On Basing Auxiliary-Input Cryptography on NP-Hardness via Nonadaptive Black-Box Reductions

Mikito Nanashima
Tokyo Institute of Technology, Japan
https://nanashima.github.io
nanashima.m.aa@is.c.titech.ac.jp

Abstract

Constructing one-way functions based on NP-hardness is a central challenge in theoretical computer science. Unfortunately, Akavia et al. [2] presented strong evidence that a nonadaptive black-box (BB) reduction is insufficient to solve this challenge. However, should we give up such a central proof technique even for an intermediate step?

In this paper, we turn our eyes from standard cryptographic primitives to weaker cryptographic primitives allowed to take auxiliary-input and continue to explore the capability of nonadaptive BB reductions to base auxiliary-input primitives on NP-hardness. Specifically, we prove the followings:

- if we base an auxiliary-input pseudorandom generator (AIPRG) on NP-hardness via a nonadaptive BB reduction, then the polynomial hierarchy collapses;
- if we base an auxiliary-input one-way function (AIOWF) or auxiliary-input hitting set generator (AIHSG) on NP-hardness via a nonadaptive BB reduction, then an (i.o.-)one-way function also exists based on NP-hardness (via an adaptive BB reduction).

These theorems extend our knowledge on nonadaptive BB reductions out of the current worst-to-average framework. The first result provides new evidence that nonadaptive BB reductions are insufficient to base AIPRG on NP-hardness. The second result also yields a weaker but still surprising consequence of nonadaptive BB reductions, i.e., a one-way function based on NP-hardness. In fact, the second result is interpreted in the following two opposite ways. Pessimistically, it shows that basing AIOWF or AIHSG on NP-hardness via nonadaptive BB reductions is harder than constructing a one-way function based on NP-hardness, which can be regarded as a negative result. Note that AIHSG is a weak primitive implied even by the hardness of learning; thus, this pessimistic view provides conceptually stronger limitations than the currently known limitations on nonadaptive BB reductions. Optimistically, it offers a new hope: breakthrough construction of auxiliary-input primitives might also provide construction standard cryptographic primitives. This optimistic view enhances the significance of further investigation on constructing auxiliary-input or other intermediate cryptographic primitives instead of standard cryptographic primitives.

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

Keywords and phrases Auxiliary-input cryptographic primitives, nonadaptive black-box reductions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.29

Related Version A full version of the paper is available at https://eccc.weizmann.ac.il/report/2020/095/.

Funding *Mikito Nanashima*: JST, ACT-X Grant Number JPMJAX190M, Japan.

Acknowledgements The author thanks Toshiya Itoh and the anonymous reviewers for many helpful comments.

© Mikito Nanashima;
licensed under Creative Commons License CC-BY
12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 29; pp. 29:1–29:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

How can we translate computational hardness into cryptography? This is a central question in theoretical computer science. Specifically, one of the most significant and long-standing challenges on this question is constructing fundamental cryptographic primitives such as a one-way function based on NP-hardness. At the moment, several breakthroughs seem to be required for solving this challenge, as surveyed by Impagliazzo [20].

A central ingredient for solving the above challenge is a *reduction*; in other words, the way to translate recognizing a language into breaking a cryptographic primitive. A reduction is a powerful proof technique even if it is restricted to a quite simple form, and in fact, a nonadaptive black-box (BB) reduction is sufficient to show many brilliant results and has played a crucial role in theoretical computer science. Therefore, it is a natural attempt to apply such a familiar proof technique even for constructing secure cryptographic primitives.

However, Akavia et al. [10] presented strong evidence that such a simple reduction is insufficient for cryptography based on NP-hardness. Generally, breaking cryptographic primitives is formulated as an NP problem on an efficiently samplable distribution that is fixed in advance. They showed that there is no nonadaptive BB reduction from an NP-hard problem to such a distributional NP problem unless the polynomial hierarchy collapses. As a corollary, their work excluded the attempt to apply nonadaptive BB reductions for cryptography based on NP-hardness under the reasonable assumption that the polynomial hierarchy does not collapse. Further, subsequent work provided stronger consequences in more specific cases of several cryptographic primitives [2, 12, 5, 13, 9, 8, 27, 17].

Then should we also give up all nonadaptive BB strategies even for an intermediate step towards cryptography? This question originally motivated our work. In this spirit, we focus on the capability of nonadaptive BB reduction for a weaker cryptographic notion, i.e., an *auxiliary-input* cryptographic primitive introduced first by Ostrovsky and Wigderson [30]. Informally speaking, an auxiliary-input cryptographic primitive is defined as a family of primitives indexed by the auxiliary-input and has a relaxed security requirement: at least one primitive in the family must be secure depending on each adversary (instead of one specific primitive secure against all adversaries). In other words, adversaries for auxiliary-input primitives must break all primitives in the worst-case sense on auxiliary-input. This task is not directly formulated as a distributional NP-problem because the distribution is not uniquely determined in advance due to auxiliary-input. Thus, the previous work on distributional NP problem cannot be directly applied to auxiliary-input cryptography.

Herein, we present the current status of nonadaptive BB reductions to auxiliary-input cryptography. Applebaum et al. [5] observed that nonadaptive fixed-auxiliary-input BB reductions are insufficient even for auxiliary-input cryptography unless the polynomial hierarchy collapses. Their reduction is a restricted case of nonadaptive BB reduction where only one auxiliary-input is accessible. However, this restricted access to auxiliary-input seems too strict and implicitly yields a reduction from an NP-hard language to some fixed cryptographic primitive (depending on the instance). In fact, this result was shown in almost the same way to the previous result for standard cryptographic primitives in [2]. The same work and later Xiao [34] observed that generalizing their result to nonadaptive BB reductions seems hard by giving the explicit technical issue. To the best of our knowledge, we had no negative result on general nonadaptive BB reductions to base auxiliary-input cryptography on NP-hardness before this work. For more detailed reason why the previous work such as [10, 2] is not applicable for auxiliary-input primitives, refer to Section 4.

The recent progress on the minimum circuit size problem revealed that an auxiliary-input one-way function indeed implies a hard-on-average distributional NP problem [3, 16]. However, such an implication requires adaptive techniques at present (e.g., [15]). Thus, the property of nonadaptive black-box is lost in translating reductions for the auxiliary-input primitive into reductions for the distributional NP problem.

