The Curious Case of Non-Interactive Commitments

Mohammad Mahmoody and Rafael Pass*

August 17, 2012

Abstract

It is well-known that one-way permutations (and even one-to-one one-way functions) imply the existence of non-interactive commitments. Furthermore the construction is *black-box* (i.e., the underlying one-way function is used as an oracle to implement the commitment scheme, and an adversary attacking the commitment scheme is used as an oracle in the proof of security).

We rule out the possibility of black-box constructions of non-interactive commitments from general (possibly not one-to-one) one-way functions. As far as we know, this is the first result showing a natural cryptographic task that can be achieved in a black-box way from one-way permutations but not from one-way functions.

We next extend our black-box separation to constructions of non-interactive commitments from a stronger notion of one-way functions, which we refer to as *hitting* one-way functions. Perhaps surprisingly, Barak, Ong, and Vadhan (Siam JoC '07) showed that there does exist a non-black-box construction of non-interactive commitments from hitting one-way functions. As far as we know, this is the first result to establish a "separation" between the power of black-box and non-black-box use of a primitive to implement a natural cryptographic task.

We finally show that unless the complexity class NP has program checkers, the above separations extend also to non-interactive instance-based commitments, and 3-message public-coin honest-verifier zero-knowledge protocols with $O(\log n)$-bit verifier messages. The well-known classical zero-knowledge proof for NP fall into this category.

Keywords: Non-Black-Box Constructions, Black-Box Separations, One-Way Functions, Non-Interactive Commitments, Zero-Knowledge Proofs, Program Checkers, Hitting Set Generators.

---

*Cornell University {mohammad,rafael}@cs.cornell.edu. Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

Contents

1 Introduction

It is well-known that most of the cryptographic constructions are "black-box" in the sense that they ignore the specific implementation of the primitive, and they use both the primitive and the adversary (in the proof of security) as an oracle. Thus black-box constructions capture a main body of our techniques in cryptography for designing protocols and proving their security. In addition, black-box constructions are usually much more efficient than their non-black-box counterparts. In light of this, studying the power and limits of black-box constructions has been a major line of research in cryptography, aiming at finding the "minimal cryptographic primitives" under which a cryptographic task $\mathcal{Q}$ is possible and "separating" $\mathcal{Q}$ from "weaker primitives".

Black-Box Separations. The seminal work of Impagliazzo and Rudich [IR89] put forward a framework for proving the limits of black-box constructions by separating public-key cryptography from private-key cryptography when the construction is black-box. Many other black-box separation results were subsequently established (e.g., [Sim98, GKM$^+$00, GMR01, BPR$^+$08, Vah10, KSY11, MM11][1]). Reingold, Trevisan, and Vadhan [RTV04] further studied various forms of black-box constructions (based on their proof of security). [2] In search of the "minimal" computational primitives required for accomplishing cryptographic tasks, one-way functions emerge as the central player: Almost all natural cryptographic primitives "imply" one-way functions [IL89, OW93, HO11]; moreover, all these constructions are black-box.

One-Way Functions vs. Permutations. One-way *permutations* are a closely related primitive to one-way functions. Even though it is known that there is no black-box construction of one-way permutations from one-way functions [BI87, HHry, Tar89, Rud88, KSS00][3], a surprisingly successful line of research has been to first realize a cryptographic task securely based on the existence of one-way permutations, weaken the assumption to one-to-one one-way functions, and then eventually obtain a construction solely based on the existence of general one-way functions. Examples of this phenomenon include works on pseudorandom generators [BM82, Yao82, Lev87, GKL93, GL89, HILL99] and statistical zero-knowledge arguments as well as statistically-hiding commitments [BCC88, GMR88, BCY91, NOVY98, GK96, DPP98, HHK$^+$05, NOV06, HR07, HNO$^+$07, HRVW09].

Why Trying to Rely on One-Way *Functions*? We emphasize that all known candidates for one-way permutations are based on structured number-theoretic assumptions, and the vulnerability of such structured primitives to possible algebraic (sub-exponential) attacks [LHWL93] makes the feasibility of using one-way functions (rather than permutations) interesting both from theoretical and practical points of view. This puts forward the following basic question:

[1] A closely related line of research aimed at proving lower-bounds on the *efficiency* of black-box constructions (e.g., [KST99, GGKT05, LTW05, HHRS07, BM07, DSLMM11]).

[2] Our notion of black-box construction here corresponds to the notion of *fully* black-box construction as defined in [RTV04] where we also include the security parameter; see Definition 3.9.

[3] The results implicit in [BI87, HHry, Tar89] show that there is no fully black-box construction of one-way permutations from one-way functions (see [MM11] for an exposition of this argument). This results extends even to separating one-way functions from injective one-way functions. Rudich [Rud88] observes that this separation is implicit in those previous works and improves them to separate one-way permutations from *random oracles*, even if the construction is allowed to have small completeness error, at the cost of assuming a combinatorial conjecture that was later resolved in [KSS00]. See [Rud88] for more discussions.

Main Question 1: *Is there any natural cryptographic task that can be accomplished based on the black-box assumption of one-way permutations but not one-way functions?*

We consider one-way functions and permutations both as *computational assumptions* and not as natural cryptographic *tasks*, and so the separation of one-way permutations from one-way functions does not answer our question above.

The Power of Black-Box vs. Non-Black-Box Constructions. Another similar successful line of research in the foundations of cryptography has been to start by providing non-black-box constructions of a primitive and later turning them into black-box ones. Examples include e.g., secure computations from various primitive [HIK+11, CDSMW08, CDSMW09, Wee10, Goy11], oblivious transfer from semi-honest oblivious transfer [Hai08], constant-round zero-knowledge arguments and trapdoor commitments from one-way functions [PW09], etc. Despite this, as far as we know the following intriguing question has remained open:

Main Question 2: *Is there a natural cryptographic task $\mathcal{Q}$ that can be based on a cryptographic primitive $\mathcal{P}$ in a non-black-box way, while no black-box construction of $\mathcal{Q}$ based on $\mathcal{P}$ exists?*

In this work we answer both the above questions affirmatively: **(1)** There is a cryptographic task that can be based on one-way permutations but not one-way functions in a black-box way. **(2)** The same primitive can be used to separates the power of black-box and non-black-box constructions. Interestingly, the primitive is a very natural cryptographic building block: *non-interactive commitments*.

Commitment Schemes. Bit-commitments are one of the most fundamental cryptographic building blocks. Their application ranges from zero-knowledge proofs [GMR89, GMW91] to secure computations [GMW87]. Roughly speaking, a commitments scheme is a two-stage protocol between two parties: the sender and the receiver. In the first, so-called, *commitment phase*, the sender commits to a secret bit b; and then later in the *decommitment phase*, the sender reveals the bit b together with some additional information which allows the receiver to verify the correctness of the decommitment. Commitment schemes are required to satisfy two properties: *hiding* and *binding*. Roughly speaking, the hiding property stipulates that after the commitment phase the bit b should remain hidden to the receiver, whereas the binding property asserts that in the decommitment phase the sender is not able to decommit successfully to both $b = 0$ and $b = 1$.

The results of Naor [Nao91] and Håstad, Impagliazzo, Luby and Levin [HILL99] establish that the existence of one-way functions implies the existence of commitment schemes where the commitment phase consists of two messages. Furthermore their construction is black-box and the commitment scheme uses the underlying one-way function as an oracle. On the other hand, Impagliazzo and Luby [IL89] establish that the existence of commitment schemes implies the existence of one-way functions (in a black-box way).

We focus on the black-box complexity of *non-interactive commitments*—namely, commitment schemes where both the commitment phase and the decommitment phase consist of a single message. The results of [Blu81, GL89] establish the existence of non-interactive commitments based on one-way permutations (or even one-to-one one-way functions) using a black-box construction. These results extend even to the case of *families* of one-way permutations where given an index

p one can efficiently verify that f_p is indeed a permutation.[4] The work of Naor showed how to obtain interactive commitments based on any one-way function in a black-box way, where the commitment phase consists only of a random message from the receiver followed by a message from the sender (thus the first message can be eliminated in the common reference string model). It remained an open question whether there a black-box construction of non-interactive commitments from one-way functions, or that to obtain this primitive one-way permutations are more powerful than one-way functions.

1.1 Our Results

Our first result shows that one-way functions cannot be used as a black-box to obtain noninteractive commitments, answering our first main question affirmatively.

Theorem 1.1. *There is no black-box construction of non-interactive commitments from one-way functions.*

The separation extends to stronger primitives than one-way functions (e.g., *families* of collision-resistant hash function). As far as we know, this is the first result showing a natural cryptographic task that can be constructed in a black-box way from one-way permutations but not from one-way functions resolving our first question affirmatively.

Non-Black-Box Non-Interactive Commitments from One-Way Functions. The elegant work by Barak, Ong and Vadhan [BOV03] provides a *non-black-box* construction of non-interactive commitments assuming the existence of one-way functions and certain hitting-set generators (see the discussion in Section 2.2) against co-non-deterministic circuits (see Definition 2.3 for a formalization) which can be constructed under worst-case complexity assumptions. Roughly speaking, the hitting-set generator $G\colon \{0,1\}^\ell \mapsto \{0,1\}^{\mathrm{poly}(n)}$ is used to derandomize Naor's 2-message commitment scheme by executing the commitment in parallel over *all* of $G(\{0,1\}^\ell)$ as the "first messages" of the protocol (thus we require $2^\ell = \mathrm{poly}(n)$). Naor's commitment has the nice property that for every one-way function used, most of $\{0,1\}^n$ can be fixed as the first message to make the scheme perfectly binding. The hitting property of the generator G guarantees that at least one of the fixed first messages $G(\{0,1\}^\ell)$ makes the (non-interactive) scheme binding.

Conditional Separation of the Power of Black-Box and Non-Black-Box Constructions. The result of [BOV03] together with our Theorem 1.1 show that under any complexity assumption that guarantees the existence of hitting-set generators against co-nondeterministic circuits, non-black-box constructions are more powerful than black-box constructions (since a non-black-box construction of non-interactive commitments from one-way functions would exist, while no such black-box construction exists). This answers our second main question also affirmatively based on believable complexity conjectures. As we will see shortly, we are able to make this "separation" (between the power of the two models) *unconditional* by defining a new primitive that can be used as a hitting-set generator.

[4] For example, one can sample a random prime number p and define the permutation f_p to be the discrete logarithm function in the group $\mathbb{Z}_p^*$. Primality of p can be tested efficiently [Mil76, Rab80, AKS02] and this guarantees f_p is indeed a permutation.

Non-Interactive Commitments from Hitting One-Way Functions. Inspired by the work of [BOV03], we introduce the notion of *hitting one-way functions*; roughly speaking, a (one-way) function f is said to be *hitting*, if for every co-non-deterministic circuit of size n which accepts at least half of its inputs, there exists at least one input $x \in [1, \dots, n^2] \subseteq \{0,1\}^n$ which $f(x)$ is accepted by the circuit. It is easy to see that a random oracle is a hitting one-way function with overwhelming probability (see Lemma 2.7). Furthermore, we show that there exists a non-black-box construction of non-interactive commitments from hitting one-way functions as follows. Following [BOV03], we derandomize Naor's commitment scheme by evaluating the hitting one-way function f on the inputs $1, \dots, n^2$ (appropriately planted in $\{0,1\}^n$), where n is a polynomial that is determined by the size of the verification circuit in Naor's commitment. Since Naor's commitment also relies on the use of the one-way function f, the choice of n depends on the circuit size of f; thus the construction is non-black-box. Thus we obtain the following theorem.[5]

Theorem 1.2. *There is a* non-*black-box construction of non-interactive commitments from hitting one-way functions.*

In contrast, we prove the following theorem in the black-box regime resolving our second main question affirmatively (unconditionally).

Theorem 1.3. *There is no black-box construction of non-interactive commitments from hitting one-way functions.*

As far as we know, this constitutes the first separation between the power of black-box and non-black-box use of a primitive in the implementation of a natural cryptographic task. This is different from the results of Barak [Bar01] and Goldreich-Krawczyk [GK92] which provide a separation between the power of black-box and non-black-box *proofs of security*, and in this work all proofs of security are black-box. Thus we also resolve our second main question affirmatively.

Extension to 3-Message Honest-Verifier Zero-Knowledge (3-HVZK). A major application of commitment schemes is to construct zero-knowledge proofs for NP. Non-interactive commitments for NP can be used to derive 3-HVZK for NP in a black-box way, and so a separation between 3-HVZK and one-way functions would be a stronger statement. Thus we also study whether 3-HVZK for NP can be constructed based on one-way functions in a black-box way. We extend our separation (from one-way functions) also for some forms of 3-HVZK in a *conditional* way; our separations hold assuming that the complexity class NP does not have "program checkers" [BK95]. In particular, we show that black-box constructions of public-coin 3-HVZK protocols with short verifier messages based on one-way functions would imply program checkers for SAT. Such constructions still include the classical 3-message zero-knowledge protocols, e.g., GMW Graph 3-Coloring protocol [GMW87], Blum's Hamiltonian Cycle protocol [Blu87] etc.

Theorem 1.4. *Any black-box construction of a 3-message honest-verifier zero-knowledge proofs (or arguments) for* NP *from one-way functions with the following features implies that* NP *is checkable.*

1. $1 - \mathrm{negl}(n)$ *completeness.*

2. $1/\mathrm{poly}(n)$ *soundness, i.e., soundness error as large as* $1 - 1/\mathrm{poly}(n)$*.*

[5] Our positive and negative results are robust to choosing n^2 as the size of the hitting set generator and they can be adopted to work with any function $\omega(n)$. We choose to use n^2 for sake of simplicity.

3. The verifier has no secret randomness and in the second message she sends $O(\log n)$ bits.

Whether NP has program checkers or not has been open for more than two decades [FRS88, BK95, BFL90], thus our results indicate that providing black-box constructions of 3-HVZK with properties mentioned above for NP at least requires resolving a long-standing open question in computational complexity. Note that computational assumptions such as $\mathsf{P} \neq \mathsf{NP}$ are *necessary* to obtain Theorem 1.4. It is easy to see that if $\mathsf{P} = \mathsf{NP}$, then forms of non-interactive commitments known as "instance-based commitments"[6], which are sufficient to obtain 3-HVZK exist unconditionally.

Organization. In Section 2 we sketch the ideas and tools used in the proofs of our results whose full proofs will appear in the subsequent sections. See Section 3.1 for the description of our notation and terminology and Section 8 for some concluding remarks and remaining open questions

2 Outline of Proofs and Techniques

In this section we outline the framework and the main ideas and techniques used in the proofs of Theorems 1.1, 1.3, and 1.4.

2.1 Separation from One-Way Functions

Here we outline the proof of Theorem 1.1. For simplicity of the presentation here we settle this theorem only for the natural setting that the verification of the decommitment is deterministic and the scheme has perfect completeness.

We start by formalizing the notion of black-box constructions by following the paradigm of [RTV04] with the following changes: **(1)** we include the security parameter, and **(2)** we restrict ourself to constructions that use "almost everywhere" security (i.e., that for large enough security parameter there is no efficient adversary breaking the primitive) as opposed to "infinitely often" security (i.e., that for an infinite sequence of security parameters the scheme is secure). Roughly speaking, black-box constructions consist of two reductions: implementation and proof of security. The implementation Q of the new primitive $\mathcal{Q}$ uses any implementation P of the base primitive $\mathcal{P}$ only as an oracle. The security reduction S bases the security of Q^P on the security of P as follows: for every (unbounded) adversary A who breaks the security of Q^P, $S^{A,P}$ breaks the security of P. Note that a commitment scheme has two players, and so breaking the security amounts to breaking *either* of hiding or binding properties. The following definition formalizes the above definition for the case of commitment schemes.

