle O sampled from $D_{\lambda(n)}$ in all the inputs to a RAM.

Let the polynomials g and ε^{-1} be given. Let s_1 be as in Theorem 12. Let $s_2(n) = g(s_1(n))^4$, and let λ be as in Theorem 12 applied with s_2 and ε. Let D be the distribution guaranteed by Theorem 12, and recall that $D_{\lambda(n)}$ is supported over truth-tables of length $2^{\lambda(n)}$, where $s_2(n) \cdot \varepsilon^{-1}(n) < 2^{\lambda(n)} \leq \mathsf{poly}(n)$.

The reduction, given an instance $\mathcal{E}^{(-)}$ of $D\text{-oracle-CGL}^{s_2}_{s_1}[\varepsilon, \lambda]$, samples $O \sim D_{\lambda(n)}$ and then defines the following distribution $\mathcal{E}'$:

> Sample $(x, b) \sim \mathcal{E}^O$, let $x' = (x, O)$, and output (x', b).

The reduction outputs $(\mathcal{E}', 1^{s_1(n)^2}, 1^{s_1(n)^2})$.

First suppose that $\mathcal{E}^{(-)}$ is a yes-instance of $D\text{-oracle-CGL}$. In this case, there is an oracle circuit $C^{(-)}$ of size at most $s_1(n)$ such that, for all O in the support of $D_{\lambda(n)}$ and all (x, b) in the support of $\mathcal{E}^O$, $C^O(x) = b$. Note that C can be simulated by a program M of size and running time at most $s_1(n)^2$. We obtain that for all $(x', b) \in \mathsf{supp}(\mathcal{E}')$,

$$M^{x'} = M^{x,O} = b,$$

so $(\mathcal{E}', 1^{s_1(n)^2}, 1^{s_1(n)^2})$ is a yes-instance of $\mathsf{Gap}_{g,\varepsilon}\mathsf{RAM\text{-}MINLT}$.

Now suppose that $\mathcal{E}^{(-)}$ is a no-instance of D-oracle-CGL. Consider any random-access machine M of size and running time at most $g(s_1(n)^2)$. Note that M can be simulated by an oracle circuit C of size at most $s_2(n)$, where we may let the "x" part of the input $x' = (x, O) \in \{0,1\}^n \times \{0,1\}^{2^{\lambda(n)}}$ be a standard (non-oracle) input to C. Then, since $\mathcal{E}^{(-)}$ is a no-instance of D-oracle-CGL, with probability at least $1 - \varepsilon(n)$ over $O \sim D_{\lambda(n)}$,

$$\Pr_{(x,b)\sim\mathcal{E}^O}\left[C^O(x) = b\right] < \frac{1}{2} + \varepsilon(n).$$

This implies, for such oracles O, that

$$\Pr_{(x',b)\sim\mathcal{E}'}\left[M^{x'} = b\right] < \frac{1}{2} + \varepsilon(n).$$

Thus, with probability at least $1 - \varepsilon(n)$ over the randomness of the reduction, $(\mathcal{E}', 1^{s_1(n)^2}, 1^{s_1(n)^2})$ is a no-instance of $\mathsf{Gap}_{g,\varepsilon}$RAM-MINLT, as desired. ◀

3.4 Excluding Pessiland, Heuristica, and Minicrypt

We show that if the hardness of learning can be based on the hardness of NP (under a non-adaptive reduction), then so can the existence of OWFs. Applying our main results connecting witness encryption and the NP-hardness of learning, we obtain Theorems 16 and 17 as consequences.

3.4.1 Proof of Theorem 15

We will make use of the following lemmas.

▶ **Lemma 70.** *Let $f = \{f_z\}_{z\in\{0,1\}^*}$ be a polynomial-time computable auxiliary input function. If infinitely-often one-way functions do not exist, then for every distribution family $D \in$ PSAMP and constant $b \in \mathbb{N}$, there exists a PPT machine I such that for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr_{z\sim D_n}[\ I \text{ is an } n^{-b}\text{-inverter for } f_z\ ] \geq 1 - \frac{1}{n^b}.$$

Moreover, if I does not n^{-b}-invert f_z for some $z \in \{0,1\}^n$, then for any $y \in \{0,1\}^*$,

$$\Pr_I[\ I(z,y) \text{ outputs } \perp\ ] \geq 1 - 2^{-n}.$$

Proof. Assume that i.o. one-way functions do not exist. Let S be a sampler for D. For a constant $b \in \mathbb{N}$, let A be an n^{-4b}-inverter for the function g defined as $g(r,x) := f_{S(r)}(x)$. Define the PPT machine I as follows.