In this paper, based on the above status, we continue to investigate the capability of nonadaptive BB reductions for auxiliary-input cryptographic primitives based on NP-hardness. The importance of our work is to extend our current knowledge on such a central proof technique out of the previous worst-to-average framework in [10] and to identify the inherent difficulty on constructing cryptographic primitives on NP-hardness more finely.

1.1 Our Contribution

Our main contribution is to provide new knowledge about nonadaptive BB reductions from an NP-hard problem to an auxiliary-input cryptographic primitive. In particular, we handle the auxiliary-input analogs of the following three fundamental primitives: a one-way function, a pseudorandom generator, and a hitting set generator. A definition of each primitive will be presented in Section 2 with a formal description of our main results. First, we informally state the main theorem as follows.

▶ **Theorem** (informal). *If there is a nonadaptive BB reduction from an* NP*-hard language L to breaking an auxiliary-input cryptographic primitive P, then the following statements hold according to the type of P:*

- *if P is an auxiliary-input pseudorandom generator, then the polynomial hierarchy collapses;*
- *if P is an auxiliary-input one-way function or an auxiliary-input hitting set generator, then there is also an adaptive reduction from L to inverting some (i.o.-)one-way function.*

The first result provides reasonable evidence that auxiliary-input pseudorandom generators (AIPRG) cannot be based on NP-hardness via nonadaptive BB reductions as standard cryptography. The second result shows that a nonadaptive BB reduction for basing the other auxiliary-input primitives yields another strong consequence: an "infinitely often" analog of one-way function based on NP-hardness. Note that an auxiliary-input hitting set generator (AIHSG) is much weaker primitive than standard cryptographic primitives: for example, the existence is even weaker than the hardness of PAC learning [28]. What is surprising is that even a nonadaptive BB reduction to such a weak primitive yields a solution close to the long-standing challenge, i.e., characterization of one-way functions based on NP-hardness.

The second result is not sufficient to exclude nonadaptive BB reductions which base auxiliary-input primitives on NP-hardness, and it has two opposite interpretations. However, let us stress that both interpretations are quite nontrivial and yield new knowledge about nonadaptive BB reductions. One interpretation is a pessimistic (or realistic) one. As mentioned in the introduction, no one has not come up with the construction of a one-way function based on NP-hardness for several decades despite its importance. Thus, this result is still strong evidence of difficulty finding such a simple reduction. The other interpretation is an optimistic one as a new approach to constructing a one-way function. We will further discuss this optimistic perspective and its novelty in Section 3.

A reader who is familiar with cryptography may wonder why the consequences are different between an auxiliary-input one-way function (AIOWF) and AIPRG. In fact, AIPRG is constructed from any AIOWF by applying the known BB construction of a pseudorandom generator from a one-way function. However, if such construction requires an adaptive security proof, then the property of nonadaptive is lost in translating reductions for AIOWF into reductions for AIPRG via the adaptive security reduction. To the best of our knowledge, all currently known constructions of pseudorandom generators (e.g., [15, 19, 14]) use adaptive techniques in the security proof; for instance, construction of false entropy generators and the uniform hardcore lemma [18]. This technical issue prevents us from applying the first result for AIPRG to AIOWF. For a similar reason, our second result on AIOWF is incomparable with the previous work [5] on hardness of learning because we need to construct AIPRG first to show the hardness of learning from the existence of AIOWF.

2 Formal Descriptions

In this section, we present formal descriptions of auxiliary-input primitives and our results. Let us introduce a few notations. For any $n \in \mathbb{N}$, let U_n denote a random variable selected according to a uniform distribution over $\{0,1\}^n$. For any function $f : \mathcal{X} \to \mathcal{Y}$ and subsets $X \subseteq \mathcal{X}$, $Y \subseteq \mathcal{Y}$, let $f(X) = \{f(x) : x \in X\}$ and $f^{-1}(Y) = \{x \in X : f(x) \in Y\}$. For a language L, let (L, U) denote a distributional problem of recognizing $L(x)$ on an instance x selected uniformly at random. An auxiliary-input cryptographic primitive is defined as an auxiliary-input function with some additional security conditions.

▶ **Definition 1** (Auxiliary-input function). *A (polynomial-time computable) auxiliary-input function is a family $f = \{f_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(|z|)}\}_{z\in\{0,1\}^*}$, where $n(|z|)$ and $\ell(|z|)$ are polynomially-related*[1] *to $|z|$, which satisfies that there exists a polynomial-time evaluation algorithm F such that for any $z \in \{0,1\}^*$ and $x \in \{0,1\}^{n(|z|)}$, $F(z,x)$ outputs $f_z(x)$.*

In this paper, we use the term "an auxiliary-input function (AIF)" to refer to polynomial-time computable one as in the above definition unless otherwise stated. For the sake of simplicity, we assume that $n(\cdot)$ and $\ell(\cdot)$ are increasing functions. Note that the length of auxiliary-input is possibly longer than the length of input and output, i.e., $|z| > n(|z|)$ and $|z| > \ell(|z|)$. We may write $n(|z|)$ (resp. $\ell(|z|)$) as n (resp. ℓ) when the dependence of $|z|$ is obvious.

2.1 Auxiliary-Input Pseudorandom Generator

A pseudorandom generator is a primitive stretching a short random seed to a long binary string random-looking from all efficiently computable adversaries. The auxiliary-input analog is formally defined as follows:

▶ **Definition 2** (Auxiliary-input pseudorandom generator). *Let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. For a function $\gamma : \mathbb{N} \to (0,1)$, we say that a randomized algorithm A γ-distinguishes G if for all auxiliary-inputs $z \in \{0,1\}^*$,*

$$\left| \Pr_{A,U_n}[A(z, G_z(U_n)) = 1] - \Pr_{A,U_{\ell(n)}}[A(z, U_{\ell(n)}) = 1] \right| \geq \gamma(n).$$

We say that G is an auxiliary-input pseudorandom generator (AIPRG) if $\ell(n) > n$ and for all polynomials p, there exists no polynomial-time randomized algorithm $(1/p)$-distinguishing G.

A BB reduction for AIPRG is defined as follows. It is easily verified that the following BB reduction from a language L to distinguishing an AIF G shows that G is an AIPRG if $L \notin \mathsf{BPP}$.