Definition 2.1. A black-box construction of non-interactive commitments from one-way functions is a pair of efficient oracle algorithms $\textsc{Com}^{(\cdot)} = (S^{(\cdot)}, R^{(\cdot)})$ such that: The parties receive the common input 1^n as the security parameter and access an oracle $f = \{f_m \colon \{0,1\}^m \mapsto \{0,1\}^m\}$. The security of the scheme is guaranteed through reductions to the one-wayness of f as follows.

- **Proving the Hiding:** There is an efficient security reduction H that proves that $\textsc{Com}^f$ is hiding. Namely, for every oracle f and every malicious receiver $\widehat{R}$ (who could arbitrarily

[6] In an instance-based commitment scheme w.r.t. a language L, the parties receive some common input x. The hiding needs to hold only when $x \in L$, and the binding needs to hold only when $x \notin L$.

depend on f) that distinguishes commitments to $0, 1$ with non-negligible advantage $\varepsilon > 1/\operatorname{poly}(n)$, the oracle algorithm $H^{f,\widehat{R}}$ breaks the one-wayness of f with probability at least $\operatorname{poly}(\varepsilon/n)$ over a polynomially related $m = n^{\Theta(1)}$ input length:

$$\Pr_{y \xleftarrow{\$} f(\mathbf{U}_m)}[H^{f,\widehat{S}}(y) \in f^{-1}(y)] \geq \left(\frac{\varepsilon}{m}\right)^{O(1)}.$$

- **Proving the Binding:** It is defined similarly to the definition of Hiding using another reduction B that inverts f with non-negligible probability given oracle address to f and any adversary who breaks the binding of $\textsc{Com}^f$.

In order to prove Theorem 1.1, we employ the methodology formally described in the following lemma (which is also used in the previous works of [BM07, DSLMM11]).

Lemma 2.2. *There is no black-box construction of non-interactive commitments form OWFs, if there is any randomized oracle* $\mathbf{O}$ *with the following properties:*

1. *The hiding or binding of* $\textsc{Com}^{\mathbf{O}}$ *is violated by a* $\operatorname{poly}(n)$*-query attack.*
2. $\mathbf{O}$ *is strongly one-way in the sense that no* $\operatorname{poly}(n)$*-query computationally-unbounded adversary can invert* $\mathbf{O}$ *over* $\mathbf{O}(\mathbf{U}_n)$ *with probability* $\geq 1/\operatorname{poly}(n)$.

Roughly speaking Lemma 2.2 holds because if the black-box construction from OWFs existed, one could use $\mathbf{O}$ to implement the scheme and then use the $\operatorname{poly}(n)$ attacker against this scheme combined with the black-box proof of security to get an algorithm that invert $\mathbf{O}$ with only $\operatorname{poly}(n)$ oracle queries to $\mathbf{O}$ (which is a contradiction).

In the following we describe how to find a distribution for the randomized oracle $\mathbf{O}$ so that we can apply Lemma 2.2 to prove Theorem 1.1.

The Oracle O Cannot be a Random Oracle. We first note that we can not simply use $\mathbf{O}$ to be a random oracle which is indeed a common method to derive separations from one-way functions. This is expected, since otherwise we could also get a separation from one-way permutations (since random oracle and random permutation oracle are indistinguishable over large enough input lengths), and this would be a contradiction. In particular, relative to a random oracle, with high probability, there exists a *one-to-one* one-way function[7] which is indeed sufficient for constructing non-interactive commitments in a black-box way [Blu81].

Partially-Fixed Random Oracles. We overcome the above obstacle by choosing the distribution of our oracle $\mathbf{O}$ to be *fixed* over a polynomial-size subset $\mathcal{F}$ of its domain (which in fact depends on the construction $\textsc{Com}$ itself), and at any other point out of $\mathcal{F}$ we choose the answers randomly. In general, we call oracles *partially-fixed random*. Partially-fixed random oracles allow us to bypass the obstacle explained above against random oracles, because the way we fix the part $\mathcal{F}$ most likely makes the oracle $\mathbf{O}$ have collisions; thus, $\mathbf{O}$ will not be one-to-one. In fact, the collisions of $\mathbf{O}$ are

[7]For example, using standard tricks one can make the output of the random oracle long enough, say n^3 bits, while the input is only n bits. Such function is one-to-one with overwhelming probability.

planted in an adversarial way against the construction $\textsc{Com}$ and that is why the distribution of $\mathbf{O}$ depends on $\textsc{Com}$.[8]

It is easy to see that a partially-fixed random oracle is still hard to invert using $\mathrm{poly}(n)$-query attacks. We show how to define the the distribution of $\mathbf{O}$ such that, either of the binding or hiding properties of $\textsc{Com}^{\mathbf{O}}$ will be violated through a $\mathrm{poly}(n)$-query attack. As we discussed above, this is sufficient for deriving a black-box separation. We prove the existence of such partially-fixed random oracle $\mathbf{O}$ by proving that there are in fact *two* partially-fixed random oracles $\mathbf{O}_R$ and $\mathbf{O}_S$ such that *either* of the following holds:

1. The hiding of $\textsc{Com}^{\mathbf{O}_R}$ is broken by a $\mathrm{poly}(n)$-query malicious receiver $\widehat{R}$.

2. The binding of $\textsc{Com}^{\mathbf{O}_S}$ is broken by a $\mathrm{poly}(n)$-query malicious sender $\widehat{S}$.

Therefore, there always exists an oracle $\mathbf{O} \in \{\mathbf{O}_S, \mathbf{O}_R\}$ relative to which either of the hiding or binding properties of $\textsc{Com}$ is broken by some $\textsc{Adv} \in \{\widehat{R}, \widehat{S}\}$.

2.1.1 Cheating Strategies $\widehat{S}, \widehat{R}$.

The cheating sender $\widehat{S}$ and the distribution of $\mathbf{O}_S$ are defined assuming that $\widehat{R}$ fails in its attack, but that is still sufficient for us. The oracle $\mathbf{O}_R$ is simply the random oracle, but the oracle $\mathbf{O}_S$ will always be fixed over a polynomial-size domain (thus the final oracle $\mathbf{O} \in \{\mathbf{O}_R, \mathbf{O}_S\}$ will always be a partially-fixed random oracle. The algorithm of the malicious $\widehat{R}$ is in fact very simple: try to learn any query q that has a non-negligible chance of being asked by the sender during the generation of the commitment C, and after learning these queries make a guess about the committed bit b by outputting the more likely value of b conditioned on the knowledge learned about the random oracle $\mathbf{O}_R$. In the following we formally describe this algorithm and will show that if this algorithm fails in guessing the bit b correctly with probability $1/2 + 1/\mathrm{poly}(n)$, then we can come up with a partially-fixed random oracle $\mathbf{O}_S$ such that the binding of $\textsc{Com}^{\mathbf{O}_S}$ could be violated.

Technical Tool: Learning Heavy Queries. Suppose $\textsc{Com} = (S, R)$ is a non-interactive commitment scheme in a model where some randomized oracle $\mathbf{O}$ (e.g., the random oracle) is accessed by the sender S and the receiver R and suppose S generates a commitment C to a random bit $b \xleftarrow{\$} \{0,1\}$. Let S be the view of the sender consisting its randomness as well as its oracle query-answers and R be the view of the receiver after the verification of C which consists of C itself, the revealed bit b and some "decommitment" string D justifying the claim of S that he had committed to b. We can look at all of $\mathsf{S}, \mathsf{R}, C, b$, and D as random variables depending on the randomness of the parties and the randomness of $\mathbf{O}$. That is the case also for the set of queries $\mathcal{Q}(\mathsf{S}), \mathcal{Q}(\mathsf{R})$ asked by the sender and the receiver represented in their views.

Consider the following simple learning algorithm that upon receiving C, which is the commitment to a random $b \xleftarrow{\$} \{0,1\}$, keeps updating a "learned" set of oracle query-answer pairs $\mathcal{L}$ as follows: As long as there is an oracle query $q \notin \mathcal{L}$ which has ε probability to be asked by the sender during the generation of C or by the receiver during the verification of C:

$$\Pr[q \in \mathcal{Q}(\mathsf{S}) \cup \mathcal{Q}(\mathsf{R}) \mid C, \mathcal{L}] \geq \varepsilon,$$

[8] As far as we know, this way of choosing the oracle's distribution based on the scheme itself was fist employed in the work of Gertner et al. [GMR01].

then go ahead and ask q from the oracle. After asking q from $\mathbf{O}$, the pair $(q, \mathbf{O}(q))$ will be added to $\mathcal{L}$ and the knowledge of $\mathbf{O}(q)$ will be incorporated in deciding which other queries might be likely as described above. A result due to [BM07] shows that such learning algorithm would (on average) ask at most $\mathrm{poly}(n/\varepsilon) = \mathrm{poly}(n)$ queries and reach a point that there is no "ε-heavy" query left for the distribution of the views of the sender and the (honest) receiver conditioned on the learned information $(C, \mathcal{L})$. As we will see, this learning algorithm will essentially form the cheating receiver's algorithm $\widehat{R}$.

Defining the Cheating Strategies. Suppose we execute the learning algorithm above when the randomized oracle $\mathbf{O}$ in the scheme is simply a random oracle. We focus on the moment that the learning algorithm stops (i.e., for any query $q \notin \mathcal{L}$ it holds that $\Pr[q \in \mathcal{Q}(\mathsf{S}) \cup \mathcal{Q}(\mathsf{R}) \mid C, \mathcal{L}] < \varepsilon$), and divide possible the cases into two. In each case we show how to derive a cheating party and a corresponding randomized oracle.

- **Case 1.** In the first case, with non-negligible probability $1/\mathrm{poly}(n)$ over the executing of the learning algorithm, at the end there is a value $b \in \{0,1\}$ such that $\Pr[b \text{ is the committed bit } \mid (C, \mathcal{L})] > 1/2 + 1/\mathrm{poly}(n)$. In this case we can simply take $\mathbf{O}_R$ to be the random oracle, and relative to $\mathbf{O}_R$ the cheating strategy $\widehat{R}$ could just follow the learning algorithm above and output the more likely value of b conditioned on its view $(C, \mathcal{L})$ at the end. It is easy to see that this malicious receiver $\widehat{R}$ can guess the bit b with probability at least $1/2 + 1/\mathrm{poly}(n)$.

- **Case 2.** In the second case, at the end of the learning phase when there is no ε-heavy query left to be learned, with overwhelming probability: both of the values of $b \in \{0,1\}$ are almost equally likely to be the committed bit conditioned on knowing $(C, \mathcal{L})$. We will show that at this point there is always a way to fix a set of oracle query-answer pairs $\mathcal{F}$ for some partially-fixed random oracle $\mathbf{O}_S$ such that $\widehat{S}$ can successfully open the commitment C (which is the result of a single execution of the learning algorithm and is fixed forever) into both of $\{0,1\}$.

  Since we are in the case that conditioned on $(C, \mathcal{L})$ both values of $b \in \{0,1\}$ have non-zero (in fact $\approx 1/2$) chance to be the committed bit, we can always sample two views $\mathsf{V}_0 = (\mathsf{S}_0, \mathsf{R}_0), \mathsf{V}_1 = (\mathsf{S}_1, \mathsf{R}_1)$ of full executions of the system for the sender and the receiver where they are both consistent with $(C, \mathcal{L})$ and V_b describes a case where C is a commitment to the bit b. Note that due to the (assumed) perfect completeness of the scheme, in both of the views $\mathsf{V}_0, \mathsf{V}_1$ the verification leads to an accept. We claim that if S_0 and S_1 are *consistent* over the query-answer pairs that they posses (i.e., use the same answer for the queries that they *both* have asked: $\mathcal{Q}(\mathsf{S}_0) \cap \mathcal{Q}(\mathsf{S}_1)$) then we are done, because we can take $\mathcal{F}$ to be the answers to $\mathcal{Q}(\mathsf{S}_0) \cup \mathcal{Q}(\mathsf{S}_1)$ plus the query-answer pairs of $\mathcal{L}$ and fix $\mathcal{F}$ as part of the partially-fixed random oracle $\mathbf{O}_S$. This way, whenever the sender wants to decommit to the bit $b \in \{0,1\}$ it can use the fixed view $\mathsf{S}_b \in \mathsf{V}_b$ for the needed decommitment, and he knows that such decommitment will always lead to the verification described by $\mathsf{R}_0 \in \mathsf{V}_b$ (since the verification is deterministic) which is an accept.

  It only remains to show how to find a pair of *consistent* views $\mathsf{V}_0, \mathsf{V}_1$. Here we use the fact that conditioned on $(C, \mathcal{L})$, both of $\{0,1\}$ have chance $> 1/3$ to be the committed bit. Using a probabilistic analysis and also relying on the fact that there is no ε-heavy query left conditioned on $(C, \mathcal{L})$ (when the committed bit is considered random), and assuming that the total number of oracle queries of (S, R) is at most m, one can show that with probability

$\geq 1 - 3m\varepsilon$ a pair of *random* samples $\mathsf{V}_0, \mathsf{V}_1$, where V_b is sampled conditioned on $(C, \mathcal{L}, b)$, would have no query in common out of $\mathcal{L}$ (i.e., $\mathcal{Q}(\mathsf{V}_0) \cap \mathcal{Q}(\mathsf{V}_1) \subseteq \mathcal{L}$). The reason is that for any query q which has probability at most ε to be in the queries of V, by conditioning on $b = 0$ or $b = 1$, this probability can increase at most to 3ε. Therefore, if we sample and fix V_0, any of the m queries of the sampled V_0 would be sampled in V_1 only with probability at most 3ε. Thus, by a union bound, with probability at least $1 - 3m\varepsilon$, none of the quereis of V_0 will be sampled in V_1. Since both of $\mathsf{V}_0, \mathsf{V}_1$ are sampled conditioned on (and consistent with) $\mathcal{L}$, we conclude that such samples are in fact consistent.

The Role of Non-Interactivity. Our argument above only applies to the non-interactive setting because of the way we constructed $(\widehat{S}, \mathbf{O}_S)$ in case $\widehat{R}$ does not succeed. In particular, in the interactive setting C would be the transcript of an interactive protocol which could change every time that the protocol is executed, even if the sender commits to the same message using the same randomness, simply because the receiver's randomness might change every time. That should not be a surprise since Naor's commitment scheme [Nao91] is a black-box construction based on one-way functions and has only two messages during the commitment phase (which perfectly complements our negative result of Theorem 1.1).

2.2 Separation from Hitting One-Way Functions

Here we outline the proof of Theorem 1.3. Before doing so we need to develop the notion of a hitting one-way function.

2.2.1 Hitting One-Way Functions

Hitting Set Generators. A (basic) *hitting set generator* G is an efficient deterministic procedure to generate sets that intersect any "dense" set recognized by an efficient circuit. More formally, given $n \geq m$, G runs in time $\text{poly}(n)$ and generates a set of m-bit strings $\mathcal{H}$ such that for any circuit T accepting at least half of $\{0,1\}^m$, it holds that $T(h) = 1$ for at least one $h \in \mathcal{H}$ (see [GW99] and references therein for more background on the subject). A hitting set generator G can be directly used to derandomize the complexity class RP and perhaps surprisingly even to derandomize the class BPP [ACP98, ACPT99]. Here we are interested in the notion of hitting set generators against co-nondeterministic circuits defined as follows.

A more general notion of hitting set generators was also developed for the purpose of derandomizing *nondeterministic* algorithms by Miltersen and Vinodchandran [MV05] based on the previous works of [AK01, KvM02] in the broader context of using NW-type pseudorandom generators for derandomization purposes. Such hitting set generators are proved to exist under the complexity assumption that the class $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ has $2^{\Omega(n)}$ nondeterministic circuit complexity. Barak, Ong, and Vadhan [BOV03] showed the first application of NW-type generators (against co-nondeterministic circuits) to cryptography by derandomizing Naor's bit commitment scheme.