> On input (z,y), sample uniformly random strings $y_1, ..., y_{n^{8b}}$. Compute the fraction γ of $i \in [n^{8b}]$ such that $A(z, y_i)$ fails to output a pre-image of its input under g. If γ is less than $2/n^{4b}$, output $A(z,y)$. Otherwise, output $\perp$.

We first claim that, with probability at least $1 - n^{-2b}$ over $z \sim D$, $A(z,-)$ is an n^{-b} inverter for f_z. Toward a contradiction, suppose the claim does not hold. That is, with probability greater than n^{-2b} over $z \sim D_n$, $A(z,-)$ only inverts f_z with success probability $1 - n^{-b}$. But then the overall probability of A inverting g, for uniformly random r and y, is at most

$$1 - n^{-2b} + n^{-2b} \cdot (1 - n^{-b}) < 1 - n^{-4b},$$

contradicting our assumption on A. Moreover, by a Chernoff bound, the probability that γ is greater than $2/n^{4b}$ is less than 2^{-n}. By the definition of I and a union bound, this proves the first part of the lemma.

For the second part, suppose that $A(z,-)$ does not n^{-b}-invert f_z. Then the expected value of γ in the definition of I is at least n^{-b}. By a Chernoff bound, the probability that γ is less than $2/n^{4b}$ is less than 2^{-n}. This completes the proof of the lemma. ◀

We will require the following lemma, which states that if NP is easy on average for randomized algorithms, then agnostic PAC learning over efficiently samplable distributions is possible.

▶ **Lemma 71** ([12]). *If* DistNP $\subseteq$ AvgBPP*, then there is a PPT machine that, for any polynomial s, for all sufficiently large $n \in \mathbb{N}$, agnostically learns* SIZE$[s(n)]$ *in the improper setting over distributions in* PSAMP/poly*.*

It remains to exhibit a reduction from SAT to inverting an auxiliary input one-way function, assuming NP-hardness of learning. The following lemma is as in [2]. We include a proof for completeness.

▶ **Lemma 72** ([2]). *Suppose there is a randomized non-adaptive honest reduction R and a polynomial $s : \mathbb{N} \to \mathbb{N}$ such that, for every constant $c \in \mathbb{N}$, R reduces* SAT *to agnostically PAC learning* SIZE$[s(n)]$ *by* SIZE$[s(n)^c]$*. Then there exist a poly-time computable auxiliary input function family $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$, a constant $b \in \mathbb{N}$, and a PPT oracle machine A satisfying the following: for any $n \in \mathbb{N}$, $\varphi \in \{0,1\}^n$, and PPT machine I that n^{-b}-distributionally inverts f_φ,*

$$A^I(\varphi) = \mathsf{SAT}(\varphi).$$

Proof. Let R be the assumed reduction from SAT to agnostic PAC learning, let the polynomial t_R denote its running time, and let the inverse polynomial ε be such that R requires an $(s, 2 \cdot \varepsilon)$-optimal hypothesis in response to each query. Define $\delta(n) := \varepsilon(n)/t_R(n)^2$. Define an auxiliary-input function

$$f_\varphi(r_0, i, r) := (r_0, i, \mathcal{E}_i^{(\varphi, r_0)}(r)[1]),$$

where $\mathcal{E}_i^{(\varphi,r_0)}$ (henceforth denoted $\mathcal{E}_i$) is the i^{th} query of R on input φ and randomness r_0, sampling a distribution (X_i, Y_i). In particular, for $r \sim \mathcal{U}_{t_R}$, $\mathcal{E}_i(r)[1]$ represents a random sample $x \sim X_i$. Note that all such samplers $\mathcal{E}_i$ can be simulated uniformly in polynomial time given (φ, r_0, i), by the efficiency of R.

Let I be a PPT machine that distributionally inverts f_φ. In particular, we require the success probability of I to be such that, with probability $1 - o(1)$ over $r_0 \sim \mathcal{U}_{t_R(n)}$, for every $i \in [t_R(n)]$, I $\delta(n)^{10}$-distributionally inverts $f_\varphi(r_0, i, -)$.