[1] In the case of $n(|z|)$, it means that there exist $c, c' \in \mathbb{N}$ such that $|z| \leq c \cdot n(|z|)^c$ and $n(|z|) \leq c' \cdot |z|^{c'}$.

▶ **Definition 3** (Black-box reduction to distinguishing AIF). *Let L be a language and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function with $\ell(n) > n$. We say that there exists a black-box (BB) reduction from L to distinguishing G if for all polynomials p, there exists a randomized polynomial-time oracle machine $R^?$ such that for all oracles $\mathcal{O}$ that $(1/p)$-distinguish G and $x \in \{0,1\}^*$, R satisfies that*

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that there exists a nonadaptive BB reduction from L to distinguishing G if all R make their queries independently of any answer by oracle for previous queries.

The first main result on AIPRG is stated as follows.

▶ **Theorem 4.** *For any auxiliary-input function $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ with $\ell(n) > n$, there exists no nonadaptive BB reduction from an* NP*-hard language L to distinguishing G unless the polynomial hierarchy collapses.*

2.2 Auxiliary-Input One-Way Function

A one-way function is a function which is easy to compute but hard to invert, and it is a fundamental primitive in the sense that most cryptographic tools do not exist without a one-way function [23, 32]. The formal definition is the following:

▶ **Definition 5** (One-way function). *Let s, ℓ be polynomials. We say that a family of function $f = \{f_n\}_{n\in\mathbb{N}}$ where $f_n : \{0,1\}^{s(n)} \to \{0,1\}^{\ell(n)}$ is an (i.o.-)one-way function (OWF)[2] if f is polynomial-time computable, and there exists a polynomial p such that for all polynomial-time randomized algorithms A, there exist infinitely many $n \in \mathbb{N}$ such that*

$$\Pr_{A,U_{s(n)}}[A(1^n, f_n(U_{s(n)})) \notin f_n^{-1}(f_n(U_{s(n)}))] \geq 1/p(n).$$

For the sake of simplicity, we may omit to write the input 1^n to A.

The auxiliary-input analog of OWF, first introduced by Ostrovsky and Wigderson [30], is defined as follows.

▶ **Definition 6** (Auxiliary-input one-way function). *Let $f = \{f_z : \{0,1\}^n \to \{0,1\}^{\ell}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function and $\gamma : \mathbb{N} \to (0,1)$ be a function. We say that a randomized algorithm A γ-inverts f if for all $z \in \{0,1\}^*$,*

$$\Pr_{A,U_n}[A(z, f_z(U_n)) \in f_z^{-1}(f_z(U_n))] \geq \gamma(n).$$

We say that f is an auxiliary-input one-way function (AIOWF) if there exists a polynomial p such that no polynomial-time randomized algorithm $(1-1/p)$-inverts f.

In fact, the existence of AIOWF and AIPRG is equivalent [15]. However, we cannot directly apply Theorem 4 to AIOWF due to the adaptive security reduction, as we mentioned in Section 1.1.

A BB reduction for AIOWF is defined as follows. It is easily verified that for any polynomial p, the following BB reduction from a language L to $(1-1/p)$-inverting an AIF f shows that f is an AIOWF if $L \notin$ BPP.

[2] Strictly speaking, a one-way function defined in Definition 5 is usually called a "weak" one-way function, which implies the standard (strong) one-way function.

▶ **Definition 7** (Black-box reduction to inverting AIF). *Let L be a language, p be a polynomial, and $f := \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. We say that a randomized polynomial-time oracle machine $R^?$ is a black-box (BB) reduction from L to $(1-1/p)$-inverting f if for all oracles $\mathcal{O}$ that $(1-1/p)$-invert f and $x \in \{0,1\}^*$, R satisfies that*

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that R is nonadaptive if all R's queries are made independently of any answer by oracle for previous queries.

The second main result on AIOWF is stated as follows.

▶ **Theorem 8.** *For any auxiliary-input function $f = \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ and polynomial p, if there exists a nonadaptive BB reduction from an* NP*-hard language L to $(1-1/p)$-inverting f, then* NP $\not\subseteq$ BPP *also implies that a one-way function exists (via an adaptive BB reduction).*

2.3 Auxiliary-Input Hitting Set Generator

A hitting set generator is a weak variant of a pseudorandom generator, introduced in the context of derandomization by Andreev et al. [4]. For the original purpose, they considered (possibly) exponential-time computable generators. In this paper, we focus on polynomial-time computable generators as in cryptography. We define the auxiliary-input analog as follows.

▶ **Definition 9** (Auxiliary-input hitting set generator). *Let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. For a function $\gamma : \mathbb{N} \to (0,1)$, we say that a randomized adversary A γ-avoids G if for all (public) auxiliary-inputs $z \in \{0,1\}^*$ and (private) inputs $x \in \{0,1\}^{n(|z|)}$,*

$$\Pr_A[A(z, G_z(x)) = 0] \geq 2/3 \ \ \textit{and} \ \ \Pr_{y\sim\{0,1\}^{\ell(n(|z|))}}\left[\Pr_A[A(z,y) = 1] \geq 2/3\right] \geq \min(\gamma(n), \tau_z),$$

where τ_z be a trivial limitation[3] defined as $\tau_z = 1 - \frac{|G_z(\{0,1\}^n)|}{2^{\ell(n)}}$.

We say that G is a γ-secure auxiliary-input hitting set generator (AIHSG) if $\ell(n) > n$ and there exists no polynomial-time randomized algorithm $(1-\gamma)$-avoiding G.

Although it is easily verified that AIPRG is also AIHSG (for any security $\gamma(n) = 1/\mathsf{poly}(n)$), the opposite direction is open at present. In fact, the hardness of learning implies the existence of AIHSG [28]; on the other hand, we must overcome the barrier by oracle separation to show the existence of AIPRG (equivalently, AIOWF) from the hardness of learning [35]. Thus, AIHSG seems to be a much weaker notion than AIOWF and AIPRG under current knowledge.

A BB reduction for AIHSG is defined as follows. It is easily verified that the following BB reduction from a language L to $(1-\gamma)$-avoiding an AIF G shows that G is a γ-secure AIHSG if $L \notin$ BPP.