Definition 2.3 (Co-Nondeterministic Circuits)**.** A *nondeterministic* Boolean circuit T takes two inputs and accepts the set $\mathcal{S}_T$ defined as follows $\mathcal{S}_T = \{x \mid \exists\, w, T(x, w) = 1\}$. A *co-nondeterministic* Boolean circuit T also takes two inputs and accepts the set $\mathcal{S}_T = \{x \mid \forall\, w, T(x, w) = 0\}$. By abusing the notation we call the first input simply the "input" and call the second input the "witness". Thus, the input length refers to the length of x. If the input length is n, we call $d_T(n) = \frac{|\mathcal{S}_T \cap \{0,1\}^n|}{2^n}$ the input density of T.

Now we introduce a new primitive that combines a one-way function and a hitting set generator against co-nondeterministic circuits.

Definition 2.4 (Hitting One-Way Functions)**.** We say a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ *hits* a co-nondeterministic circuit T of size n and input length m if it holds that $\{f(1)|_m, \dots f(n^2)|_m\} \cap \mathcal{S}_T \neq \varnothing$ where $1, 2, \dots, n^2$ are analogs of the first n^2 elements of $\{0,1\}^n$ and $y|_m$ refers to the first m bits of y. We say that a sequence of functions $\{f_n\colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is a *hitting function*, if f_n hits every circuit T of size n and input density $d_T \geq 1/2$ for large enough n. A length preserving function family $f = \{f_n\colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is simply called a *hitting one-way* function, if it is both hitting and one-way simultaneously.

As we will see later, a random oracle is a hitting one-way function with overwhelming probability, and thus being hitting one-way could be thought of as a natural abstracted property of a random oracle (similar to e.g., collision resistance). Moveover, it is easy to see that hitting one-wayness can be formalized using a standard cryptographic security game, and as such, the assumption that a function f is a hitting one-way function is a falsiafiable assumption, in the terminology of Naor [Nao03].[9]

Construction 2.5 (Security Game of Hitting One-Way Functions)**.** The security of hitting one-way functions can be defined through a two-party game whose winner can be efficiently and publicly verified.[10] For the security parameter n, the challenger sends $f(\mathbf{U}_n) = y$ to the adversary Adv who in return does as follows:

1. Either Adv sends back some x such that $f(x) = y$, or
2. Adv sends back a "proof" that f_n is not hitting, which includes a circuit T of size n and input length m and a sequence $w_1, \dots, w_{n^2}$ such that $T(f(i)|_m, w_i) = 1$ for all $i \in [n^2] \subset \{0,1\}^n$.

Clearly, if an efficient adversary wins in this game for an infinite sequence of security parameters, then either f is not hitting or it is not one-way. On the other hand, if f is not hitting one-way, it is easy to see that there is always a *non-uniform* adversary Adv of size $\mathrm{poly}(n)$ that wins the game above for an infinite sequence $n \in \{n_1 < n_2 < \dots\}$ with a non-negligible probability, because for every input length n over which f is not hitting Adv can know the sequence $w_1, \dots, w_{n^2}$ through its non-uniform advice. This motivates the definition of a weaker primitive: *uniformly-secure* hitting one-way functions as follows.

Definition 2.6 (Uniform Hitting One-Way Functions)**.** We call an efficiently computable sequence of functions $\{f_n\colon \{0,1\}^n \mapsto \{0,1\}^n\}$ *uniform hitting one-way*, if any efficient uniform adversary Adv participating in the security game of Construction 2.5 can win only with negligible probability.

Note that any uniform hitting one-way function f is also a (uniformly-secure) one-way function, but it might be that it is only *hard* to refute the hitting property of f even though it is not actually a hitting function. Interestingly, Theorem 1.2 can be proved only based on the existence of uniform hitting one-way functions (resulting in a uniformly secure commitment scheme). We believe it is a reasonable conjecture to assume that (a generalized version of say) AES is a uniform hitting

[9] A subtle point here is that the hitting property is defined w.r.t. *co*-nondeterministic (as opposed to nondeterministic) circuits. Thus when f is not hitting, there always exits a polynomal-size witness for that: a circuit T of size n and input length m and a sequence $w_1, \dots, w_{n^2}$ such that $T(f(i)|_m, w_i) = 1$ for all $i \in [n^2] \subset \{0,1\}^n$.

[10] Note that in a similar game that captures the hitting property of a function against *nondeterministic* (as opposed to *co*-nondeterministic) circuits, one can not provide short witness that the function is *not* hitting.

one-way function, since even though it might not be hitting it seems extremely hard to refute it *efficiently*. As we show later on, a random oracle is clearly is a hitting one-way function, and so an attack against the hitting property of AES would also constitute a concise evidence against AES as a random oracle.

Finally, we note that it is easy to prove the existence of hitting one-way functions assuming that **(1)** one-way functions exist and that **(2)** there exist efficient hitting set generators against co-nondeterministic circuits. More formally, let $G(1^n, 1^m)$ be the hitting set generator which generates $q = \text{poly}(n, m) \leq \text{poly}(n)$ output strings of length m hitting any co-nondeterministic circuit of size n and input length m. Suppose also that f is a one-way function. First, we can get $G'(1^n)$ to be an efficient algorithm that enumerates over all $m \leq n$ as possible first input lengths and obtains a larger output set of size $q' = m \cdot q \leq \text{poly}(n)$ that hits any co-nondeterministic circuit of size n. Then we can "substitute" the first q' outputs of f (i.e., $f(1), \ldots, f(q')$) over the domain $\{0,1\}^n$ with the elements of $G'(1^n)$ (when padded to n bits).[11]

2.2.2 Outline of the Proof of Theorem 1.2

Here we assume the reader is familiar with Naor's commitment scheme. Following [BOV03] our non-black-box construction of non-interactive commitments from one-way functions is essentially a derandomization of Naor's protocol, with the difference that here we use a hitting one-way function rather than worst-case complexity assumptions. Recall that in Naor's protocol the first message sent by the verifier is a random string $r \xleftarrow{\$} \{0,1\}^{3m}$ where $g \colon \{0,1\}^m \mapsto \{0,1\}^{3m}$ is a pseudorandom generators constructed based on a one-way function $f \colon \{0,1\}^\ell \mapsto \{0,1\}^\ell$. To guarantee (perfect) binding, it is sufficient to have $f(\{0,1\}^m) \cap \{y \colon y \in f(\{0,1\}^m) + r\} = \varnothing$. The set of all such "good" r (that satisfy $f(\{0,1\}^m) \cap \{y \colon y \in f(\{0,1\}^m) + r\} = \varnothing$) is accepted by a co-nondeterministic circuit T defined as: $T(r, w) = 1$ iff $w = (x_1, x_2)$ and $g(x_1) = g(x_2) + r$.

The derandomized protocol uses the hitting one-way function f over input length $n = |T|$ to obtain a small set $\{r_1, \ldots, r_{n^2}\}$ such that at least one of r_i is good (i.e., accepted by T). This is possible assuming that f is hitting because T accepts at least $1 - 2^{-m} > 1/2$ fraction of its (first) inputs. Therefore, one can eliminate the first round of the protocol and run the original protocol over *all* of $\{r_1, \ldots, r_{n^2}\}$ in parallel.

The hiding property of the new non-interactive protocol follows from the hiding of the original protocol and a standard hybrid argument. The hiding of the original protocol relies on the pseudorandomness of g, which in turn relies on the fact that f is one-way.

The binding property of the new protocol directly relies on the hitting property of f. Namely, any adversary that breaks the binding of the new protocol can be efficiently (and uniformly) turned into an algorithm that refutes the hitting property of f. Therefore, the new protocol is uniformly binding, assuming that f is uniform hitting, and it is *perfectly* binding if the one-way function f is simply hitting.

Why is the Construction Non-Black-Box? Following the steps above, the final construction of non-interactive commitments from hitting one-way functions that one gets has the following crucial property: In order to use the one-way function f as a hitting set generator, one needs to call f over an input of length s where s depends on the running-time of the (one-way) function f itself (on smaller input lengths). Therefore, one needs to first know the running time of f, which makes

[11] Working with a fixed polynomial n^2 as the size of the hitting set makes our negative result only stronger.

the implementation of the final construction (based on hitting one-way functions) non-black-box. Similar results where the only non-black-box use of an oracle is the knowledge of its running time were previously known (e.g., [BCKT94, GTS07, GV08]).

2.2.3 Outline of the Proof of Theorem 1.3

Before describing the main ideas and the technical tools developed and employed in the proof of Theorem 1.3 let us start by reviewing what a black-box construction based on hitting one-way functions looks like.

Black-Box Constructions from Hitting One-Way Functions. We skip a separate definition for black-box constructions based on hitting one-way functions, since this definition could be obtained from Definition 3.9 and Definition 2.4. Namely, given any oracle adversary that breaks the security of $\textsc{Com}^f$, the security reduction $\textsc{Sec}^{f,\textsc{Adv}}$, with non-negligible probability shall break the hitting one-way property of the oracle f (by either inverting f, *or* finding a circuit T of input density $d_T \geq 1/2$ that is not hit by f together with a witness of such claim).

In order to prove Theorem 1.3, we rely on the proof of Theorem 1.1 outlined in Section 2.1. A natural approach would be to show that our partially-fixed random oracle $\mathbf{O}$ is already a hitting one-way function with overwhelming probability. Doing so would prove Theorem 1.3 as a direct extension of the proof of Theorem 1.1, however, the problem with this approach is that a partially-fixed random function $\mathbf{f}$, in general, might not be a hitting function, simply because the fixed part of the randomized function $\mathbf{f}$ could be adversarially chosen to make it not hit a particular circuit T. However, recall that our oracle $\mathbf{O}_R$ relative to which the cheating receiver $\widehat{R}$ was successful, was indeed the random oracle. So in the following we start by handling the case that $\widehat{R}$ was a successful cheating receiver.

Case 1: The cheating receiver $\widehat{R}$ succeeds relative to a random oracle. It is easy to see that a random oracle is one-way with high probability.[12] We first show that a random oracle is also a hitting function with overwhelming probability (and so it will be hitting one-way).

Lemma 2.7. *For every $n \in \mathbb{N}$, with probability at least $1 - 2^{-n^2(1-o(1))}$ a random function $\mathbf{f}\colon \{0,1\}^n \mapsto \{0,1\}^n$ hits* all *co-nondeterministic circuits of size n and input density $d_T \geq 1/2$.*

Proof. Fix any co-nondeterministic circuit T of size n and input density $d_T \geq 1/2$. Any of the random images of $\mathbf{f}(j)$ for $j \in [n^2] \subseteq \{0,1\}^n$ (when truncated to the right size) will hit an element in $\mathcal{S}_T$ with probability at least the input density of T which is $d_T \geq 1/2$. Therefore, the probability that none of $\{f(1), \ldots f(n^2)\}$ hits $\mathcal{S}_T$ is at most 2^{-n^2}. Since the total number of circuits of size[13] n is at most $2^{O(n \log n)}$, the lemma follows by a union bound. □

Lemma 2.7 implies that for large enough n a random function from $\{0,1\}^n$ to $\{0,1\}^n$ is hitting with overwhelming probability. Therefore Lemma 2.7 is sufficient for refuting the existence of black-box constructions of non-interactive commitments from hitting one-way functions. Namely, for large enough n, with overwhelming probability, there exists no circuit T of size n that the security reduction $\textsc{Sec}$ (of any potential black-box construction $\textsc{Com}$) can output to refute the

[12] Recall that our random oracle chooses its randomness after the adversary is fixed and is different from the settings of [IR89, GT00] who *fix* the random oracle after sampling it once and for all.

[13] Here we denote the size of a circuit by the number of its wires.

hitting property of f. Therefore, in this case the security reduction $\textsc{Sec}$ might as well just try to invert the random oracle (with the help of the adversary). Therefore, if we are in Case 1 (where $\mathbf{O}_R$ is the random oracle), we can safely assume that we are back to the setting of Theorem 1.1 where the security reduction only tries to invert f, but we have already settled this case!

Remark 2.8 (Generalization to Separations in the Random Oracle Model)**.** The argument above can be generalized to any black-box separation result that is established through an attack in the *random-oracle model* to also handle primitives that in addition are hitting (e.g., hitting one-way functions, hitting collision resistant hash functions, etc). Thus, the result of [IR89] can be extended to separate key-agreement from hitting one-way functions.

Case 2: The cheating receiver $\widehat{R}$ *fails* relative to a random oracle. In this case, we would like to follow the general structure of Case 2 in Section 2.1, but as we mentioned before the issue is that the partially-fixed randomized oracle $\mathbf{O}_S$ might not be a hitting function. However, recall that the fixed part of $\mathbf{O}_S$ was due to the learned set $\mathcal{L}$ and the query-answer pairs inside the two *randomly* sampled views V_0 and V_1. Therefore, even though we fixed the sampled part of the oracle inside $(\mathcal{L}, \mathcal{Q}(\mathsf{V}_0), \mathcal{Q}(\mathsf{V}_1))$ and relied on the remaining randomness of $\mathbf{O}_S$ to conclude that $\mathbf{O}_S$ is one-way, this fixed part was also generated through a randomized process (even though it was fixed after being sampled). This lets us to still have a hope that the whole random process of generating $\mathbf{O}_S$ (also over the randomness of generating the fixed part at the beginning) makes the final result a hitting one-way function with overwhelming probability.

Recall that the two sampled views $\mathsf{V}_0, \mathsf{V}_1$ were obtained conditioned on ($C, \mathcal{L}$, and) the committed bit to be 0 and 1. Now suppose instead of such samples we would have sampled only one view V (for the sender and the receiver) conditioned on the values of $(C, \mathcal{L})$ but *without* conditioning the committed bit b to be 0 or 1. Then, since C was already the commitment to a random bit b, V would be a sample from the real distribution of the views of the sender and the receiver conditioned on $(C, \mathcal{L})$. Therefore, the joint samples $(C, \mathcal{L}, \mathsf{V})$ together have the same marginal distribution as $(C, \mathcal{L}, \mathsf{V}')$ where V' is the *true* view of the parties. Therefore we can conclude the following crucial property of our sampling process: If we first sample $(C, \mathcal{L}, \mathsf{V})$ to get a partial oracle over $\mathcal{F} = (\mathcal{L}, \mathcal{Q}(\mathsf{V}))$ and then choose the oracle answers to any query out of $\mathcal{F}$ at random, the final result is a random oracle. The reason simply is that this property holds for $(C, \mathcal{L}, \mathsf{V}')$ which has the same marginal distribution as that of $(C, \mathcal{L}, \mathsf{V})$; so the same should hold for $(C, \mathcal{L}, \mathsf{V})$ as well! We call such randomized *partial* functions (which are not defined over some of their inputs) *partially-defined* random functions.

Definition 2.9 (Partially-Defined Random Functions)**.** Suppose $\mathbf{f}$ is a random variable whose output is a *partial* function from $\{0,1\}^n$ to $\{0,1\}^n$ (therefore, a sample $f \leftarrow \mathbf{f}$ might be defined only over a *subset* of its domain $\{0,1\}$). Define the randomized *total* function $\widetilde{\mathbf{f}}$ over the domain $\{0,1\}^n$ (as the *random extension* of $\mathbf{f}$) as follows: First sample $f \xleftarrow{\$} \mathbf{f}$. Then for every point $x \in \{0,1\}$ which is *not* answered by $\mathbf{f}$ choose a random answer from $\{0,1\}^n$. Call the resulting function $\widetilde{f}$ (and its random variable $\widetilde{\mathbf{f}}$). If the randomized function $\widetilde{\mathbf{f}}$ is distributed exactly the same as a uniformly sampled function from $\{0,1\}^n$ to $\{0,1\}^n$, then we call $\mathbf{f}$ a *partially-defined* random function.