For $i \in [t_R(n)]$, define hypotheses h_i as follows:

> On input x, invoke I to find a pre-image r under $f_\varphi(r_0, i, -)$ of (r_0, i, x), and then return $y := \mathcal{E}_i(r)[2]$,

and define a machine A (taking oracle access to I) as follows:

> On input $\varphi \in \{0,1\}^n$ and randomness $r_0 \in \{0,1\}^{t_R(n)}$, simulate $R(\varphi; r_0)$, obtaining queries $\mathcal{E}_1, ..., \mathcal{E}_m$. Answer the i^{th} query of R with the hypothesis H_i, which evaluates h_i on $\delta(n)^{-8}$ independent choices of randomness (used by I) and then outputs the majority output $y \in \{0,1\}$. After answering all the queries this way and completing the simulation of R, accept φ iff R accepts.

For $i \in [t_R(n)]$, let $f_{opt}^{(i)}$ be a fixed function (of arbitrary complexity) that maximizes

$$\Pr_{(x,y)\sim(X_i,Y_i)}[f_{opt}^{(i)}(x) = y].$$

We will argue that $H_i(x)$ predicts y almost as well as $f_{opt}^{(i)}$ over (X_i, Y_i).

For any $x \in \mathsf{supp}(X_i)$, let the random variable Y_i^x denote the conditional distribution $y \sim \mathcal{E}_i(r)[2]$ given that $\mathcal{E}_i(r)[1] = x$, for $r \sim \mathcal{U}_{t_R(n)}$. For simplicity, start by considering the hypotheses $\hat{h}_i$ and $\hat{H}_i$, which are the same as h_i and H_i except taking an oracle $\hat{I}$ that "perfectly" distributionally inverts $f_\varphi(r_0, i, -)$. That is,

$$\Delta\Big((r, f_\varphi(r_0,i,r)) \ , \ (\hat{I}(\varphi, f_\varphi(r_0,i,r)), f_\varphi(r_0,i,r)) \Big) = 0,$$

for $r \sim \mathcal{U}_{t_R(n)}$. Observe that for $x \sim X_i$, outputs of $\hat{h}_i(x)$ are distributed exactly according to Y_i^x. In particular, for $r \sim \mathcal{U}_{t_R(n)}$ and $x = \mathcal{E}_i(r)[1]$,

$$\begin{aligned}
\Delta\Big((x, Y_i^x) \ , \ (x, \hat{h}_i(x)) \Big) &= \Delta\Big((x, \mathcal{E}_i(r)[2]) \ , \ (x, \mathcal{E}_i(\hat{I}(\varphi, (r_0,i,x)))[2]) \Big) && \text{(def. of } \hat{h}_i\text{)} \\
&\le \Delta\Big((x, r) \ , \ (x, \hat{I}(\varphi, (r_0,i,x))) \Big) \\
&= \Delta\Big((r, f_\varphi(r_0,i,r)) \ , \ (\hat{I}(\varphi, f_\varphi(r_0,i,r)), f_\varphi(r_0,i,r)) \Big) \\
&= 0. && (2)
\end{aligned}$$

Now, for $i \in [t_R(n)]$ and strings x in the support of X_i, define

$$\alpha_i(x) := \max_{y\in\{0,1\}} \{\Pr[Y_i^x = y]\}.$$

That is, $y \in \{0,1\}$ is the label most likely to be associated with x by (X_i, Y_i), and $\alpha_i(x)$ is the corresponding conditional probability. Note that $f_{opt}^{(i)}(x)$ must always output this optimal label y, so $\alpha_i(x)$ is the success probability of $f_{opt}^{(i)}$ on input x. Moreover, by Eq. (2), for every $x \in \mathsf{supp}(X_i)$,

$$\Pr_{\hat{h}_i}\Big[\hat{h}_i(x) = f_{opt}^{(i)}(x)\Big] = \alpha_i(x). \tag{3}$$

We now fix i and x and break the subsequent analysis into two cases: either $f_{opt}^{(i)}$ predicts the label of x with probability substantially bounded away from $1/2$, or not.