[3] In this paper, we consider general settings of γ and ℓ. Thus, we adopted the trivial limitation in the definition to avoid arguing about invalid settings where γ-avoiding the generator is impossible by definition.

▶ **Definition 10** (Black-box reduction to avoiding AIF). *Let L be a language, γ be a function, and $G := \{G_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. We say that a randomized polynomial-time oracle machine $R^?$ is a black-box (BB) reduction from L to $(1-\gamma)$-avoiding G if for all oracles $\mathcal{O}$ that $(1-\gamma)$-avoid G and $x \in \{0,1\}^*$, R satisfies that*

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that R is nonadaptive if all R's queries are made independently of any answer by oracle for previous queries.

The third main result on AIHSG is stated as follows.

▶ **Theorem 11.** *Let p be a polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function where $\ell(n) > (1+\epsilon)\cdot n$ for some constant $\epsilon > 0$. If there exists a nonadaptive BB reduction from an* NP*-hard language L to $(1-1/p)$-avoiding G, then* NP $\not\subseteq$ BPP *also implies that a one-way function exists (via an adaptive BB reduction).*

3 Discussion and Future Directions

As discussed in Section 1.1, Theorems 8 and 11 are also regarded as approaches to construct one-way functions based on NP-hardness. In this section, we discuss the novelty of this optimistic perspective and suggest future directions, including the investigation of the validity.

Our results are rephrased as follows: Assume that we could connect NP-hardness to some auxiliary-input primitives (i.e., AIOWF or AIHSG) via a novel nonadaptive BB reduction, then we can automatically extend the connection to standard cryptographic primitives, that is, OWF. At present, the latter task of removing auxiliary-input from primitives seems quite non-trivial, as mentioned in [5, 33]. In this paper, we also provide a simple oracle separation between AIOWF and OWF as follows. This indicates that we cannot expect any relativized technique to remove auxiliary-input from cryptographic primitives.

▶ **Theorem 12.** *There exists an oracle $\mathcal{O}$ such that relative to $\mathcal{O}$ an auxiliary-input one-way function exists, but a one-way function does not exist.*

Additionally, there are several barriers by other oracle separations at the intermediate levels to base OWF on NP-hardness (e.g., [35, 21]). Although such barriers on relativization are common throughout theoretical computer science (e.g., the P vs. NP problem [6]), there are only a few success stories of overcoming such barriers at present. Unfortunately, Theorems 8 and 11 do not provide any solution to break these barriers, and a new non-relativized technique is still required. Specifically, if a nonadaptive BB reduction to AIOWF or AIHSG is also relativized[4], then our proof also yields relativized reductions that contradict Theorem 12 or the oracle separation presented in [21].

However, our result offers one hope. Although there seems to be several barriers towards cryptography based on NP-hardness as discussed above, the essential barrier we must overcome might be few. Theorems 8 to 12 certainly show that if we could find a non-relativized breakthrough at an intermediate level toward cryptography (that is, auxiliary-input primitives), then it will be lifted and break the other barriers at the higher level. From this perspective, we conjecture that the difficulty in basing OWF on NP-hardness could rely on

[4] Note that oracle separations do not necessarily rule out BB reductions from particular languages, not as fully BB reductions defined in [31].

a much smaller part of tasks at an intermediate level. This conjecture seems somewhat controversial but enhances the significance of further investigation on auxiliary-input or other intermediate cryptographic primitives instead of standard ones.

The above discussion leads to the following two possible directions. The first direction is to find other scenarios where a breakthrough at an intermediate level also brings benefits at the higher level. This direction might reduce constructing standard cryptographic primitives to the task at the low level and give new insights into complexity-based cryptography. The second direction is to refute such an attempt on intermediate primitives with convincing evidence if it gives the wrong direction. Particularly, in our case, there is a possibility that nonadaptive BB reductions for AIOWF and AIHSG indeed yield the collapse of the polynomial-hierarchy as in the case of AIPRG.

For the second direction, we list two concrete ways: (1) finding a new construction of AIPRG from AIOWF with nonadaptive security proof; (2) generalizing the previous results for OWF [2] or HSG [17] to each auxiliary-input analog for the stronger consequence. At least the latter approach seems to require some new technique to simulate nonadaptive BB reductions by constant-round interactive proof systems, as observed in [5] and [34].

4 A First Attempt: Applying [10] and [2]

Before presenting our proof strategies, we roughly explain why the previous technique developed by Bogdanov and Trevisan [10] for the worst-to-average framework is not applicable in the case of auxiliary-input cryptography. For the sake of simplicity, we assume that there exists a nonadaptive BB reduction R from an NP-hard language L to a distributional NP-problem (L', U), and R makes queries of the same length n determined by the size of input to R. Note that if we can answer these queries by an oracle which correctly recognizes L' on average, then R must recognize L. Bogdanov and Trevisan construct an $\mathsf{AM/poly}$ protocol for recognizing L by leaving this role of the oracle to a prover, which implies that $\mathsf{coNP} \subseteq \mathsf{AM/poly}$ and the collapse of the polynomial hierarchy.

Roughly speaking, their central idea is to divide each R's query $x \in \{0,1\}^n$ into "light" and "heavy" queries according to the probability p_x that the query x is generated by R. Specifically, they determine a threshold $p(n) = \mathsf{poly}(n)$ depending on the permissible error probability for solving (L', U) on average and define a light (resp. heavy) query x as a query satisfying the condition $p_x \leq p(n)2^{-n}$ (resp. $p_x > p(n)2^{-n}$). Then, they make the prover answer (ideally) all light queries correctly, i.e., simulate the following oracle.

$$\mathcal{O}_{\mathcal{L}} = \{x \in \{0,1\}^* : x \in L' \text{ and } x \text{ is a light query}\}$$

Because the number of heavy queries is at most $2^n/p(n)$, the above oracle $\mathcal{O}_{\mathcal{L}}$ solves (L', U) with error probability at most $p(n)$. Therefore, it is enough to make a prover simulate $\mathcal{O}_{\mathcal{L}}$ for constructing an $\mathsf{AM/poly}$ protocol which recognizes L based on R. For the soundness, the verifier must accomplish the following two tasks without deceived by malicious provers: (1) distinguishing between light and heavy queries and (2) identifying the correct answer for each light query. Bogdanov and Trevisan developed such a verifier by introducing sophisticated protocols called the heavy sampling protocol and the hiding protocol.