The New Definition of $\mathbf{O}_S$. The fact that a random extension of the randomized partial function described in $(C, \mathcal{L}, \mathsf{V})$ is a random oracle indicates that our randomized oracle $\mathbf{O}_S$ which was generated through *two* sampled views $\mathsf{V}_0, \mathsf{V}_1$ might have similar properties and be a hitting one-way function. With this intuition in mind, we change the distribution of the randomized oracle $\mathbf{O}_S$ as follows: The two sampled views V_0 and V_1 are sampled independently conditioned on $(C, \mathcal{L})$ *without* conditioning on the bit b to be 0 or 1 (just like the way V was sampled). The final (new) definition of the randomized oracle $\mathbf{O}_S$ is as follows. We first sample $(C, \mathcal{L}, \mathsf{V}_0, \mathsf{V}_1)$ as above to randomly sample a partial oracle $\mathbf{f}$, and then randomly extend it to a full oracle $\widetilde{\mathbf{f}} \equiv \mathbf{O}_S$ according to Definition 2.9. Since we would like to avoid *rejection-sampling* (not to change the marginal distributions of $(C, \mathcal{L}, \mathsf{V}_0)$ and $(C, \mathcal{L}, \mathsf{V}_1)$) if the sampled views $\mathsf{V}_0, \mathsf{V}_1$ had contradicting answers for any oracle query q we choose the answer provided by one of $\mathsf{V}_0, \mathsf{V}_1$ at random. In the following we will show that a cheating sender $\widehat{S}$ still exists relative to $\mathbf{O}_S$, and that relative to $\widehat{S}$, $\mathbf{O}_S$ remains one-way and hitting.

Concluding Theorem 1.3. The following three claims imply Theorem 1.3.

Claim 2.10. *If $\widehat{R}$ does not break the hiding of* $\textsc{Com}^{\mathbf{O}_R}$*, then there exists a malicious sender $\widehat{S}$ that breaks the binding of* $\textsc{Com}^{\mathbf{O}_S}$*.*

Proof. Since we are in the case that the cheating receiver $\widehat{R}$ is not successful, thus the distribution of the bit b conditioned on $(C, \mathcal{L})$ remains close to uniform over $\{0, 1\}$, which means that in our new way of sampling $(\mathsf{V}_0, \mathsf{V}_1)$, still with probability polynomially close to $1/2$ (and so bigger than, say, $1/3$) we get that V_0 (resp. V_1) corresponds to the bit $b = 0$ (resp. $b = 1$) used as the committed bit by the sender. Therefore by choosing the fixed part of $\mathbf{O}_S$ based on the sampled answers of $(\mathcal{L}, \mathcal{Q}(\mathsf{V}_0), \mathcal{Q}(\mathsf{V}_1))$, with $\Omega(1)$ probability we get a cheating sender $\widehat{S}$ who is able to successfully decommit to both values of the bit b using the fixed sampled view V_b. ☐

Claim 2.11 ($\mathbf{O}_S$ Remains One-Way)**.** *No* $\mathrm{poly}(n)$*-query adversary* $\textsc{Adv}$ *can invert $\mathbf{O}_S(\mathbf{U}_n)$ with probability* $1/\mathrm{poly}(n)$*, even when* $\textsc{Adv}$ *is given oracle access to the cheating sender $\widehat{S}$.*

Proof. Any query out of $\mathcal{L} \cup \mathcal{Q}(\mathsf{V}_0) \cup \mathcal{Q}(\mathsf{V}_1)$ is answered at random and the cheating sender $\widehat{S}$ is defined solely based on $(\mathcal{L}, \mathsf{V}_0, \mathsf{V}_1)$. ☐

The main techincal meat of the proof of Theorem 1.3 is found in the following claim. Due to lack to space, we only provide a very high-level outline.

Claim 2.12. *$\mathbf{O}_S$ is hitting with overwhelming probability.*

In the following we outline the proof of Claim 2.12. We would like to show that when one evaluates the oracle $\mathbf{O}_S$ over $[n^2]$ it will hit at least one of the accepted inputs of any (co-nondeterministic) circuit T of input density $d_T \geq 1/2$ with "high" probability ρ. We want the probability ρ to be extremely close to one so that we can change the order of quantifiers and conclude that $\mathbf{O}_S$ hits *all* circuits of size n.

Recall that each of the sampled partial oracles $\mathbf{f}_0, \mathbf{f}_1$ described by the query-answer pairs in $(\mathcal{L}, \mathsf{V}_0)$ and $(\mathcal{L}, \mathsf{V}_1)$ is a partially-defined random oracle, and that the final oracle $\mathbf{O}_S$ is a random extension of the "combination" of $\mathbf{f}_0$ and $\mathbf{f}_1$ (that combines the answers of f_0 and f_1 whenever their sets of queries out of $\mathcal{L}$ do not collide). The intuition is that now, over the domain $[n^2]$ (planted at the beginning of $\{0, 1\}^n$) at least half of the queries are answered randomly and independently and would behave like a random function because they either come from $\mathbf{f}_0$, or $\mathbf{f}_1$, or from the final random extension of $(\mathbf{f}_0, \mathbf{f}_1)$ to the full domain of $\{0, 1\}^n$ which we denote by $\mathbf{f}'$ (and is chosen independently of $(\mathbf{f}_0, \mathbf{f}_1)$). More formally, since $\mathbf{f}'$ is chosen independently of $(\mathbf{f}_0, \mathbf{f}_1)$, both of $(\mathbf{f}_0, \mathbf{f}')$

and $(\mathbf{f}_1, \mathbf{f}')$ are also partially-defined random oracles. Moreover, we know that over the domain $[n^2]$, at least half of the queries are answered either by $(\mathbf{f}_0, \mathbf{f}')$ or by $(\mathbf{f}_1, \mathbf{f}')$. We would like to use this property to conclude that $\mathbf{O}_S$ hits every circuit with high probability.

For any circuit T we would like to know whether there is any oracle query $q \in [n^2]$ whose answer hits $\mathcal{S}_T$ or or not. We can assume w.l.o.g that the input density d_T of T is *equal* to 1/2 and directly work with *Boolean* functions, where the output of the Boolean function indicates whether we have hit a point in $\mathcal{S}_T$ or not. We prove the following Chernoff-type concentration bound for partially-defined random functions, which we find of independent interest, and then will use it over $[m] = [n^2]$ to conclude the hitting property of $\mathbf{O}_S$.

Theorem 2.13. *let* $\mathbf{g}$ *be a partially-defined random function with domain* $\mathcal{D} = [m]$ *and range* $\mathcal{R} = \{0, 1\}$*, and let* $\mathcal{Q}(\mathbf{g})$ *be the set of queries answered in* $\mathbf{g}$*. Then for every* $k \in [m]$ *and* $\omega(\sqrt{m}) < \delta \cdot m < o(m)$ *it holds that*

$$\Pr_{g \xleftarrow{\$} \mathbf{g}} \left[|\mathcal{Q}(g)| \geq k \text{ and } \sum_{x \in \mathcal{Q}(g)} g(x) \leq (\tfrac{1}{2} - \delta) \cdot k\right] \leq e^{-(2+o(1))\delta^2 k}.$$

Employing Theorem 2.13 to Show that $\mathbf{O}_S$ is Hitting. By choosing $m = n^2$ and, $\delta = n^{-1/10}$ and $k = n^2/2$, Theorem 2.13 asserts that either of the partially-defined random oracles $(\mathbf{f}_0, \mathbf{f}')$, $(\mathbf{f}_1, \mathbf{f}')$ who answers at least $n^2/2$ queries of the domain $[n^2]$ hits every fixed circuit T at least at $\omega(n)$ points with probability at least $1 - 2^{-\omega(n)}$. Suppose w.l.o.g that it is the partial function $(\mathbf{f}_0, \mathbf{f}')$ which hits $\mathcal{S}_T$ at $\omega(n)$ points. The only way that these hitting points are "lost" in the process of combining $(\mathbf{f}_0, \mathbf{f}')$ and $(\mathbf{f}_1, \mathbf{f}')$ to get $\mathbf{O}_S$ is that all these queries are also answered by $(\mathbf{f}_1, \mathbf{f}')$ and for *all* of these $\omega(n)$ points during the (randomized) combination process the answer of $(\mathbf{f}_1, \mathbf{f}')$ is chosen over the answer of $(\mathbf{f}_0, \mathbf{f}')$. But, this would happen only with probability $2^{-\omega(n)}$. The error probability $2^{-\omega(n)}$ is small enough to allow us to take union bound over all circuits of size n and conclude that $\mathbf{O}_S$ is hitting with overwhelming probability.

3 Preliminaries

3.1 Notation

By $|x|$ we denote the length of any Boolean string x. For $m \leq |x|$, by $x|_m$ we refer to the first m bits of x. By $[k]$ we denote the set $\{1, \ldots, k\}$. We use bold letters (e.g., $\mathbf{x}$) when referring to random variables. By $x \xleftarrow{\$} \mathbf{x}$ we mean that x is sampled according to the distribution of the random variable $\mathbf{x}$. We use calligraphic letters (e.g., $\mathcal{S}$) to denote sets (e.g., events over random variables) and cryptographic primitives. We use sans-serif letters (e.g., NP) to denote complexity classes. For a set $\mathcal{S}$, by $\mathbf{U}_{\mathcal{S}}$ we mean the random variable with uniform distribution over $\mathcal{S}$, and by $x \xleftarrow{\$} \mathcal{S}$ we mean $x \xleftarrow{\$} \mathbf{U}_S$. By $\mathbf{U}_n$ we denote $\mathbf{U}_{[n]}$.

The *support* of the random variable $\mathbf{y}$, represented by $\mathrm{Supp}(\mathbf{y})$, is the set of values y such that $\Pr[\mathbf{y} = y] > 0$. For an event $\mathcal{B}$, by $\overline{\mathcal{B}}$ we denote the complement of $\mathcal{B}$ (i.e., for $\mathcal{B}$ defined over $\mathbf{x}$, it holds that $\overline{\mathcal{B}} = \mathrm{Supp}(\mathbf{x}) \setminus \mathcal{B}$). For jointly distributed random variables $(\mathbf{x}, \mathbf{y})$, and for any $y \in \mathrm{Supp}(\mathbf{y})$, the conditional distribution $(\mathbf{x} \mid y)$ is the random variable $\mathbf{x}$ conditioned on $\mathbf{y} = y$. We say that an event parameterized by n occurs with *negligible* probability, denoted by $\mathrm{negl}(n)$, if it occurs with probability $n^{-\omega(1)}$, and we say it happens with *overwhelming* probability if it happens with probability $1 - \text{negl}(n)$. We call two discrete random variables $\mathbf{x}, \mathbf{y}$ (or their corresponding distributions) ε-*close* if their statistical distance, defined as $\Delta(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \cdot \sum_{s \in \mathcal{S}} |\Pr[\mathbf{x} = s] - \Pr[\mathbf{y} = s]|$, is at most ε. We call the algorithm D an ε-*distinguisher* between the random variables $\mathbf{x}$ and $\mathbf{y}$ if $|\Pr[D(\mathbf{x}) = 1] - \Pr[D(\mathbf{y}) = 1]| \geq \varepsilon$. It is easy to see that if some algorithm D can ε-distinguish between $\mathbf{x}$ and $\mathbf{y}$, then $\Delta(\mathbf{x}, \mathbf{y}) \geq \varepsilon$.

We denote identically distributed random variables $\mathbf{x}$ and $\mathbf{y}$ by $\mathbf{x} \equiv \mathbf{y}$. We call $\{\mathbf{x}_i\}_{\mathcal{I}}$ an *ensemble* of random variables indexed by $\mathcal{I}$ if for every $i \in \mathcal{I}$ x_i is a random variable defined over $\{0,1\}^{\text{poly}(|i|)}$. When it is clear from the context, we might simply use $\mathbf{x}$ (rather than $\{\mathbf{x}_i\}_{\mathcal{I}}$) to denote an ensemble of random variables. We call two ensembles of random variables $\{\mathbf{x}_i\}_{\mathcal{I}}, \{\mathbf{y}_i\}_{\mathcal{I}}$ (with the same index set $\mathcal{I}$) *statistically close* if there is a negligible function $\varepsilon(n) = \text{negl}(n)$ such that for every $i \in \mathcal{I}$ it holds that $\mathbf{x}_i$ and $\mathbf{y}_i$ are $\varepsilon(|i|)$-close. We call $\{\mathbf{x}_i\}_{\mathcal{I}}$ and $\{\mathbf{y}_i\}_{\mathcal{I}}$ *computationally indistinguishable*, denoted by $\{\mathbf{x}_i\}_{\mathcal{I}} \approx_c \{\mathbf{y}_i\}_{\mathcal{I}}$, if for every polynomial $p(n) = \text{poly}(n)$, there exists a negligible function $\varepsilon(n) = \text{negl}(n)$ such that for every circuit D of size at most $p(n)$ and for every $i \in \mathcal{I}$, D can distinguish between $\mathbf{x}_i$ and $\mathbf{y}_i$ by at most $\varepsilon(n)$ advantage.

For a function f and a set $\mathcal{S}$, by $f(\mathcal{S})$ we denote the set $f(\mathcal{S}) = \{y \mid \exists\, x \in \mathcal{S}, y = f(x)\}$. By the *view* of an interactive algorithm A we refer to the the transcript of the messages exchanged between A and others as well as the output and the random coins of A and the oracle answers returned to A's queries. We use sans-serif letters (e.g., S) to denote the views of the parties. We always use the operation $\mathcal{Q}(\mathsf{V})$ to "extract" the queries inside V for a view V or even if V is a set of query-answer pairs.

We use the term *efficient* for any *probabilistic* algorithm that runs in polynomial time over its input. We usually denote the malicious algorithms (also called the *adversary*) with hatted letters, e.g., $\widehat{S}$ refers to an adversary who participates in a game that the honest party is called S (e.g., S is the honest sender and $\widehat{S}$ is some malicious sender).

For every two Boolean strings v, u of the same length, by $u + v$ we mean their componentwise addition modulo 2. For a list of oracle query-answer pairs $\mathcal{L}$ (which can be thought of as some partial function), and an oracle query q, we abuse the notation and let $q \in \mathcal{L}$ denote that $(q, a) \in \mathcal{L}$ for some oracle answer a.

For any circuit T, by $|T|$ we denote the size of T which counts the number of wires in T. It is easy to see that the number circuits of size n is at most $2^{O(n \log n)}$.

3.2 (Partially-Fixed) Random oracles

In this work, the random oracles are length preserving.

Definition 3.1 (Random Oracle)**.** The random oracle, denoted by $\mathbf{RO}$, is a randomized oracle which given a query $x \in \{0,1\}^n$ returns a random answer of the same length $\mathbf{RO}(x) \xleftarrow{\$} \{0,1\}^{|x|}$.

Note that here we choose to work with randomized oracles (similar to [BR93]) as opposed to previous works on black-box separations (e.g., [IR89]) which sample a random oracle and fix it forever. That is because we only aim for refuting black-box constructions rather than relativizing constructions.

Definition 3.2 (Partially-Fixed Random Oracles)**.** We call a randomized function $\mathbf{f}$ a $k(n)$-*partially-fixed random* oracle if it is fixed over some sub-domain $\mathcal{S}$ and chooses its answers similarly to the random oracle $\mathbf{RO}$ at any point q out of $\mathcal{S}$ and it holds that $\mathcal{S} \cap \{0,1\}^n \leq k(n)$ for every n. We simply call $\mathbf{f}$ partially-fixed random if it is $2^{o(n)}$-partially-fixed random.