Case I: $\alpha_i(x) \ge 1/2 + \delta(n)^2$. For $j \in [\delta(n)^{-8}]$, let z_j be the Bernoulli random variable that equals 1 if $\hat{h}_i(x) = f_{opt}^{(i)}(x)$ in the j^{th} iteration of $\hat{H}_i(x)$, and 0 otherwise. Recall that $\hat{H}_i(x) = f_{opt}^{(i)}(x)$ iff the former event holds in a majority of trials $j \in [\delta(n)^{-8}]$. Let

$$Z := \sum_{j\in[\delta(n)^{-8}]} z_j,$$

and note that, by Eq. (3),

$$\mathbb{E}[Z] = \delta(n)^{-8} \cdot \alpha_i(x) \ge \delta(n)^{-8} \cdot \left(\frac{1}{2} + \delta(n)\right).$$

Then, by a Chernoff bound,

$$\begin{aligned}
\Pr\Big[\hat{H}_i(x) \ne f_{opt}^{(i)}(x)\Big] &= \Pr\left[Z \le \delta(n)^{-8}/2\right] \\
&\le \Pr\left[Z \le \mathbb{E}[Z] \cdot (1 - \delta(n)^3)\right] \\
&\le 2^{-\Omega(n)}.
\end{aligned}$$

Case II: $\alpha_i(x) < 1/2 + \delta(n)^2$. Now,

$$\Pr_{y\sim Y_i^x}[\hat{H}_i(x) \neq y] - \Pr_{y\sim Y_i^x}[f_{opt}^{(i)}(x) \neq y] \leq \frac{1}{2} - (1 - \alpha_i(x)) \leq \delta(n)^2.$$

Thus, in both Cases I and II, it holds that

$$\Pr_{y\sim Y_i^x}[\hat{H}_i(x) \neq y] - \Pr_{y\sim Y_i^x}[f_{opt}^{(i)}(x) \neq y] \leq \delta(n)^2.$$

Moreover,

$$\Delta(\hat{H}_i(x), H_i(x)) \leq \delta(n)^{-8} \cdot \delta(n)^{10} = \delta(n)^2.$$

for $x \sim X_i$. Overall, by a union bound and the definition of δ,

$$\Pr_A[A(\varphi) = \mathsf{SAT}(\varphi)] \geq \frac{2}{3} - 2 \cdot t_R(n) \cdot \delta(n)^2 - o(1) = \frac{2}{3} - o(1),$$

as desired. ◀

We are now ready to finish the proof of Theorem 15, restated below.

▶ **Theorem 73** (Restatement of Theorem 15)**.** *Suppose there is a randomized non-adaptive honest reduction R and a polynomial $s : \mathbb{N} \to \mathbb{N}$ such that, for every constant $c \in \mathbb{N}$, R reduces* SAT *to agnostically PAC learning* SIZE$[s(n)]$ *by* SIZE$[s(n)^c]$*. Then, unless* NP $\subseteq$ BPP*, there exist infinitely-often one-way functions.*

Proof. Assume the non-existence of i.o. one-way functions. Let R be the assumed randomized non-adaptive reduction from SAT to agnostic learning SIZE$[s]$ running in polynomial time t_R. Let the inverse polynomial ε be such that R requires (s, ε)-optimal hypotheses in response to each query, and let the constant $b \in \mathbb{N}$ be as in Lemma 72. Combining Lemma 70 with Lemma 51, for every distribution family $D \in$ PSAMP, there exists a PPT machine I such that, for all sufficiently large $n \in \mathbb{N}$,

$$\Pr_{z\sim D_n}[\ I \text{ is an } n^{-b}\text{-distributional inverter for } f_z\ ] \geq 1 - \frac{1}{n^b},$$

and if I does not n^{-b}-invert f_z for some $z \in \{0,1\}^*$, then for any $y \in \{0,1\}^*$,

$$\Pr_I[\ I(z, y) \text{ outputs } \perp\ ] \geq 1 - 2^{-n}.$$

Along with Lemma 72, this implies DistNP $\subseteq$ AvgBPP. Let B be the PPT agnostic learner for SIZE$[s]$ guaranteed by Lemma 71, with accuracy and confidence parameters δ chosen such that $\delta^{-1}(n) \geq \max\{t_R(n)^2, \varepsilon^{-1}(n)\}$ for all $n \in \mathbb{N}$.

Now consider the PPT machine R^B defined as follows.

> On input $\varphi \in \{0,1\}^n$ and randomness $r_0 \sim \mathcal{U}_{t_R(n)}$, simulate $R(\varphi; r_0)$, obtaining queries $\mathcal{E}_1, ..., \mathcal{E}_m$, where each $\mathcal{E}_i$ samples a distribution (X_i, Y_i). For each such $i \in [m]$, let h_i be the output of B on input $\mathcal{E}_i$. Answer the i^{th} query of R with h_i. Accept iff R accepts.