Herein, we consider the case of auxiliary-input primitives, where each R's query takes the form of (z, x) where z denotes auxiliary-input. For the sake of simplicity, we assume that the task of breaking an auxiliary-input primitive is further reduced to an average-case deterministic problem on uniform distribution with auxiliary-input by applying the techniques in [22, 7] as in the previous work [10] and the length of instances of the average-case problem is the same as the length of auxiliary-input. We also assume that a reduction R makes queries of the form (z, x) where $|z| = |x| = n$ and n is determined by the size of input to R.

There are two possible ways to extend the above idea to the case of auxiliary-input primitives: considering auxiliary-input in queries (a) together or (b) separately. For the first approach (a), we must determine a light query as a query satisfying the condition $p_x \leq \mathsf{poly}(n)2^{-2n}$ for applying the hiding protocol. This is problematic because the number of heavy queries is possibly $2^{2n}/\mathsf{poly}(n)$, and many of them can be concentrated on one auxiliary-input. In other words, the above oracle $\mathcal{O}_{\mathcal{L}}$ does not always solve the average-case problem in the worst-case sense on auxiliary-input. On the other hand, for the second approach (b), their verifier needs information on some statistics as advice for each auxiliary-input. Because there are exponentially many possibilities on auxiliary-input, the total length of such advice is exponentially large, which is unfeasible as an $\mathsf{AM}/\mathsf{poly}$ protocol.

The subsequent work [2] provided the method to remove the above advice in the case of standard cryptographic primitives by applying an additional property of breaking cryptographic primitives (therefore, they constructed an AM protocol for $\neg L$ instead of an $\mathsf{AM}/\mathsf{poly}$ protocol). Unfortunately, even this method cannot be applied directly in the case of auxiliary-input cryptography. To obtain the statistics corresponding to the above advice, their protocol needs to generate query set by executing R. In our case, remember that we consider each auxiliary-input separately, so we need to simulate a conditional distribution on queries for fixed auxiliary-input. However, such distributions are not efficiently samplable in general: for example, consider the query distribution on $(h(y), y)$ where h is a collision-free hashing function. Then, a polynomial-time verifier which simulates a conditional distribution for a fixed auxiliary-input (i.e., hash value) can easily find the collision of h.

5 Proof Sketches

In this section, we present proof ideas of Theorems 4, 8, 11, and 12. Note that Theorem 11 heavily relies on Theorem 8, and Theorem 8 heavily relies on Theorem 4. Therefore, although each proof idea may look pretty simple and intuitive, our construction of OWF for Theorem 11 becomes complicated and quite non-trivial as a whole. For the formal proofs, refer to the full version [29].

5.1 The Case of AIPRG: Proof Idea of Theorem 4

First, we formally introduce a hitting set generator, which takes a crucial role in our proof.

▶ **Definition 13** (Hitting set generator). *Let $\gamma(n)$ be a function. A function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ with $\ell(n) > n$ is a (polynomial-time computable) γ-secure hitting set generator (HSG) if G is polynomially computable and there is no polynomial-time randomized adversary A γ-avoiding G, i.e., satisfying the condition that for all sufficiently large $n \in \mathbb{N}$,*

$$\forall x \in \{0,1\}^n \Pr_A[A(G(x)) = 0] \geq 2/3 \text{ and } \Pr_{y \sim \{0,1\}^{\ell(n)}}\left[\Pr_A[A(y) = 1] \geq 2/3\right] \geq \min(\gamma(n), \tau_n),$$

where τ_n be a trivial limitation defined as $\tau_n := 1 - \frac{|G(\{0,1\}^n)|}{2^{\ell(n)}}$.

Theorem 4 essentially follows from a nonadaptive BB security reduction from distinguishing AIPRG to avoiding HSG. Note that HSG based on AIPRG with a nonadaptive BB security reduction has been implicitly given in the study on MCSP [3, 16]. To see this explicitly, we will provide a much simpler construction of HSG based on AIPRG and a self-contained proof. Although the reader may think that our construction is too fundamental and looks somewhat trivial, to the best of our knowledge, no one has mentioned such a direct relationship between AIPRG and HSG.

First, we assume that there is a nonadaptive BB security reduction from distinguishing AIPRG to avoiding HSG. Avoiding HSG is directly formulated as the following distributional NP problem (with zero-error): for uniformly chosen y, determine whether y is contained in the image of HSG. Therefore, the reduction also yields a nonadaptive BB reduction from distinguishing AIPRG to the distributional NP problem. Thus, any nonadaptive BB reduction from an NP-hard problem to distinguishing AIPRG indeed yields a nonadaptive BB reduction from the same NP-hard problem to the distributional NP problem. By the previous result by Bogdanov and Trevisan [10], such a reduction implies the collapse of the polynomial-hierarchy.

Our construction of HSG from AIPRG is the following: just considering the both of auxiliary-input and input to AIPRG as usual input to HSG. More specifically, let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an AIPRG. Then the construction of HSG G' is given as $G'(z \circ x) = G_z(x)$. Note that, when $z + n(|z|) \geq \ell(n(|z|))$ holds, G' does not satisfy the syntax on stretching input. In the formal proof, therefore, we first stretch the output of G by the standard technique in cryptography. It can be easily verified that the security reduction for this stretching (shown by the famous hybrid argument) is nonadaptive.

Let $\gamma(n)$ be a reciprocal of polynomial. The security reduction from γ-avoiding G' to distinguishing G is also simple: just employing an adversary A for G' as an adversary for G. Obviously, this reduction is nonadaptive. To show the correctness, assume that A γ-avoids G'. For the sake of simplicity, we also assume that A is deterministic and $\gamma(n) < \tau_n$. Whenever the input y is pseudorandom string contained in the image of G', $A(y)$ does not output 1. On the other hand, if y is a truly random string, then $A(y)$ outputs 1 with probability at least $\gamma(n)$. Thus, A can distinguish the uniform distribution from all distributions on the image of G' with an advantage at least $\gamma(n)$. For any auxiliary-input z, $G_z(U_{n(|z|)})$ is distributed on the image of G'. Thus, A also γ-distinguishes G.