3.3 Commitment Schemes

Definition 3.3. A (computationally secure) non-interactive commitment scheme $\textsc{Com} = (S, R)$ for a message space $\mathcal{W}_n$ is composed of an efficient sender S and an efficient receiver R such that:

- Both parties receive 1^n, where n is the security parameter. The sender uses the randomness $\mathbf{r}_S$ and the receiver uses the randomness $\mathbf{r}_R$.
- **Commitment Phase:** The sender receives a private input $w \in \mathcal{W}_n$, and outputs a commitment string $C = C(1^n, w, r_S)$. Thus $C(1^n, w)$ is a random variable whose randomness comes from $\mathbf{r}_S$. By abusing the notation we might simply denote the commitment string as $C(w)$.
- **Decommitment/Verification Phase:** The sender sends a decommitment value (w, D) to the receiver and the receiver uses the randomness $\mathbf{r}_R$ to verify (C, w, D) to accept or reject.

We desire the following properties to hold.

1. **Completeness:** When both parties follow the protocol honestly, the receiver accepts $(C(w), w, D)$ with probability $1 - \operatorname{negl}(n)$ (over $\mathbf{r}_S$ and $\mathbf{r}_R$).
2. **Hiding:** For every two sequence of inputs $(w_1, w_2, \dots), (w'_1, w'_2, \dots)$ where $\{w_i, w'_i\} \subseteq \mathcal{W}_i$, the two ensembles $\{C(1^i, w_i)\}$ and $\{C(1^i, w'_i)\}$ are computationally indistinguishable.
3. **Binding:** Suppose $\widehat{S}$ is an efficient malicious sender who first sends some commitment string C, then receives some input w, and then tries to decommit into (w, D_w) successfully. Roughly speaking, the binding property asserts that any such sender after sending C is able to decommit successfully into at most one string w. More formally we call the scheme (α, β)-binding if the following holds: With probability at least α over the generation of C, there exists some value $w \in \mathcal{W}_n$ such that for every other value $w' \neq w$ we have $\Pr[R(C, w', D_{w'}) \text{ rejects}] \geq \beta$ where the probability is over $\mathbf{r}_R$ and the randomness of $\widehat{S}$ in generating $D_{w'}$. We simply call the scheme binding if it is (α, β)-binding for $\alpha, \beta \geq 1 - \operatorname{negl}(n)$, and call it weakly-binding if it is (α, β)-binding for some $\alpha, \beta \geq 1/\operatorname{poly}(n)$.

Perfect Binding. Note that if we want a non-interactive commitment scheme to be binding against *non-uniform* cheating senders, then any (α, β)-binding scheme according to Definition 3.3 is already $(1, \beta)$-binding (and it is in fact *perfectly* binding, if the verification is deterministic). The reason is that, if for any commitment string C there exist (w, D_w) and $(w', D_{w'})$ for $w \neq w'$ such that the receiver would accept both of (C, w, D_w) and $(C, w', D_{w'})$ with probability at leat $1 - \beta$, then a non-uniform cheating sender $\widehat{S}$ can "know" these values through its non-uniform advice. Again, since we are proving in an impossibility result, we will to work with the more general setting where the binding is not perfect and it is proved by a computational reduction to the security of the primitive used (e.g., the one-way function).

Definition 3.4 ([BMO90, IOS97])**.** A (computationally secure) *instance-based* non-interactive commitment scheme for the language L is a two-party protocol (S, R) between an efficient sender S and an efficient receiver R such that:

- Both parties receive some x as input, where $|x| = n$ is the security parameter.
- The commitment and decommitment phases are the same as in Definition 3.3.

- If $x \in L$, then the completeness and hiding properties hold the same as in Definition 3.3.
- If $x \notin L$ then the binding property holds the same as in Definition 3.3.

Even though we only gave a formal definition for *non-interactive* commitments, we shall review the two-message commitment scheme of Naor for proving Theorem 1.2.

Construction 3.5 (Naor's Two-Message Commitment [Nao91])**.** Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a one-way function. First we the black-box construction [HILL99] over f to get a pseudorandom generator $g\colon \{0,1\}^m \mapsto \{0,1\}^{3m}$ for some $m = \mathrm{poly}(n)$. Now suppose the sender holds a bit b. The commitment phase has two messages as follows, and the decommitment phase is canonical.

1. The receiver R chooses $r \xleftarrow{\$} \{0,1\}^{3m}$ and sends r to the sender.
2. The sender S chooses $s \xleftarrow{\$} \{0,1\}^m$. If $b = 0$, it takes $c = f(x)$, and if $b = 1$, it takes $c = f(x)+r$ (where the addition is componentwise modulo 2) and sends c to the receiver.

The hiding property of Construction 3.5 is implied by the pseudorandomness of f for *every* first message of the verifier. The (statistical) binding property holds with overwhelming probability over the randomness of r. Namely, by a union bound, with probability at least $1-2^m \cdot 2^m \cdot 2^{-3m} = 1-2^{-m}$ over the choice of r, the two sets $f(\{0,1\}^m)$ and $f(\{0,1\}^m)+r$ are disjoint, in which case the scheme is perfectly binding.

3.4 Black-Box Constructions and Separations

We refer the reader to [RTV04] for a comprehensive discussion on black-box constructions and their variants. In the following we define black-box constructions[14] in a general way, and then will tailor them specifically for our specific primitives of interest.

Definition 3.6 (Black-Box Constructions)**.** A black-box implementation Q^P of a primitive $\mathcal{Q}$ from another primitive $\mathcal{P}$ is an oracle algorithm Q (called the implementation reduction) such that Q^P is an implementation of $\mathcal{Q}$ for any oracle P that implements $\mathcal{P}$. We say that Q^P has a black-box proof of security, if there exists an efficient oracle algorithm $\textsc{Sec}$ such that for any oracle P implementing $\mathcal{P}$ and for any (computationally unbounded) adversary $\textsc{Adv}$ who breaks the security of Q^P (as an implementation of $\mathcal{Q}$) with non-negligible advantage for some security parameter n, the oracle algorithm $\textsc{Sec}^{P,\textsc{Adv}}$ breaks the security of P over a polynomially related security parameter $n' = n^{\Theta(1)}$. We say that $\mathcal{Q}$ has a black-box construction from $\mathcal{P}$ if there is a black-box implementation Q and a black-box proof of security $\textsc{Sec}$ as above.

Remark 3.7 (Why using the adversary only over one security parameter n?)**.** If a black-box reduction wants to convert *any* infinite sequence of successful adversaries $\{A_{n_1}, A_{n_2}, \ldots\}$ breaking the security of the implementation Q^P over the security parameters $\{n_1, n_2, \ldots\}$ into another infinite sequence of successful adversaries $\{A'_{n'_1}, A'_{n'_2}, \ldots\}$ breaking the implementation P over the security parameters $\{n'_1, n'_2, \ldots\}$, we can always assume w.l.o.g that the security reduction $\textsc{Sec}$ does *not* call the adversary A_m over different values of m. The reason is that if the set $\{n_1, n_2, \ldots\}$ is extremely sparse (whose sparsity can be chosen *after* the security reduction is fixed) an execution of $\textsc{Sec}^{P,A}$ might be able to only use $A = A_n$ over a *single* "good" index $n \in \{n_1, n_2, \ldots\}$ and the

[14] What we call black-box here is denoted as *fully* black-box in the terminology of [RTV04].

oracle A_m could be completely useless on other parameters of m called by $\textsc{Sec}^{P,A}$, and since there is no guarantee over the quality of such adversaries used, it can completely void the promise of the security reduction to break the used implementation P. Based on this explanation, it is of no surprise that all the black-box constructions in the literature fall into the category of Definition 3.6.

However, we note that there are some black-box constructions that transform any adversary A who is successful *almost everywhere* (i.e., over all security parameters n larger than some n_0) into another adversary A' who breaks the implementation P also almost everywhere. In this case it makes perfect sense to allow the security reduction $\textsc{Sec}$ to call A_m over many security parameters m. Such constructions are particularly known in the context of proving that some closely-related variants of "one-way functions" are *necessary* for the existence of other primitives (e.g., [IL89]). However, we choose to work with Definition 3.6 since here we are interested in "almost-everywhere security" which is the standard definition used in the theory of cryptography for the security of the primitives.

Remark 3.8 (Security Parameters)**.** The reason for $n' \in n^{\Theta(1)}$ in Definition 3.6 is that we are using super-polynomial timed (or sized) adversaries. If the used primitive is, say, exponentially secure, it is possible for the security reduction to work with much smaller values of n'. On a different note, in general, it is possible to design a black-box construction of a primitive $\mathcal{Q}$ (e.g., non-interactive commitments) based on another primitive $\mathcal{P}$ in which the security reduction $\textsc{Sec}$ breaks the implementation P over some security parameter $n' = n^{\Theta(1)}$ given oracle access to any adversary that breaks the implementation Q^P over the security parameter n while the reduction itself does not know *explicitly* for what n' it will be the case. However, since $n' \in n^{\Theta(1)}$ and since we only aim to break the security with $1/\operatorname{poly}(n')$ probability, by pigeonhole principle there is always some *fixed* $n' \in n^{\Theta(1)}$ over which the security reduction performs well, and we can think of a well-defined mapping $n' = n'(n)$ with respect to the security reduction S.

3.4.1 Black-Box Constructions of Commitments

Now we give a formal definition for the case of black-box constructions of non-interactive commitments from one-way functions.

Definition 3.9. A black-box construction of non-interactive commitments from one-way functions is a pair of efficient oracle algorithms (S, R) such that: The parties receive the common input 1^n as the security parameter and access an oracle $f = \{f_m \colon \{0,1\}^m \mapsto \{0,1\}^m\}$. The commitment and decommitment phases are the same as in Definition 3.3, and we require the completeness conditions to hold for *every* oracle f. The security of the scheme is guaranteed through reductions to the one-way feature of f as follows.

- **Proving the Hiding:** There is an efficient security reduction H that proves the hiding property based on the one-way property of f. Namely, for every oracle f and every malicious receiver $\widehat{R}$ (who could arbitrarily depend on f) given as an oracle, every $\varepsilon \geq 1/\operatorname{poly}(n)$, and every pair of messages $w_0 \neq w_1$ from the space of messages $\{w_0, w_1\} \subseteq W$ such that $\widehat{R}$ distinguishes between $C(w_0)$ and $C(w_1)$ with non-negligible advantage ε, the oracle algorithm $H^{f,\widehat{R}}$ breaks the one-way property of f with probability at least $\operatorname{poly}(\varepsilon/n) = \operatorname{poly}(\varepsilon/n')$ over a polynomially-related $n' = n^{\Theta(1)}$ input length; namely:
$$\Pr_{y \xleftarrow{\$} f(\mathbf{U}_{n'})}[H^{f,\widehat{S}}(y) \in f^{-1}(y)] \geq \left(\frac{\varepsilon}{n'}\right)^{O(1)}.$$

- **Proving the Binding:** It is defined similarly to the definition of Hiding using another reduction B that inverts f with non-negligible probability given oracle address to f and any adversary who breaks the binding of the constructed commitment scheme.

Perfect Binding. It is true that when the adversary is non-uniform, w.l.o.g one can work with *perfectly binding* non-interactive commitment schemes. However, as we said before, in this work we want to rule out even constructions with *uniform* proofs of security that start with uniformly secure primitives (e.g., , when one uses *uniform* hitting OWFs in Theorem 1.2). Therefore, we choose to work with the definition of binding according to Definition 3.9 as opposed to a possible simpler definition tailored for perfect binding. In particular we note that Theorem 1.2 can only rely on *uniform* hitting one-way functions, and as another examle: by assuming the existence of *uniformly-secure* collision-resistant hash functions (which do not have an index and can be though of as a stronger primitive than *families* of collision-resistant hash functions), one can obtain non-interactive commitment schemes which are only computationally binding.

Black-Box Constructions from Other Primitives. A black-box construction of non-interactive commitments from a primitive $\mathcal{P}$ other than one-way functions (e.g., FCRH) can be defined similarly to Definition 3.9. For example, for the case of FCRH, the parties get access to an oracle $h = \{h_i \colon \{0,1\}^i \times \{0,1\}^i \mapsto \{0,1\}^{i/2}\}_{i \in \mathbb{N}}$, (where $h(d, \cdot)$ is supposedly collision-resistant for a randomly chosen index d). Again the completeness property should hold for every oracle h, and there will be two security reductions that both break the collision-resistance of h over a random index $d \in \{0,1\}^{n'}$ for $n' = n^{\Theta(1)}$ given oracle access to any adversary who breaks the hiding or binding property of the scheme over security parameter n.

Instance-Based Commitments. A black-box construction of an instance-based non-interactive commitments (from one-way functions or other primitives) is also defined similarly to Definition 3.9 by adopting the hiding and binding properties to the instance-based setting of Definition 3.4. Namely, a successful cheating sender $\widehat{S}$ (which is given as an oracle to the reduction B) should break the binding over some input $x \notin L$, and the successful cheating receiver $\widehat{R}$ (given as oracle to the reduction H) should break the hiding property of the commitment scheme over some $x \in L$.

3.4.2 Randomized Implementation of Primitives

Definition 3.10 (Security Threshold and Security-Transparent Primitives)**.** We say a primitive $\mathcal{P}$ has a *security threshold* $\mathcal{P}$ if any adversary who breaks $\mathcal{P}$ has to "output" some special string in its security game with probability $\tau_{\mathcal{P}} + \varepsilon$ for some non-negligible probability ε. We call a primitive *security-transparent* if its security-threshold is zero.

Examples. One-way functions and FCRH are security transparent because it is enough to find an inverse or a collision with a non-negligible probability to break the primitive. and the security threshold of pseudorandom-generators and block-ciphers (and other natural "indistinguishability-based" primitives) is $1/2$. For some other primitives such as fair coin-tossing, the adversary tries to make the *other party* output some string by an advantage over $\tau_{\mathcal{P}}$ (which in case of coin-tossing it is $\tau_{\mathcal{P}} = 1/2$).

Security Threshold of Commitment. The security definition of commitment schemes uses *two* security thresholds (and in general for multi-party primitives every party could have its own security threshold). The binding property has security threshold 0 (because the cheating sender wants to win with non-negligible probability), but the hiding property has security threshold 1/2 (because the cheating receiver wants to distinguish between the commitments of two different messages). But in this work we deal with security transparent primitive that are *used* in a construction of commitments (e.g., one-way functions and FCRH).

Definition 3.11. Let $\mathcal{O}$ be a *set* of *randomized* oracles (e.g., the set of all partially-fixed random oracles). We say that a primitive $\mathcal{P}$ (e.g., one-way functions or FCRH) has a secure black-box implementation using $\mathcal{O}$, if there is a $\text{poly}(n)$-time implementation algorithm $P^{\mathbf{O}}$ for $\mathcal{P}$ such that for every randomized oracle $\mathbf{O} \in \mathcal{O}$, any computationally-unbounded adversary A who (only) knows the distribution of $\mathbf{O}$ and asks $\text{poly}(n)$ queries to $\mathbf{O}$ can break $P^{\mathbf{O}}$ with advantage at most $\text{negl}(n)$ (above the security threshold $\tau_{\mathcal{P}}$).

An important primitive that can be securely realized (even with exponential security) from the set of partially-fixed random oracles is FCRH. We emphasize that having an *index* for the hash function (and thus making it a *family* of hash functions) is necessary for deriving this primitive from partially-fixed random oracles. That is because for any k-query construction of hash functions $h^f \colon \{0,1\}^i \mapsto \{0,1\}^{i/2}$ from the oracle f, one can always fix $2k$ points of f to guarantee a collision which could be known to the adversary attacking the collision resistance of h^f since the adversary knows the distribution of the function f used (and the fixed part is part of the description of the distribution).