By Lemma 71, for all sufficiently large $n \in \mathbb{N}$, for each $i \in [m]$, with probability $1 - \delta(n)$ over the randomness of B, B returns an (s, ε)-optimal hypothesis h_i for $\mathcal{E}_i$.

By a union bound and the definition of R,

$$\Pr_{R^B}[R^B(\varphi) = \mathsf{SAT}(\varphi)] \geq \frac{2}{3} - \frac{1}{t_R(n)}.$$

We conclude that SAT $\in$ BPP, as desired. ◀

3.4.2 Proofs of Theorems 16 and 17

We will need the following lemma. For the reader's convenience, we give a sketch of the proof in Appendix B. Though not originally stated for the infinitely-often setting, it is easily seen to follow from the same argument.

▶ **Lemma 74** ([11]). *Suppose* BPP*-secure witness encryption exists for* NP*. Then, if infinitely-often one-way functions exist, there exists public-key encryption secure infinitely often against polynomial-time adversaries.*

Theorem 16 is now immediate from Theorem 4, Lemma 33, Theorem 15, and Lemma 74. Theorem 17 follows from Theorem 4 and Theorem 16.

4 Concluding Remarks

We have shown that the existence of secure witness encryption is equivalent to NP-hardness of a certain learning problem (CGL) under restricted deterministic reductions. This may be taken as evidence that computational learning is indeed NP-hard, if one believes in cryptography. At the same time, this result also suggests that one may want to consider a more general kind of reductions when trying to prove NP-hardness of learning, as establishing the existence of secure witness encryption seems hard. On the other hand, actually proving NP-hardness of learning for restricted deterministic reductions would imply a breakthrough result that public-key cryptography can be based on the worst-case assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

There are many interesting research directions to explore in the context of cryptography and NP-hardness of meta-complexity problems. For example, Mazor and Pass [38] have recently showed that the existence of secure Indistinguishability Obfuscation (a cryptographic primitive that implies Witness Encryption) would mean that a polynomial-gap version of MCSP cannot be shown NP-complete under Levin reductions, unless $\mathsf{NP} \subseteq \mathsf{BPP}$. Can one get similar results for some computational learning problems? What if one has secure Witness Encryption only? Another possible research direction is to find out whether existing tools from NP-hardness of learning could unconditionally yield new non-trivial forms of witness encryption (eg. with security against restricted circuit classes).

Lastly, of course, the NP-hardness of improper learning for P/poly remains open. We have made some partial progress, and we have demonstrated cryptographic consequences of NP-hardness under certain strong formulations, but we are optimistic that further progress can be made.

References

1 Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is np-hard to linearly approximate. *J. Symb. Log.*, 66(1):171–191, 2001. doi:10.2307/2694916.

2 Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 211–220. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.35.

3 Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012. doi:10.1145/2160158.2160159.

4 Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology – Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011. doi:10.1007/978-3-642-20901-7_2.

5 Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988. doi:10.1007/0-387-34799-2_3.

6 Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 278–291, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

7 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Agnostic learning from tolerant natural proofs. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 35:1–35:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.APPROX-RANDOM.2017.35.

8 Irit Dinur, Prahladh Harsha, and Guy Kindler. Polynomially low error pcps with polyloglog n queries via modular composition. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 267–276. ACM, 2015. doi:10.1145/2746539.2746630.

9 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004. doi:10.1016/J.IPL.2003.11.007.

10 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016. doi:10.1137/14095772X.

11 Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 467–476. ACM, 2013. doi:10.1145/2488608.2488667.

12 Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Probabilistic kolmogorov complexity with applications to average-case complexity. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 16:1–16:60. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.CCC.2022.16.

13 Dan Gutfreund and Amnon Ta-Shma. Worst-case to average-case reductions revisited. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, volume 4627 of *Lecture Notes in Computer Science*, pages 569–583. Springer, 2007. doi:10.1007/978-3-540-74208-1_41.

14 Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.

15 Shuichi Hirahara. Non-disjoint promise problems from meta-computational view of pseudorandom generator constructions. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 20:1–20:47. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.20.