5.2 The Case of AIOWF: Proof Idea of Theorem 8

In this section, we omit all arguments about the success probabilities of adversaries to focus on the proof idea. First, we introduce several reductions as elements of a standard OWF. Let $R_{L\to f}$ denote the nonadaptive BB reduction from L to inverting f in the assumption. By the construction of PRG from OWF (e.g., [15]), there exist an auxiliary-input generator G and an adaptive BB reduction $R_{f\to G}$ from inverting f to distinguishing G. By the result in Section 5.1, there exist an NP-language L' and a nonadaptive BB reduction $R_{G\to L'}$ from distinguishing G to a distributional NP problem (L', U) (with zero-error). Since $L' \in \mathsf{NP}$ and L is NP-hard, there exists a Karp reduction $R_{L'\to L}$ from L' to L.

Now we consider the following procedure:

1. select an instance x' of L' at random;
2. translate x' into an instance x of L as $x = R_{L'\to L}(x')$;
3. plug x into $R_{L\to f}$ with a random tape r;

At this stage, $R_{L\to f}$ makes polynomially many queries $(z_1, y_1), \ldots, (z_q, y_q)$.

4. answer the queries by some inverting oracle $\mathcal{O}$;
5. if $R_{L\to f}$ outputs $b \in \{0,1\}$, then output the same decision b.

Note that if the oracle $\mathcal{O}$ correctly inverts f, then the resulting decision b is $L(x)$ with high probability by the property of $R_{L\to f}$, and $L(x)$ is equal to $L'(x')$ by the property of $R_{L'\to L}$.

The crucial observation is that there is no worst-case sense at all in the above procedure because both x' and r are selected at random. Therefore, all queries at the stage 3 are indeed efficiently samplable, and the inverting oracle no longer needs to invert f for every auxiliary-input at the stage 4. This observation leads to the following construction of a standard OWF g.

The function g takes three inputs x', r, and x^f, which intuitively represents a random instance of L', randomness for $R_{L \to f}$, and input for f, respectively. Then $g(x', r, x^f)$ imitates the above procedure as follows: (2') translate x' into an instance x of L as $x = R_{L' \to L}(x')$, (3') plug x into $R_{L \to f}$ with randomness r, then randomly pick one of auxiliary-input z in queries by $R_{L \to f}$ and output $f_z(x^f)$.

We will show that the above g is one-way if $\mathsf{NP} \not\subseteq \mathsf{BPP}$. For contradiction, we assume that there exists an adversary A that inverts g. Remember that g simulates a distribution on queries produced by $R_{L \to f}$ in the above procedure. Thus, intuitively, we can replace the inverting oracle $\mathcal{O}$ with the adversary A at the stage 4 with high probability. This is a little technical part, and we present further details in the full version [29]. Then the above procedure no longer needs any oracle and yields a randomized algorithm solving (L', U) on average. By applying reductions $R_{G \to L'}$, $R_{f \to G}$, and $R_{L \to f}$ in this order, this also yields a randomized polynomial-time algorithm for L. Since L is NP-hard, we conclude that $\mathsf{NP} \subseteq \mathsf{BPP}$.

Remark that $R_{G \to L'}$ is a nonadaptive BB reduction thanks to our simple construction in Section 5.1. Therefore, if we also have a construction of AIPRG G from AIOWF f with a nonadaptive BB reduction from inverting f to distinguishing G, then the above proof leads to a nonadaptive BB reduction from L to (L', U), which implies the collapse of the polynomial hierarchy as in Theorem 4. Thus, finding such a simple construction of AIPRG is one direction for excluding a nonadaptive BB reduction to base AIOWF on NP-hardness, as mentioned in Section 3.

5.3 The Case of AIHSG: Proof Idea of Theorem 11

Our goal is to simulate an avoiding oracle for a nonadaptive BB reduction by another protocol in some restricted complexity class, in our case, BPP. The key idea for this is to classify each query generated by the nonadaptive BB reduction into "light" and "heavy" queries as in [10]. A similar technique was also applied in the previous work for HSG [12, 17]. Thus, we first review the previous case of HSG and then explain the difference to our case of AIHSG.

The Case of Hitting Set Generator (Previous work)

Let $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ denote a generator with $\ell(n) \geq (1 + \Omega(1)) \cdot n$ and $R^?$ denote a nonadaptive BB reduction from an NP-language L to avoiding G. W.l.o.g., we can assume that marginal distributions on each query by R are identical regardless of each query position by applying a random permutation on query positions before asking them to oracle. Thus, for each input $x \in \{0,1\}^n$ to R, one marginal distribution Q_x on R's queries is determined. We choose a threshold (roughly) $\tau = 1/\tilde{\Theta}(2^n)$ and define a light (resp. heavy) query $y \in \{0,1\}^{\ell(n)}$ as a query generated according to Q_x with probability less (resp. greater) than the threshold τ.

We simulate the avoiding oracle for G by using the classification of queries as follows. First, assume that we could (somehow) distinguish the heavy case and the light case for a given query. Then we can also simulate one of avoiding oracles simply as follows: for each query y generated by $R(x)$, (1) determine whether y is heavy or light; (2) answer 0 (resp. 1) if y is heavy (resp. light) query. Let $\mathcal{O}'$ denote the induced oracle by the above simulating procedure. Note that the probability that $\mathcal{O}'(y)$ outputs 0 is exponentially small because the fraction of light queries is $\tilde{\Theta}(2^n)/2^{\ell(n)} \leq 2^{-\Omega(n)}$. Thus, $\mathcal{O}'$ satisfies the condition on the probability of outputting 1. However, $\mathcal{O}'$ is not avoiding oracle for G, because there is possibly a query y such that y is heavy but contained in $\mathrm{Im}G$. In this case, $\mathcal{O}'(y)$ outputs 1 even for $y \in \mathrm{Im}G$ and fails to avoid G.

The key observation to overcome this issue is the following:

($\star$) For each length $\ell(n)$ of query (i.e., the input size is n), the size of $\mathrm{Im}G$ is at most 2^n; thus the probability that R asks some light query contained in $\mathrm{Im}G$ (we refer to it as a "bad" query) is bounded above by $2^n/\tilde{\Theta}(2^n) \leq 1/\mathsf{poly}(n)$.