Lemma 3.12. *FCRH can be black-box securely realized from all partially-fixed random oracles.*

Proof. Let $f \colon \{0,1\}^n \mapsto \{0,1\}^n$ be a partially-fixed random oracle which is randomly chosen on any point out of a fixed set $\mathcal{S}$ which $\mathcal{S}_n = \mathcal{S} \cap \{0,1\}^n \leq 2^{o(n)}$. Consider the following construction of FCRH $h \mid h \colon \{0,1\}^{n/2} \times \{0,1\}^{n/2} \mapsto \{0,1\}^{n/4}$ from f: For every $d, x \in \{0,1\}^{n/2}$, $h(d,x)$ is equal to the first $n/4$ bits of $f(d,x)$. We prove that the construction above is black-box secure according to Definition 3.11. Call d a *bad* index if there exist some x such that $(d,x) \in \mathcal{S}$, and call it a *good* index otherwise. Since $|\mathcal{S}_n| = 2^{o(n)}$, a random index $d \xleftarrow{\$} \{0,1\}^{n/2}$ is a bad index only with probability at most $2^{o(n)}/2^{n/2}$.

Now suppose a computationally unbounded adversary A is given some good index d and tries to find collision in the function $h_d(\cdot)$. Since d is a good index, $h_d(\cdot)$ will be a random function from $\{0,1\}^{n/2}$ to $\{0,1\}^{n/4}$. It is easy to see that a q-query attacker can find a collision in a random function to a domain of size N only with probability $O(q^2/N)$. Therefore, for a good index d, a $\text{poly}(n)$-query adversary A is able to find a collision only with probability $\text{poly}(n)/2^{n/4}$. Therefore by a union bound the chance of A to find a collision (over the randomness of h) is at most $2^{o(n)}/2^{n/2} + \text{poly}(n)/2^{n/4} < \text{negl}(n)$. □

3.4.3 Refuting Black-Box Constructions

A similar argument to that of Lemma 3.13 below for the special case of $\mathbf{O} = \mathbf{RO}$ is implicit in [BM07] and [DSLMM11]. Here we need this lemma for the case that $\mathbf{O}$ is a partially-fixed random oracle. A more general lemma (also incorporating the case that $\mathcal{P}$ is not security-transparent) has appeared in [MP].

Lemma 3.13. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two cryptographic primitives and $\mathcal{P}$ is security transparent. For a randomized oracle $\mathbf{O}$, suppose one can break the black-box security of* any *implementation $Q^{\mathbf{O}}$ of $\mathcal{Q}$ with non-negligible probability and asking* $\operatorname{poly}(n)$ *oracle queries to $\mathbf{O}$. Suppose also that there exists a black-box secure implementation P of $\mathcal{P}$ from $\mathbf{O}$. Then there is no black-box construction of $\mathcal{Q}$ from $\mathcal{P}$.*

Proof. Suppose on the contrary that (Q, S) is a black-box construction of $\mathcal{Q}$ from $\mathcal{P}$. By feeding the randomized implementation $P^{\mathbf{O}}$ of $\mathcal{P}$ to the implementation Q of $\mathcal{Q}$ we get $Q^{P^{\mathbf{O}}} = (Q^P)^{\mathbf{O}}$ as a randomized implementation of $\mathcal{Q}$ using $\mathbf{O}$. Since we assumed that any such implementation is insecure, therefore there is some (computationally unbounded) adversary A who breaks the security of $(Q^P)^{\mathbf{O}}$ with non-negligible advantaging $\varepsilon(n) > 1/\operatorname{poly}(n)$ (above $\tau_{\mathcal{Q}}$) for security parameter n by asking only $m = \operatorname{poly}(n)$ number of oracle queries to $\mathbf{O}$.

Call an oracle $O \xleftarrow{\$} \mathbf{O}$ a *good* oracle if A breaks $(Q^P)^O$ (as an implementation of $\mathcal{Q}$ for) with advantage at least $\varepsilon(n)/2$. An averaging argument shows that a random $O \xleftarrow{\$} \mathbf{O}$ is good with probability at least $\varepsilon(n)/2$. For every good oracle $O \xleftarrow{\$} \mathbf{O}$, since it holds that A breaks $(Q^P)^{\mathbf{O}}$ with advantage at least $\varepsilon(n)/2$, therefore the security reduction S^{P^O, A^O} would break P^O over some security parameter $n' = n^{\Theta(1)}$ with probability at least $\delta = \operatorname{poly}(\varepsilon(n)/n') > 1/\operatorname{poly}(n')$.

Note that we can combine the algorithms S, P, and A to get an algorithm $S^{P,A}$ who queries at most $\operatorname{poly}(n) \cdot m \le \operatorname{poly}(n')$ oracle queries and breaks the security of P^O with probability $\delta(n') > 1/\operatorname{poly}(n')$ whenever O is a good oracle. Thus if we choose $O \xleftarrow{\$} \mathbf{O}$ the attacker $S^{P,A}$ still succeeds in breaking P^O with a non-negligible probability at least $\delta'(n') = (\varepsilon(n)/2) \cdot \delta(n') > 1/\operatorname{poly}(n')$. Since we assumed $\mathcal{P}$ to be security transparent the success probability $\delta'(n)$ is already non-negligibly above the security threshold $\tau_{\mathcal{P}} = 0$. Therefore $S^{P,A}$ breaks the black-box security of $P^{\mathbf{O}}$ (over the security parameter n') which is a contradiction. $\square$

4 Separation from Partially-Fixed Random Oracles

In this section we will prove the following theorem.

Theorem 4.1. *Suppose there exists a secure implementation of some primitive $\mathcal{P}$ from partially-fixed random oracles (see Definition 3.2) where $\mathcal{P}$ is security-transparent (see Definition 3.10). Then there exists no black-box construction of non-interactive commitments from $\mathcal{P}$ even for the message space $\mathcal{W} = \{0, 1\}$.*

Partially-fixed random oracles imply (security-transparent) primitives such as one-way functions and FCRHs and Theorem 4.1 implies Theorem 1.1 as a corollary. In the rest of this section we prove Theorem 4.1.

The intuition behind the proof is to find a $\operatorname{poly}(n)$-query attacker to the scheme from some partially-fixed random oracle and apply Lemma 3.13. More specifically, we first design a natural cheating strategy $\widehat{R}$ for the receiver which is computationally unbounded, but asks only $\operatorname{poly}(n)$ number of queries to its (potentially randomized) oracle $\mathbf{f}$. Then, we show that either the algorithm $\widehat{R}$ would succeed in guessing the bit b with probability $1/2 + 1/\operatorname{poly}(n)$ in the *random* oracle model $\mathbf{f} \equiv \mathbf{RO}$, or that there exists a cheating strategy $\widehat{S}$ who breaks the binding property in a model where the randomized oracle $\widetilde{\mathbf{f}}$ used is partially-fixed random.

Comparing the Notation with Section 2.1. In section 2.1 we denoted the first randomized oracle suitable for $\widehat{R}$ by $\mathbf{O}_R$ and the second randomized oracle suitable for $\widehat{S}$ by $\mathbf{O}_S$. Here we no longer use those names.

Lemma 4.2 below carries the heart of the proof. In this section we will use Lemma 4.2 only for the simple case of $\mathcal{W} = \{0, 1\}$, but we state and prove this lemma for the more general case of $|\mathcal{W}| = \text{poly}(n)$ because of its application to the proof of Theorem 1.4.

Lemma 4.2. *For any black-box implementation (S, R) of non-interactive commitment from the oracle f (regardless of whether the scheme is secure or not) and the message space $\mathcal{W}$ of size $|\mathcal{W}| \le \text{poly}(n)$ in which the parties ask m oracle queries, and for any given parameter $\delta < 1/100$, there are two cheating strategies: $\widehat{S}$ for the sender and $\widehat{R}$ for the receiver such that at least one of the following cases holds.*

1. *$\widehat{R}$ asks $O(m/\delta^2)$ oracle queries, and there are two messages $\{w_0, w_1\} \subseteq \mathcal{W}$ such that: if the oracle is a random function $\mathbf{f} \equiv \mathbf{RO}$, then $\widehat{R}$ can distinguish between w_0 and w_1 with advantage at least δ. We call such a receiver $\widehat{R}$ a δ-successful (cheating) receiver w.r.t. the random oracle (and messages (w_0, w_1)).*

2. *There is a $O(\frac{m}{\delta^2} + m|\mathcal{W}|)$-*partially-fixed *random oracle $\widetilde{\mathbf{f}}$ such that whenever it is used in the commitment scheme, $\widehat{S}$ can send a commitment C and then open it successfully into* every *$w \in \mathcal{W}$ with probability at least $1 - \delta'$ for $\delta' = (m \cdot |\mathcal{W}|)^{O(1)} \cdot \delta^{\Omega(1)} + \text{negl}(n)$. We call $\widehat{S}$ a $(1 - \delta')$-successful (cheating) sender w.r.t. the partially-fixed random oracle $\widetilde{\mathbf{f}}$.*

Note that if $|\mathcal{W}| \le \text{poly}(n)$ and $m = \text{poly}(n)$, we can always take $\delta = 1/\text{poly}(n)$ to be small enough so that $\delta' < 1/100$. In both cases of Lemma 4.2 we get an adversary (either a cheating sender or a cheating receiver) that breaks the security of the commitment scheme w.r.t. a partially-fixed random oracle by asking only $\text{poly}(n)$ oracle queries (a random oracle $\mathbf{RO}$ can also be thought of as a partially-fixed random oracle which is fixed over zero elements of its domain). Therefore, Theorem 4.1 follows directly from Lemma 3.13.

Now we prove Lemma 4.2.

Proof of Lemma 4.2. In the following we will assume that (S^f, R^f) is a black-box implementation of non-interactive commitments from the oracle f, and we assume that the oracle f is sampled from $f \xleftarrow{\$} \mathbf{f}$ where $\mathbf{f}$ is the random oracle $\mathbf{f} \equiv \mathbf{RO}$. We first present the cheating receiver strategy $\widehat{R}$, and then assuming that it is not δ-successful w.r.t. $\mathbf{RO}$ we derive the needed (cheating) sender strategy $\widehat{S}$ and its partially-fixed random oracle $\widetilde{\mathbf{f}}$ such that $\widetilde{R}$ is $(1 - \delta')$-successful w.r.t. $\widetilde{\mathbf{f}}$.

Before describing the cheating algorithms $\widehat{R}$ and $\widehat{S}$, we need to borrow a tool from [BM07]. Barak and Mahmoody [BM07] proved the following lemma in a more general *interactive* setting, but here we only specify it in a special case which is sufficient for us.

Lemma 4.3 (Learning Heavy Queries Efficiently [BM07])**.** *Let A be a randomized algorithm which asks up to m oracle queries to the random oracle $\mathbf{RO}$, denoted by the set $\mathcal{Q}(A)$ and outputs some message C. Let $0 < \varepsilon < 1$ be a given parameter. There is learning algorithm G in $\mathsf{BPP}^{N}P$ which (given an NP oracle run in polynomial time and) learns a list $\mathcal{L}$ of query-answer pairs from the oracle $\mathbf{RO}$ and the following two conditions hold.*

1. *Efficiency of the learner: $|\mathcal{L}| \le 10m/\varepsilon^2$.*

2. *Learning heavy queries: With probability at least $1-\varepsilon$ over the choice of* **RO** *and the random coins of A and G, for* every *$q \notin \mathcal{L}$ it holds that $\Pr[q \in \mathcal{Q}(A) \mid (C, \mathcal{L})] < \varepsilon$ where the latter probability is over the remaining randomness of* **RO** *and A conditioned on $(C, \mathcal{L})$.*

We employ the $\mathsf{BPP}^{\mathsf{NP}}$ complexity of the learner in Section 6. Now we describe our cheating receiver $\widehat{R}$.

Construction 4.4 (The Cheating Receiver $\widehat{R}$ with Parameter δ)**.** Let m be the total number of oracle queries asked by the sender and the receiver (during the verification).

1. The cheating receiver $\widehat{R}$ first runs the learning algorithm of Lemma 4.3 with parameter $\varepsilon = \delta$ over the algorithm A which is composed of *both* the sender's algorithm when committing to a random message $\mathbf{w} \xleftarrow{\$} \mathcal{W}$ *continued* with the execution of the verification algorithm. Even though the verification is not executed yet, the learner $\widehat{R}$ can simply "imagine" that it is already executed. Thus, the randomness of A will be $(\mathbf{r}_S, \mathbf{r}_R, \mathbf{w})$ and its output will be the commitment C.

2. Let $\mathbf{X}$ be the random variable that includes the view of $\widehat{R}$ at the end of the learning phase. The content of $\mathbf{X}$ includes the commitment C and the learned oracle query-answer pairs $\mathcal{L}$. If there exists any pair $(w_0, w_1) \in \mathcal{W}^2$ such that $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \geq \delta$, then $\widehat{R}$, when the messages are restricted to $\{w_0, w_1\}$, can always output the more likely input among $\{w_0, w_1\}$ conditioned on its view $\mathbf{X}$, and because $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \geq \delta$, this will make a δ-successful receiver strategy w.r.t. the inputs $\{w_0, w_1\}$ and the random oracle.

Note that by the efficiency property of Lemma 4.3, $\widehat{R}$ asks at most $10(m/\delta^2)$ number of queries. Thus, as we also mentioned in the description of $\widehat{R}$, if there are two inputs $(w_0, w_1) \in \mathcal{W}^2$ such that $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \geq \delta$ we are done with the proof of Lemma 4.2. So in the following we assume that no such pair exists. The description of the cheating sender $\widehat{S}$ is as follows.

Construction 4.5 (The Cheating Sender $\widehat{S}$ and the Partially-Fixed Random Oracle $\widetilde{\mathbf{f}}$)**.**

1. First sample $(C, \mathcal{L})$ according to the first step of the cheating receiver $\widehat{R}$ in Construction 4.4 (by internally simulating a random oracle $\mathbf{f}$ and throwing it away at the end). Recall that C is the commitment to a randomly chosen message $\mathbf{w} \xleftarrow{\$} \mathcal{W}$.

2. Then for every $w \in \mathcal{W}$ sample a view S_w for the sender from the distribution of sender's view conditioned on $\mathbf{w} = w$ (and if $\Pr[\mathbf{w} = w \mid (C, \mathcal{L})] = 0$, let $\mathsf{S}_w = \bot$).

3. The partially-fixed random oracle $\widetilde{\mathbf{f}}$ will be fixed over the set $\mathcal{F} = \bigcup_w \mathcal{Q}(\mathsf{S}_w)$, and is random at any other point. Below we describe how $\widetilde{\mathbf{f}}$ is defined over the sub-domain $\mathcal{F}$. Once this part is fixed, the distribution of $\mathbf{f}$ will be random at any other point.

   - If $x \in \mathcal{Q}(\mathcal{L})$ (i.e., x is learned in the first step), use the answer of $\mathcal{L}$.
   - Otherwise, let $\mathcal{U}_x = \{w \mid x \in \mathcal{Q}(\mathsf{S}_w)\}$ be the set of messages whose corresponding sampled views have an answer defined for the query x. Then choose $u_x \xleftarrow{\$} \mathcal{U}_x$ at random once and for all, and set $\widetilde{\mathbf{f}}(x)$ equal to the answer specified for x in the view S_{u_x}.[15]

[15]We emphasize that even though for the purpose of proving Theorem 1.1 we can define the oracle answers in this case arbitrarily, but we used this randomized version of the definition of $\mathcal{F}$ to facilitate the proof of Theorem 1.3.

As usual (following our abuse of notation), we might use $\mathcal{F}$ both to denote the set of fixed queries and also the set of fixed queries together with their answers.

4. The cheating sender $\widehat{S}$ sends C as its commitment. Then in order to decommit to any message w, $\widehat{S}$ uses the sample view S_w to derive the required decommitment string D_w.