16 Shuichi Hirahara. Np-hardness of learning programs and partial MCSP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022*, pages 968–979. IEEE, 2022. doi:10.1109/FOCS54457.2022.00095.

17 Shuichi Hirahara. Capturing one-way functions via np-hardness of meta-complexity. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1027–1038. ACM, 2023. doi:10.1145/3564246.3585130.

18 Shuichi Hirahara and Mikito Nanashima. Learning in pessiland via inductive inference. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 447–457, 2023. doi:10.1109/FOCS57990.2023.00033.

19 Shuichi Hirahara and Mikito Nanashima. One-way functions and zero knowledge. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1731–1738. ACM, 2024. doi:10.1145/3618260.3649701.

20 Justin Holmgren and Ron Rothblum. Linear-size boolean circuits for multiselection. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 11:1–11:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.CCC.2024.11.

21 Yizhi Huang, Rahul Ilango, and Hanlin Ren. Np-hardness of approximating meta-complexity: A cryptographic approach. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1067–1075. ACM, 2023. doi:10.1145/3564246.3585154.

22 Russell 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*, pages 134–147. IEEE Computer Society, 1995. doi:10.1109/SCT.1995.514853.

23 Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. Synergy between circuit obfuscation and circuit minimization. In Nicole Megow and Adam D. Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA*, volume 275 of *LIPIcs*, pages 31:1–31:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.APPROX/RANDOM.2023.31.

24 Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 812–821. IEEE Computer Society, 1990. doi:10.1109/FSCS.1990.89604.

25 Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October – 1 November 1989*, pages 230–235. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63483.

26 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229. ACM, 1997. doi:10.1145/258533.258590.

27 Mitsuru Ito, Akira Saio, and Takao Nishizeki. Multiple assignment scheme for sharing secret. *J. Cryptol.*, 6(1):15–20, 1993. doi:10.1007/BF02620229.

28 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. *Commun. ACM*, 67(3):97–105, 2024. doi:10.1145/3611095.

29 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79. ACM, 2000. doi:10.1145/335305.335314.

30 Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. *Mach. Learn.*, 17(2-3):115–141, 1994. doi:10.1007/BF00993468.

31 Ker-I Ko. On the complexity of learning minimum time-bounded turing machines. *SIAM J. Comput.*, 20(5):962–986, 1991. doi:10.1137/0220059.

32 Caleb Koch, Carmen Strassle, and Li-Yang Tan. Properly learning decision trees with queries is np-hard. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 2383–2407. IEEE, 2023. doi:10.1109/FOCS57990.2023.00146.

33 Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 374–383. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.47.

34 Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 – 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 254–273. Springer, 2014. doi:10.1007/978-3-662-45608-8_14.

35 Yanyi Liu, Noam Mazor, and Rafael Pass. A note on zero-knowledge for NP and one-way functions. *Electron. Colloquium Comput. Complex.*, TR24-095, 2024. URL: https://eccc.weizmann.ac.il/report/2024/095.

36 Yanyi Liu, Noam Mazor, and Rafael Pass. On witness encryption and laconic zero-knowledge arguments. *Electron. Colloquium Comput. Complex.*, TR24-194, 2024. URL: https://eccc.weizmann.ac.il/report/2024/194.

37 Zhenjian Lu, Noam Mazor, Igor C. Oliveira, and Rafael Pass. Lower bounds on the overhead of indistinguishability obfuscation. *Electron. Colloquium Comput. Complex.*, TR24-146, 2024. URL: https://eccc.weizmann.ac.il/report/2024/146.

38 Noam Mazor and Rafael Pass. Gap MCSP is not (levin) np-complete in obfustopia. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 36:1–36:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.CCC.2024.36.

39 Mikito Nanashima. A theory of heuristic learnability. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3483–3525. PMLR, August 2021. URL: https://proceedings.mlr.press/v134/nanashima21a.html.

40 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

41 Igor C. Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 18:1–18:49. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.CCC.2017.18.

42 Shien Jin Ong and Salil P. Vadhan. Zero knowledge and soundness are symmetric. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, volume 4515 of *Lecture Notes in Computer Science*, pages 187–209. Springer, 2007. doi:10.1007/978-3-540-72540-4_11.

43 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988. doi:10.1145/48014.63140.

44 Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002. doi:10.1006/JCSS.2002.1824.

45 Rahul Santhanam. Pseudorandomness and the minimum circuit size problem. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 68:1–68:26. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.ITCS.2020.68.