Therefore, $\mathcal{O}'$ is consistent with some avoiding oracle, and $R^{\mathcal{O}'}(x)$ correctly recognizes x with high probability over the execution of R.

By the above argument, we can reduce avoiding a generator to distinguishing heavy and light queries. For the latter task, Gutfreund and Vadhan [12] presented a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm by approximation of counting in [26], and Hirahara and Watanabe [17] presented an $\mathsf{AM} \cap \mathsf{coAM}$ algorithm by generalizing the protocol in [11].

The Case of Auxiliary-input Hitting Set Generator (Our work)

We move on to our case of AIHSG. Let $G = \{G_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(n(|z|))}\}_{z\in\{0,1\}^*}$ denote an auxiliary-input generator with $\ell(n) \geq (1+\Omega(1)) \cdot n$ and $R^?$ denote a nonadaptive BB reduction from an NP-language L to avoiding G. We can also assume that all marginal query distributions of $R^?(x)$ are identical to Q_x regardless of query position.

To extend the above argument to our case of AIHSG, the problematic part is the key observation ($\star$). Remember that an adversary for AIHSG must avoid G_z for all $z \in \{0,1\}^*$, and auxiliary-input is possibly longer than output. Therefore, we cannot bound the size of the image of the generator in general because the image may span the whole range (for example, consider the following generator $G_z(x) = z \oplus (x \circ 0^{|z|-|x|})$ for $|z| > n(|z|)$).

To overcome this, we need to consider each case of auxiliary-input z separately. Therefore, we change the definitions of "light" and "heavy" queries depending on auxiliary-input. Let $p_x(z)$ denote a probability that Q_x generates a query of auxiliary-input z. If we can bound the probability that R makes light query (z,y) with $y \in \mathrm{Im}G_z$ by $1/(\mathsf{poly}(n)\cdot p_x(z))$ for any z, then R makes such a "bad" query (z,y) with probability at most $\sum_z p_x(z) \cdot 1/(\mathsf{poly}(n) \cdot p_x(z)) = 1/\mathsf{poly}(n)$. Then we can use the same argument in the case of HSG and reduce avoiding G to distinguishing heavy and light cases. This idea naturally leads to the following new definition of "light" and "heavy": separating each query (z,y) by the conditional probability $p_x(y|z)$ that y is asked conditioned on the event that the auxiliary-input in the query is z. In fact, this modification will work well even for AIHSG (for the formal argument, refer to the full version [29]).

However, one issue remains: how can we distinguish heavy and light queries? To this end, we must verify the largeness of the conditional probability of the given query. This part essentially prevents us from applying the previous results. Since we consider a polynomial-time computable generator, the simulation with NP oracle does not yield any nontrivial result, not as the work in [12][5]. Even for the simulation in $\mathsf{AM} \cap \mathsf{coAM}$ in [17], there are

[5] Their work concerned the original aim of HSG, i.e., derandomization (e.g., [25]). For this purpose, they considered (possibly) exponential-time computable HSG G, where avoiding G in $\mathsf{BPP}^{\mathsf{NP}}$ is quite nontrivial. However, in our case where G is polynomial-time computable, avoiding G is in NP trivially.

several technical issues. We cannot trivially verify the size of conditional probability by such protocols due to the restricted use of the upper bound protocol developed in [1]. Moreover, we cannot possibly even sample the conditional distribution efficiently for fixed auxiliary-input, as discussed in Section 4.

Our idea is to adopt universal extrapolation in [22]. Intuitively speaking, the universal extrapolation is a tool to reduce approximating the probability $p_y = \Pr_{U_n}[y = f(U_n)]$ to inverting f for a polynomial-time computable f and a given $y = f(x)$ where $x \in \{0,1\}^n$ is selected at random. In fact, the universal extrapolation holds even for an auxiliary-input function, and a similar technique was also used in [30]. By using the universal extrapolation for each circuit which generates query and auxiliary-input, we have a good approximation of $p_x(y|z)$ for query (z, y) generated by $R^?(x)$. Thus, the universal extrapolation enables us to classify the given (z, y) correctly. Note that the auxiliary-input in the universal extrapolation essentially corresponds to the input x for each circuit sampling query and auxiliary-input.

To show Theorem 11, we need further observations. Since R makes its queries nonadaptively, we can also invoke the universal extrapolation nonadaptively. Moreover, the universal extrapolation algorithm indeed uses an inverting adversary for a certain AIOWF as black-box and nonadaptively (we also see this formally in the full version [29]). As a result, a nonadaptive BB reduction from an NP-hard language L to avoiding AIHSG yields a nonadaptive BB reduction from L to inverting AIOWF. Thus, by Theorem 8, R also yields a one-way function under the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

5.4 Oracle Separation between OWF and AIOWF: Proof Idea of Theorem 12

To show Theorem 12, we employ a random function $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \to \{0,1\}^n\}_{n\in\mathbb{N}}$, where each $\mathcal{F}_n$ is selected uniformly from length-preserving functions of input size n. As shown in [24], any polynomial-time oracle machine cannot invert $\mathcal{F}$ with non-negligible probability (with probability 1 over the choice of $\mathcal{F}$). In other words, if a primitive given access to $\mathcal{F}$ directly outputs the value of $\mathcal{F}$, such a primitive must be one-way. Therefore, all we have to do is to let a random function $\mathcal{F}$ available for auxiliary-input primitives but unavailable for standard primitives.

To this end, we simply add n-bit auxiliary-input to a random function of the input size n. Then we choose one auxiliary-input z_n from 2^n possibilities of $\{0,1\}^n$ as a target auxiliary-input and embed the random function to the position indexed by z_n. Let $\mathcal{F} = \{F_z : \{0,1\}^{|z|} \to \{0,1\}^{|z|}\}_{z\in\{0,1\}^*}$ be such an embedded random function. Note that the similar random embedding technique was also used in the previous work for other oracle separations (e.g., [35]). If an auxiliary-input primitive f given access to $\mathcal{F}$ identifies the auxiliary-input of F with own auxiliary-input, then f must be AIOWF because an adversary for f must invert f_z for all auxiliary-inputs z, including the random function. On the other hand, any polynomial-time computable primitive (without auxiliary-input) cannot find the target auxiliary-input of $\mathcal{F}$ with non-negligible probability because they were selected at random. Thus, any (standard) primitive does not take nontrivial advantage of $\mathcal{F}$.