The fixed set $\mathcal{F}$ above describes the distribution of the partially-fixed random oracle $\widetilde{\mathbf{f}}$, and the values of C and $\{D_w\}_{w\in\mathcal{W}}$ describe the behavior of the cheating sender $\widehat{S}$ in its decommitment phase.

Claim 4.6. *Assuming that* $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \le \delta$ *for every pair* $(w_0, w_1) \in \mathcal{W}^2$ *(i.e., that* $\widehat{R}$ *fails), if we run* $\widehat{S}$*, and ask it to decommit to every* $w \in \mathcal{W}$*, with probability at least* $1-\delta''$*, the verifications of* (C, w, D_w) *will be accepted for* all $w \in \mathcal{W}$*, where* $\delta'' = (m \cdot |\mathcal{W}|)^{O(1)} \cdot \delta^{\Omega(1)} + \mathrm{negl}(n)$.

Proof. The proof is through a hybrid argument based on two experiments. The first experiment is just the real experiment that we deal with during the cheating sender's attack.

Experiment Real**.** Sample $(\mathcal{L}, C, \{\mathsf{S}_w\}_{w\in\mathcal{W}}, \mathcal{F})$ through the process of Construction 4.5 (which would determine $\{D_w\}_{w\in\mathcal{W}}$), then choose the random coins $\mathbf{r}_w$ for the receiver for each $w \in \mathcal{W}$ independently at random, and choose a single $\widetilde{\mathbf{f}} \xleftarrow{\$} (\mathbf{RO} \mid \mathcal{F})$. Finally, for every $w \in \mathcal{W}$, let R_w be the view of the receiver's verification when executed over (C, w, D_w) using the randomness $\mathbf{r}_w$ and the oracle $\widetilde{\mathbf{f}}$.

Experiment Imag**.** The difference between Imag and Real is that in Imag to verify (C, w, D_w) we ignore the sampled views $\mathsf{S}_{w'}$ for any other $w' \neq w$ and will use a random oracle which is chosen by *only* conditioning on $(\mathcal{L}, \mathsf{S}_w)$. More formally, we first sample $(\mathcal{L}, C, \{\mathsf{S}_w\}_{w\in\mathcal{W}})$ through the process of Construction 4.5 and then will choose the random coins $\{\mathbf{r}_w\}_{w\in\mathcal{W}}$ independently at random for the receiver. After that, for each $w \in \mathcal{W}$, we execute the commitment verification over (C, w, D_w) using the randomness $\mathbf{r}_w$, and for each new query $q \notin \mathcal{Q}(\mathsf{S}_w) \cup \mathcal{Q}(\mathcal{L})$ we choose a fresh random answer (even though this answer might be inconsistent with the answers chosen in $\mathcal{Q}(\mathsf{S}_{w'})$ for some $w' \neq w$). Finally we let R_w to be the view of such verification.

We emphasize that in Experiment Imag we do *not* sample a full instance of any partially-fixed random oracle $\widetilde{\mathbf{f}}$, and we only sample the answers to the queries that are required for verifications "on demand" (which in fact as we mentioned might not be sampled all consistently).

Output of the Experiments. The output of both experiments is a random variable containing the tuple $(C, \mathcal{L}, \{\mathsf{R}_w\}_{w\in\mathcal{W}})$ from that experiment which includes also the decision of the receivers (to accept or reject). We use $\mathbf{Out}_R$ to denote the output of Real and use $\mathbf{Out}_I$ to denote the output of Imag.

Claim 4.7. $\Pr_{\mathsf{Imag}}[\forall\, w \in \mathcal{W}, \mathsf{R}_w \text{ accepts}] \ge 1 - |\mathcal{W}| \cdot (\delta + \mathrm{negl}(n))$ *which is least* $1 - O(|\mathcal{W}| \cdot \delta) - \mathrm{negl}(n)$ *for* $|\mathcal{W}| = \mathrm{poly}(n)$.

Proof. We only prove $\Pr_{\mathsf{Imag}}[\mathsf{R}_w \text{ accepts}] \ge 1 - (\delta + \mathrm{negl}(n))$ for a fixed $w \in \mathcal{W}$, and the claims follows by a union bound. For a moment suppose C was generated as the commitment to this particular w rather than the commitment to a random message. In this case the sampled $(\mathsf{R}_w, C, \mathcal{L})$ in Imag will have the same marginal distribution to that of the following Experiment in the random oracle model, called Ideal, in which there is no adversary and the sender honestly commits to w and the receiver $\widehat{R}$ runs its learning algorithm to learn $\mathcal{L}$. It is clear that in Ideal we can go ahead and execute the verification of (C, w, D_w) using a lazy evaluation of the random oracle (i.e., answering any new query at random) and thus by the completeness of the commitment scheme the verification should accept by probability $1 - \operatorname{negl}(n)$ (i.e., $\Pr_{\mathsf{Ideal}}[\mathsf{R}_w \text{ accepts}] \ge 1 - \operatorname{negl}(n)$).

The verification in both of Ideal and Imag uses the same lazy evaluation; their only difference is the way $\mathbf{X} = (C, \mathcal{L})$ is sampled. But recall that here we are assuming that $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \le \delta$ for every pair $(w_0, w_1) \in \mathcal{W}^2$, and thus we will have $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), \mathbf{X} \le \delta$ as well (because $\mathbf{X}$ can be thought of sampling $\mathbf{X}$ conditioned on a random w). Therefore the statistical distance between $(C, \mathcal{L})$ in Imag compared to $(C, \mathcal{L})$ in Ideal is at most δ. Since the statistical distance can not be increased by applying any function we conclude that $\Pr_{\mathsf{Imag}}[\mathsf{R}_w \text{ accepts}] \ge 1 - (\delta + \operatorname{negl}(n))$. □

For every $(C, F, \{\mathsf{S}_w\}, \{\mathsf{R}_w\})$ (of Real or Imag) we define the "bad" event $\mathcal{B}$ to hold if and only if there exists some $w' \neq w$ such that $(\mathcal{Q}(\mathsf{S}_w) \cup \mathcal{Q}(\mathsf{R}_w)) \cap (\mathcal{Q}(\mathsf{S}_{w'}) \cup \mathcal{Q}(\mathsf{R}_{w'})) \nsubseteq \mathcal{Q}(\mathcal{L})$ (we define $\mathcal{Q}(\perp) = \varnothing$ in case $\Pr[\mathbf{w} = w \mid \mathcal{L}, C] = 0$ for some values of $\mathcal{L}, C, w \in \mathcal{W}$).

Claim 4.8. *Conditioned on $\overline{\mathcal{B}}$, the output of the Experiments* Real *and* Imag *are identically distributed.*

Proof. Fix a pair $(w, w') \in \mathcal{W}^2$. One possibility because of which Real and Imag might deviate is when it happens that $\mathcal{Q}(\mathsf{S}_w) \cap \mathcal{Q}(\mathsf{S}_{w'}) \nsubseteq \mathcal{Q}(\mathcal{L})$ (in which case we need random choices to choose answers from either of S_w or $\mathsf{S}_{w'}$). Note that the latter is guaranteed not to happen when conditioning on $\overline{\mathcal{B}}$. Now suppose we have sampled the same $(\mathcal{L}, C, \mathsf{S}_w, \mathsf{S}_{w'})$ in both experiments, and then we will see how the experiments continue in generating R_w and $\mathsf{R}_{w'}$. In Experiment Real, we sample the random oracle $\widetilde{\mathbf{f}} \xleftarrow{\$} (\mathbf{RO} \mid \mathcal{F})$ first, then execute the receiver's verification $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_w}(C, w, D_w)$, and then execute $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_{w'}}(C, w', D_{w'})$. By lazy evaluation in the sampling of $\widetilde{\mathbf{f}} \xleftarrow{\$} (\mathbf{RO} \mid \mathcal{F})$, it can be seen that the distribution of the view of $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_w}(C, w, D_w)$ in Experiment Real is exactly the same as the distribution of R_w in Experiment Imag. After this step, the value of $\mathsf{R}_{w'}$ (i.e., the view of $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_{w'}}(C, w', D_{w'})$) between the Experiments Real and Imag might deviate from each other *only* if $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_{w'}}(C, w', D_{w'})$ asks a query that is already answered in $R^{\widetilde{\mathbf{f}}}_{\mathbf{r}_w}(C, w, D_w)$ or is used in S_w. But, again this event does not happen if we condition on $\overline{\mathcal{B}}$. □

Now we bound the probability of the bad event $\mathcal{B}$.

Claim 4.9. $\Pr[\mathcal{B}] \le \delta + 2|\mathcal{W}|^2\delta + 2|\mathcal{W}|^2(m \cdot 2|\mathcal{W}|\delta)$ *in all experiments.*

Before proving Claim 4.9 we need to prove the following intuitive technical lemma.

Lemma 4.10. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_k$ be k random variables such that for every pair $\{i, j\} \subseteq [k]$, we have $\Delta(\mathbf{x}_i, \mathbf{x}_j) \le \delta$. Suppose $\mathbf{x} \xleftarrow{\$} \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be a random choice among them and let $\mathcal{E}_i$ be the event that $\mathbf{x}$ is selecting $\mathbf{x}_i$ (therefore $\Pr[\mathcal{E}_i] = 1/k$). Then with probability at least $1 - 2k^2\delta$ over the choice of $x \xleftarrow{\$} \mathbf{x}$, for* all *$i \in [k]$ it holds that $\Pr[\mathcal{B}_i \mid x] \ge \frac{1}{2k}$.*

Proof. Suppose on the contrary that with probability at least $2k^3\delta$ over $x \xleftarrow{\$} \mathbf{x}$, there exists an $i \in [k]$ such that $\Pr[\mathcal{B}_i \mid x] < \frac{1}{2k}$. Since for every choice of $x \xleftarrow{\$} \mathbf{x}$ there is some $j \in [k]$ such that $\Pr[\mathcal{B}_j \mid x] \ge \frac{1}{k}$, by the pigeonhole principle, there exists a fixed pair $(i, j) \in [k]^2$ such that with probability at least $2k^2\delta/k^2 = 2\delta$ over the choice of $x \xleftarrow{\$} \mathbf{x}$, it holds that $\Pr[\mathcal{B}_j \mid x] > 1/k$ and $\Pr[\mathcal{B}_i \mid x] < 1/(2k)$. Let $\mathcal{E}$ be the set of all such x. For every $x \in \mathcal{E}$ we have $\frac{1}{k} \leq \Pr[\mathcal{B}_j \mid x] = \frac{\Pr[\mathbf{x}_j = x]}{\sum_{\ell \in [k]} \Pr[\mathbf{x}_\ell = x]} = \frac{\Pr[\mathbf{x}_j = x]}{k \cdot \Pr[\mathbf{x} = x]}$. Therefore $\Pr[\mathbf{x}_j \in \mathcal{E}] \geq \Pr[\mathbf{x} \in \mathcal{E}] \geq 2\delta$. On the other hand, for every $x \in \mathcal{E}$, it holds that $\Pr[\mathbf{x}_j = x] \geq 2 \cdot \Pr[\mathbf{x}_i = x]$. Therefore we get that $\Pr[\mathbf{x}_i \in \mathcal{E}] \leq \delta$ which implies that $\Delta(\mathbf{x}_j, \mathbf{x}_i) \geq 2\delta - \delta = \delta$, but the latter is a contradiction. □

Proof of Claim 4.9. By the discussion in the proof of Claim 4.7 it should be clear that after getting $(C, \mathcal{L})$ in Real the pair $(\mathsf{S}_w, \mathsf{R}_w)$ is sampled from the distribution of the view of a full execution of the commitment scheme (i.e., the commitment phase followed by the decommitment phase) in the random oracle model *conditioned* on C being the commitment of w and $\mathcal{L}$ being part of the oracle. By the definition of the learning algorithm of Lemma 4.3, with probability at least $1 - \delta$, for every $q \notin \mathcal{L}$ it holds that $\Pr[q \in \mathcal{Q}(\mathsf{S}_{\mathbf{w}}) \cup \mathcal{Q}(\mathsf{R}_{\mathbf{w}}) \mid \mathcal{L}, C] \leq \varepsilon$ where $\mathbf{w}$ is the random message that the sender has used to generate the commitment C. In the following we assume that this is the case (and it will cost us an error of δ in bounding the probability $\Pr[\mathcal{B}]$). Also recall that we are assuming that for every pair $(w_0, w_1) \in \mathcal{W}^2$ it holds that $\Delta((\mathbf{X} \mid \mathbf{w} = w_0), (\mathbf{X} \mid \mathbf{w} = w_1)) \leq \delta$ (otherwise $\widehat{R}$ would have been δ-successful). Thus by Lemma 4.10, with probability at least $1 - 2|\mathcal{W}|^2\delta$ over the choice of $(C, \mathcal{L}) \xleftarrow{\$} \mathbf{X}$, for *every* $w \in \mathcal{W}$ it holds that $\Pr[w = \mathbf{w} \mid C, \mathcal{L}] \geq \frac{1}{2 \cdot |\mathcal{W}|}$. Again we assume that this is the case and it will cost us another additive error of $2|\mathcal{W}|^2\delta$ in bounding $\Pr[\mathcal{B}]$. Now for every $w \in \mathcal{W}$ it holds that $\Pr[q \in \mathcal{Q}(\mathsf{S}_w) \cup \mathcal{Q}(\mathsf{R}_w) \mid \mathcal{L}, C] \leq \frac{\delta}{1/(2|\mathcal{W}|)} < 2|\mathcal{W}|\delta$. Since in Experiment Imag_2 we can sample and fix $(\mathsf{S}_{w'}, \mathsf{R}_{w'})$ for any $w' \neq w$ before sampling $(\mathsf{S}_w, \mathsf{R}_w)$, and since $|\mathcal{Q}(\mathsf{S}_{w'}) \cup \mathcal{Q}(\mathsf{R}_{w'})| \leq m$, by a union bound the probability that at least of the queries in $\mathcal{Q}(\mathsf{S}_{w'}) \cup \mathcal{Q}(\mathsf{R}_{w'})$ is selected in $\mathcal{Q}(\mathsf{S}_w) \cup \mathcal{Q}(\mathsf{R}_w)$ is at most $m \cdot (2|\mathcal{W}|\delta)$. By a union bound over all pairs $w \neq w'$, we get that $\Pr[\mathcal{B}] \leq \delta + 2|\mathcal{W}|^2\delta + 2|\mathcal{W}|^2(m \cdot 2|\mathcal{W}|\delta)$. □

Putting Claims 4.7–4.9 together, we get that:

$$\Pr_{\mathbf{Out}_R}[\forall\, w \in \mathcal{W}, \mathsf{R}_w \text{ accepts}] \geq \Pr_{\mathbf{Out}_I}[\forall\, w \in \mathcal{W}, \mathsf{R}_w \text{ accepts}] - \Pr[\mathcal{B}] \geq 1 - \delta''$$

for $\delta'' = (m \cdot |\mathcal{W}|)^{O(1)} \cdot \delta^{\Omega(1)} + \mathrm{negl}(n)$. □

By using Claim 4.6 and an averaging argument, we conclude that for $\delta' = \sqrt{\delta''} \in (m \cdot |\mathcal{W}|)^{O(1)} \cdot \delta^{\Omega(1)} + \mathrm{negl}(n)$, with probability at least $1 - \delta'$ the sampled $(\mathcal{F}, C, \{D_w\})$ makes $\widehat{S}$ a $(1-\delta')$-successful cheating sender w.r.t. the randomized oracle $\widetilde{\mathbf{f}}$, and this finishes the proof of Lemma 4.2. In fact we only needed to show that such $(\mathcal{F}, C, \{D_w\})$ can be selected with *nonzero* probability, yet Claim 4.6 shows that this indeed happens with probability close to one.