46 Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

47 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91. IEEE Computer Society, 1982. doi:10.1109/SFCS.1982.45.

A WE from Indistinguishability Obfuscation

In this section, we give a sketch of the construction showing that indistinguishability obfuscation for P/poly implies witness encryption for NP.

▶ **Definition 75** (Indistinguishability obfuscation (iO)). *A PPT algorithm $i\mathcal{O}$ is an* indistinguishability obfuscator *for a circuit class $\mathcal{C}$ if the following conditions are met.*

- ***Correctness:*** *for all $C \in \mathcal{C}$ and all inputs x, letting $C' = i\mathcal{O}(C)$, it holds that $C'(x) = C(x)$.*
- ***Security:*** *for any efficient adversary A, for any circuits C_0 and C_1 computing the same function, $i\mathcal{O}(C_0)$ and $i\mathcal{O}(C_1)$ are indistinguishable for A.*

▶ **Theorem 76** ([10]). *For any $L \in \mathsf{NP}$, iO for $\mathsf{P/poly}$ implies WE for L.*

Proof sketch. Let V be a poly-time verifier for L. For an instance x of L and a bit $b \in \{0,1\}$, define

$$F_{x,b}(w) = \begin{cases} b & V(x,w) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Consider the witness encryption scheme defined as follows:

- $\mathsf{Enc}(x,b) = i\mathcal{O}(F_{x,b})$;
- $\mathsf{Dec}(c,x,w) = c(w)$.

By the correctness of the iO, if $V(x,w) = 1$, then for $C := \mathsf{Enc}(x,b) = i\mathcal{O}(F_{x,b})$, it holds that $\mathsf{Dec}(C,x,w) = C(w) = b$. Therefore, the correctness of WE is satisfied.

We leave the proof of security to the reader, but we note that if $x \notin L$, then $F_{x,0}$ and $F_{x,1}$ are the same function. ◀

B Public-key Encryption from WE and OWFs

In this section, we give a sketch of the construction showing that witness encryption, together with the existence of a one-way function, implies the possibility of public-key encryption.

▶ **Definition 77** (Public-key encryption). *A* public-key encryption scheme *is a triple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *meeting the following conditions.*

- ***Correctness:*** *for all sufficiently large $\lambda \in \mathbb{N}$ and any bit $b \in \{0,1\}$,*

$$\Pr_{(sk,pk)\sim\mathsf{Gen}(1^\lambda)}[\,\mathsf{Dec}(sk,\mathsf{Enc}(pk,b)) = b\,] = 1.$$

- ***Security:*** *for all sufficiently large $\lambda \in \mathbb{N}$ and any efficient adversary A,*

$$\Pr_{(sk,pk)\sim\mathsf{Gen}(1^\lambda)\,;\,b\sim\mathcal{U}}[\,A(pk,\mathsf{Enc}(pk,b)) = b\,] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

▶ **Theorem 78** ([11]). *Suppose that WE exists for every language in NP and that a one-way function exists. Then, public-key encryption exists.*

Proof sketch. By [14], we may assume the existence of a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ for all sufficiently large $\lambda \in \mathbb{N}$. Consider the language

$$L = \{t \in \{0,1\}^{2\lambda} \mid \exists s \in \{0,1\}^\lambda,\ t = G(s)\},$$

and let $(\mathsf{Enc}_{WE}, \mathsf{Dec}_{WE})$ be a witness encryption scheme for L. Define a public-key encryption scheme as follows.

- $\mathsf{Gen}(1^\lambda)$: sample $s \sim \mathcal{U}_\lambda$ and let $t = G(s)$. Let $sk = s$ and $pk = t$.
- $\mathsf{Enc}(pk, b)$: return ($\mathsf{Enc}_{WE}(pk, b)$, pk).
- $\mathsf{Dec}(sk, c' = (c, pk))$: return the output of $\mathsf{Dec}_{WE}(c, pk, sk)$.

By the correctness of the witness encryption, for any pair $(sk, pk) = (s, t)$ in the support of Gen, it holds that s is a witness for $t \in L$. Therefore, $\mathsf{Dec}(sk, \mathsf{Enc}(pk, b)) = \mathsf{Dec}_{WE}(\mathsf{Enc}_{WE}(t, b), t, s) = b$. This shows the correctness of the PKE scheme.

We leave the proof of security to the reader. ◀