For the oracle separation, we combine the above embedded random function $\mathcal{F}$ with the PSPACE oracle (w.l.o.g., the oracle TQBF determining satisfiability of quantified Boolean formulae). Let $\mathcal{O}_\mathcal{F}$ denote this oracle. Since the random function in $\mathcal{F}$ is selected independently of TQBF, the additional access to TQBF does not help to invert the random function at all. Thus, AIOWF still exists relative to $\mathcal{O}_\mathcal{F}$.

On the other hand, we consider a function f which is polynomial-time computable with access to $\mathcal{O}_\mathcal{F}$ arbitrarily. Since the target auxiliary-input is selected independently of TQBF, the additional access to TQBF does not help to find the target auxiliary-input at all. Thus, f

cannot still take nontrivial advantage of $\mathcal{F}$ and is regarded as a function given only access to TQBF. We can easily verify that any polynomial-time computable function with access to TQBF is efficiently invertible by TQBF. Since the above argument holds for any f, OWF does not exist relative to $\mathcal{O}_{\mathcal{F}}$. Thus, we have the oracle separation between AIOWF and OWF.

References

1 W. Aiello and J. Håstad. Perfect zero-knowledge languages can be recognized in two rounds. In *28th Annual Symposium on Foundations of Computer Science*, pages 439–448, 1987.

2 A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On Basing One-Way Functions on NP-Hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, STOC '06, pages 701–710, New York, NY, USA, 2006. ACM.

3 E. Allender and S. Hirahara. New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems. *TOCT*, 11(4):27:1–27:27, 2019.

4 A. Andreev, A. Clementi, and J. Rolim. A New General Derandomization Method. *J. ACM*, 45(1):179–213, January 1998.

5 B. Applebaum, B. Barak, and D. Xiao. On Basing Lower-Bounds for Learning on Worst-Case Assumptions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'08, pages 211–220, 2008.

6 T. Baker, J. Gill, and R. Solovay. Relativizations of the $\mathcal{P} =?\mathcal{NP}$ Question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

7 S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *Journal of Computer and System Sciences*, 44(2):193–219, 1992. doi:10.1016/0022-0000(92)90019-F.

8 A. Bogdanov and C. Brzuska. On Basing Size-Verifiable One-Way Functions on NP-Hardness. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 1–6, 2015.

9 A. Bogdanov and C. Lee. Limits of Provable Security for Homomorphic Encryption. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 111–128, 2013.

10 A. Bogdanov and L. Trevisan. On Worst-Case to Average-Case Reductions for NP Problems. *SIAM J. Comput.*, 36(4):1119—-1159, December 2006.

11 J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 124–132, 1991.

12 D. Gutfreund and S. Vadhan. Limitations of Hardness vs. Randomness under Uniform Reductions. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. APPROX 2008, RANDOM 2008*, volume 5171 of *LNCS*, pages 469–482, 2008.

13 I. Haitner, M. Mahmoody, and D. Xiao. A New Sampling Protocol and Applications to Basing Cryptographic Primitives on the Hardness of NP. In *IEEE 25th Annual Conference on Computational Complexity*, pages 76–87, 2010.

14 I. Haitner, O. Reingold, and S. Vadhan. Efficiency Improvements in Constructing Pseudorandom Generators from One-Way Functions. *SIAM Journal on Computing*, 42(3):1405–1430, 2013.

15 J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from Any One-way Function. *SIAM J. Comput.*, 28(4):1364–1396, March 1999.

16 S. Hirahara. Non-Black-Box Worst-Case to Average-Case Reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 247–258, 2018.

17 S. Hirahara and O. Watanabe. On Nonadaptive Reductions to the Set of Random Strings and Its Dense Subsets. In *Complexity and Approximation - In Memory of Ker-I Ko*, pages 67–79, 2020.

18 T. Holenstein. Key Agreement from Weak Bit Agreement. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 664—-673, New York, NY, USA, 2005. ACM.

19 T. Holenstein. Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In *Theory of Cryptography*, pages 443–461, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

20 R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of IEEE Tenth Annual Conference on Structure in Complexity Theory*, pages 134–147, 1995.

21 R. Impagliazzo. Relativized Separations of Worst-Case and Average-Case Complexities for NP. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 104–114, 2011.

22 R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, FOCS'90, pages 812–821, 1990.

23 R. Impagliazzo and M. Luby. One-way Functions Are Essential for Complexity Based Cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

24 R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 44–61, New York, NY, USA, 1989. ACM.

25 R. Impagliazzo and A. Wigderson. Randomness vs Time: Derandomization under a Uniform Assumption. *Journal of Computer and System Sciences*, 63(4):672–688, 2001.

26 M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

27 T. Liu and V. Vaikuntanathan. On Basing Private Information Retrieval on NP-Hardness. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 372–386, 2016.

28 M. Nanashima. Extending Learnability to Auxiliary-Input Cryptographic Primitives and Meta-PAC Learning. In *Proceedings of the 33rd Conference on Learning Theory, COLT'20*, volume 125, pages 2998–3029. PMLR, 09–12 July 2020.

29 M. Nanashima. On Basing Auxiliary-Input Cryptography on NP-hardness via Nonadaptive Black-Box Reductions. *Electron. Colloquium Comput. Complex.*, 27:95, 2020. URL: `https://eccc.weizmann.ac.il/report/2020/095`.

30 R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems*, ISTCS'93, pages 3–17, June 1993.

31 O. Reingold, L. Trevisan, and S. Vadhan. Notions of Reducibility between Cryptographic Primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 1–20, 2004.

32 J. Rompel. One-way Functions Are Necessary and Sufficient for Secure Signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, 1990.

33 R. Santhanam. Pseudorandomness and the Minimum Circuit Size Problem. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, volume 151 of *LIPIcs*, pages 68:1–68:26, 2020.

34 D. Xiao. *New Perspectives on the Complexity of Computational Learning, and Other Problems in Theoretical Computer Science.* PhD thesis, Princeton University, 2009.

35 D. Xiao. On basing ZK $\neq$ BPP on the hardness of PAC learning. In *Proceedings of the 24th Conference on Computational Complexity*, CCC'09, pages 304–315, 2009.