□

5 Separation from Hitting One-Way Functions

Building upon the proof of Theorem 1.1, in this section we extend the black-box separation of non-interactive commitments to prove Theorem 1.3

5.1 Black-Box Constructions from Hitting One-Way Functions

In this subsection we provide formal definitions of black-box and non-black-box constructions of commitments using hitting one-way functions.

Definition 5.1. A black-box construction $\textsc{Com}$ of commitments from hitting one-way functions is defined similarly to Definition 3.9 with the difference that the security reductions H and B will *either* invert f over some security parameter $n' = n^{\Theta(1)}$ *or* output a circuit T of size $n'' = n^{\Theta(1)}$ and input density $d_T \geq 1/2$ which is not hit by f. More formally, suppose $J \in \{H, B\}$ is one of the security reductions and is given oracle access to some adversary $\textsc{Adv}$ (supposedly breaking the hiding or binding of $\textsc{Com}^f$ over the security parameter n) and is given some input $y = f(\mathbf{U}_{n'})$ to invert. We say that J "wins" if either of the following happens:

- **Refuting the one-way property of f:** J outputs some x' such that $f(x') = y$.
- **Refuting the hitting property of f:** J outputs some co-nondeterministic circuit T of size $n'' = n^{\Theta(1)}$, and input density $d_T \geq 1/2$ which is not hit by f.

The security reduction H (resp. B) will get oracle access to f and an adversary $\textsc{Adv}$ who with (non-negligible) probability ε breaks the hiding (resp. binding) of $\textsc{Com}^f$, gets as input a random $y \xleftarrow{\$} f(\mathbf{U}_{n'})$, and outputs some x together with some circuit T, and wins (as defined above) with probability at least $(\varepsilon/n)^{O(1)}$.

Demanding the witness that f is not hitting? Note that in the definition above, we did not require the security proof to also provide the witness that f is not hitting, in case it claims so (and we only require it to be true). This only makes our negative result stronger.

Why not f gates in T. We do *not* allow the circuit T output by the security reduction of Definition 5.1 to have f gates, even though proving an impossibility result that allows such f gates is a stronger statement. The reason is that we want the black-box construction to be secure if it uses any oracle f (of *unknown* running-time) that is one-way and hits all the circuits of polynomial size. But the size of a circuit with f gates is not well-defined (and will not be $\mathrm{poly}(n)$ if f is not polynomial-time computable). Thus given circuit T with f gates and having oracle access to f one can not verify whether T is hit by f or not. In fact, if one allows such gates in T (without counting the complexity of f in the running time of T) it is easy to see that the non-black-box construction of Theorem 1.2 can be made black-box in this model. Recall, however, that any black-box separation that is obtained though breaking the primitive in the presence of a random oracle still extends to this stronger regime, since a random oracle is a strong hitting set generator that even hits circuits with its own gates.

Definition 5.2. A *non-black-box* construction of non-interactive commitments from hitting one-way functions is defined similarly to Definition 5.1 with the difference that the parties (S, R) and the security reductions (H, B) are given the circuit of an efficiently computable f as input (rather than just an oracle access to it). We shall allow also all of these algorithms to run in time $\mathrm{poly}(n, t(n))$ where t is the circuit size of f since they are receiving a circuit of size $t(n)$ as part of their input.

We showed how to prove Theorem 1.2 in Section 2.2.1. In the rest of this section we prove Theorem 1.3.

5.2 Proof of Theorem 1.3

When the used primitive is a hitting one-way function f, instead of inverting the "one-way" function f (with the help of an adversary breaking the security of the commitment) the security reduction might simply output a (co-nondeterministic) circuit T [16]of size $s = n^{\Theta(1)}$ which is not hit by the oracle f used in the scheme.

Similarly to the case of separation from one-way functions, here we assume that a black-box implementation (S^f, R^f) of the black-box non-interactive commitment scheme Com from an oracle f exists, and then we will show that it can not be black-box secure. Note that Lemma 4.2 still holds since it did not depend on how the security of the construction (S, R) is proved. Again we will show that the existence of a $\delta = (\frac{1}{\text{poly}(n)})$-successful $\widehat{R}$ contradicts the existence of the security reduction H (that proves the hiding), and the existence of or a $\delta' = (\frac{1}{\text{poly}(n)})$-successful $\widehat{S}$ contradicts the existence of the security reduction B (that proves the binding). But as we mentioned in Section **??** we need to slightly modify the definition of the randomized oracle relative to which the cheating sender $\widehat{S}$ performs. The change was to sample the views of the sender $\mathsf{S}_0, \mathsf{S}_1$ *without* conditioning on the bit b to be zero or one. As it was discussed in Section **??**, the cheating sender $\widehat{S}$ can still succeeds with probability $\approx 1/4$.

We will start by the easier case that the malicious $\widehat{R}$ of Construction 4.4 succeeds.

Case 1: $\widehat{R}$ is successful. In this case we show that whenever $\widehat{R}$ (of Lemma 4.2) is δ-successful, if we use $\mathbf{f} = \mathbf{RO}$ in the scheme, the security reduction $H^{\mathbf{f}, \widehat{R}}$ can *not* output any circuit T of size $n^{\Theta(1)}$ (and input density $d_T > 1/2$) that is not hit by $\mathbf{f}$ (unless with negligible probability). Proving so shows that (if $\widehat{R}$ succeeds), the security reduction H might as well just try to invert $\mathbf{f}$ with a non-negligible probability. The latter would again lead to a contradiction by Lemma 3.13. The reason that $\mathbf{f}$ will hit all the circuits of size and input length $n^{\Theta(1)}$ with overwhelming probability is that here we are using the random oracle $\mathbf{f} = \mathbf{RO}$ and Lemma 2.7 shows that a random oracle hits all the circuits of size $n^{\Theta(1)}$ with overwhelming probability!

Case 2: $\widehat{S}$ is successful. To simplify the notation, in the following we will use n to denote the size of the circuit T output by the security reduction B that proves the binding (even though, in general this input length could be some $n'' = n^{\Theta(1)}$). We wish to show that again, the reduction B might as well simply try to invert f rather than trying to find a circuit T not hit by f (simply because such circuit won't exist). Proving so is harder in this case than the previous case that $\widehat{R}$ was δ-successful. The reason is that now $\widehat{S}$ does *not* perform w.r.t. a totally random oracle $\mathbf{RO}$ and is only successful w.r.t. a partially-fixed random oracle $\widetilde{\mathbf{f}}$ which is fixed over some part $\mathcal{F}$ of its domain, and the fixed part $\mathcal{F}$ might include all the first n^2 points in $\{0,1\}^n$ and prevent the function $f \xleftarrow{\$} \widetilde{\mathbf{f}}$ from hitting a particular circuit T with input length n. Despite that, a closer look at the distribution of $\mathcal{F}$ shows that the function $\widetilde{\mathbf{f}}$ is a "combination" of two partially-defined random functions (see Definition 5.5), because the marginal distribution of the query-answers in $(\mathcal{L}, \mathsf{S}_0)$ and $(\mathcal{L}, \mathsf{S}_1)$ are both sampled assuming that the scheme is in the random oracle model. So, intuitively, for every circuit T of size s, input length n, and input density $d_T > 1/2$ (i.e., $|\mathcal{S}_T| > 2^{n-1}$), it still holds that at least half of the values $f(1), \ldots f(n^2)$ are chosen at random, and thus one of them will hit $\mathcal{S}_T$ with probability at least $1 - 2^{-n^2/2 - n}$ (which is still sufficiently large to let us do a union bound over the number of circuits). However, we need to study carefully why this "partitioning" of the set $[n^2] \subseteq \{0,1\}^n$ into two parts is not going to be adversarially chosen against any particular input set $\mathcal{S}_T$. The following claim finishes the proof of Theorem 1.3.

[16]Here we use the name T for the circuit not to be confused with the commitment string.

Claim 5.3. *With probability at least $1-O(2^{-n})$, the oracle $\widetilde{\mathbf{f}}$ of Construction 4.5 hits all the circuits of size n.*

Recall that during the sampling of the oracle $\widetilde{\mathbf{f}}$ in Construction 4.5, we first sample $(C, \mathcal{L})$, then sample $(\mathsf{S}_0, \mathsf{S}_1)$, and then sample the rest of $\widetilde{\mathbf{f}}$ (while making random choices between the answer of S_0 and S_1 when they disagree on a query). Because of the way we do our samplings in Construction 4.5, the marginal distribution of query-answer pairs in $(\mathcal{L}, \mathsf{S}_0)$ is the "partially-defined" part of a random oracle (and the same holds for $(\mathcal{L}, \mathsf{S}_1)$).

We first formalize the notion of a partially-defined random oracle, and then will show that when one "combines" two partially random oracles and then "randomly extend" to the full domain, the result randomized function has a strong hitting property.

5.2.1 Partially-Defined Random Functions—Definitions

Definition 5.4 (Random Extensions)**.** Let $\mathcal{D}$ and $\mathcal{R}$ be arbitrary finite sets denoting a domain and a range and let $\mathbf{f}$ be a random variable whose values are partial functions from the domain $\mathcal{D}$ to the range $\mathcal{R}$. A random extension of $\mathbf{f}$ is a randomized *total* function $\widetilde{\mathbf{f}}$ distributed as follows:

1. First sample $f \xleftarrow{\$} \mathbf{f}$ (where f is defined only over $\mathcal{Q}(f) \subseteq \mathcal{D}$) and also define $\widetilde{\mathbf{f}}(x) = f(x)$ for every $x \in \mathcal{Q}(f)$.

2. Then for every $a \in \mathcal{D} \setminus \mathcal{Q}(f)$ choose a random answer $\widetilde{\mathbf{f}}(a) = b \xleftarrow{\$} \mathcal{R}$.

Definition 5.5 (Partially-Defined Random Functions)**.** Let $\mathcal{D}$ be a finite domain and $\mathcal{R}$ be a finite range. By the *random function* $\mathbf{U}(\mathcal{D}, \mathcal{R})$ from $\mathcal{D}$ to $\mathcal{R}$ we mean the random variable whose value is a random choice among all possible functions from $\mathcal{D}$ to $\mathcal{R}$. Let $\mathbf{f}$ be a random variable whose value is a *partial* function f defined over the domain set $\mathcal{Q}(f) \subseteq \mathcal{D}$. We call $\mathbf{f}$ a *partially-defined* random function (from $\mathcal{D}$ to $\mathcal{R}$) if and only if the random extension of f is identical to the random function from $\mathcal{D}$ to $\mathcal{R}$ (i.e., $\widetilde{\mathbf{f}} \equiv \mathbf{U}(\mathcal{D}, \mathcal{R})$).[17]

Definition 5.5 can be generalized to functions with a sequence of domains $\mathcal{D}_1, \mathcal{D}_2, \ldots$ and a sequence of ranges $\mathcal{R}_1, \mathcal{R}_2, \ldots$ with the restriction that $f(\mathcal{D}_n) \subseteq \mathcal{R}_n$ (e.g., by using $\mathcal{D}_n = \mathcal{R}_n = \{0,1\}^n$ we can consider length preserving functions and the random oracle $\mathbf{RO}$ as special cases). We will, however, only use the simpler definition above.

Definition 5.6 (Randomized Combination of Partial Functions)**.** For every two partial functions f_0 and f_1 we define a randomized procedure that *combines* them into a new randomized function $\mathbf{f}$, denoted by $\mathbf{f} \leftarrow \mathsf{Comb}(f_0, f_1)$, as follows:

- For every $a \in \mathcal{Q}(f_0) \setminus \mathcal{Q}(f_1)$ use $\mathbf{f}(a) = f_0(a)$.

- For every $x \in \mathcal{Q}(f_1) \setminus \mathcal{Q}(f_0)$ use $\mathbf{f}(a) = f_1(a)$.

- For every $a \in \mathcal{Q}(f_0) \setminus \mathcal{Q}(f_1)$ choose a random answer $\mathbf{f}(a) \xleftarrow{\$} \{f_0(a), f_1(a)\}$.

[17] Using the notation of Definition 5.5 the partially-fixed random oracle $\widetilde{\mathbf{f}}$ with a fixed part $\mathcal{F}$ can be thought of as $\widetilde{\mathbf{f}} \equiv \widetilde{\mathbf{F}}$ where $\mathbf{F}$ is a random variable whose value is fixed as $\mathbf{F} = \mathcal{F}$.

5.2.2 Proving Claim 5.3

Now we show how to prove Claim 5.3 which finishes the proof of Theorem 1.3.

Lemma 5.7. *Let A be a set of interactive algorithms with each algorithm described in A having their own private randomness. Suppose $A^{\mathbf{U}}$ is the system of oracle algorithms that interact with each other while they have access to the random oracle $\mathbf{U}\colon \mathcal{D} \mapsto \mathcal{R}$. Let $\mathbf{V}$ be the random variable that describes the view of all the parties in an execution of the system $A^{\mathbf{U}}$ where this view only includes their oracle queries $\mathcal{Q}(\mathbf{V})$ and their answers. It holds that $\mathbf{V}$ is a partially-defined random function (with domain $\mathcal{D}$ and range $\mathcal{R}$).*

Proof. We can choose the answers of the oracle $\mathbf{U}$ through the so called "lazy evaluation" method and choose its answers at random only when a query is asked. This way, the view $\mathbf{V}$ will include the sampled part of $\mathbf{U}$ by the end of the protocol, and we can sample the rest of $\mathbf{U}$ after sampling $\mathbf{V}$ first. But the latter sampling procedure is the same as sampling $\mathbf{U}$ directly (which is a uniformly random function from $\mathcal{D}$ to $\mathcal{R}$) by definition. $\square$

We emphasize that Lemma 5.7 does *not* extend (in general) to the case that $\mathbf{V}$ includes only a part of the views of the parties, because by knowing the partial view one might be able to conclude some information about the other oracle queries. Also recall that we are only interested in what happens over the sampled function $\widetilde{\mathbf{f}}$ for the domain $\{0,1\}^n$ since we assumed in the beginning that n is going to be the size of the circuit T output by the security reduction B proving the binding (and thus the input lengths of f other than n are irrelevant for that purpose).

Claim 5.8. *Both of $(\mathcal{L}, \mathsf{S}_0)$ and $(\mathcal{L}, \mathsf{S}_1)$ when restricted to the range and domain $\{0,1\}^n$ are partially-defined random functions (see Definition 5.5) with range and domain $\{0,1\}^n$.*

Proof. To show that $(\mathcal{L}, \mathsf{S}_0)$ is a partially-defined random function we employ Lemma 5.7 as follows. Consider a system in which there is only a sender S who generates the commitment C based on the bit $\mathbf{b} = 0$ and another party who receives C and learns the set $\mathcal{L}$ (according to the algorithm of Construction 4.4). This way, the distribution of $(\mathcal{L}, \mathsf{S}_0)$ is the same as $\mathbf{V}$ of Lemma 5.7, and so is a partially-defined random function. A similar argument holds for $(\mathcal{L}, \mathsf{S}_1)$. Note that even though the parties are allowed to ask oracle queries of length other than n we can "restrict" our attention only to $\{0,1\}^n$ and other queries asked to not harm the analysis of the distribution of the query-answer pairs over $\{0,1\}^n$. $\square$

For simplicity, in the following we will assume that the query-answer pairs appearing in $(\mathcal{L}, \mathsf{S}_0)$ and $(\mathcal{L}, \mathsf{S}_1)$ are all of length n (even though this is not the case, the other input-output queries are relevant to our claims). The following lemma can be easily verified by inspection.

Lemma 5.9 (Projecting Partially-Defined Random Functions)**.** *Suppose $\mathcal{S} \subseteq \mathcal{D}$, and let $\mathcal{R} = \bigcup_{i \in [k]} \mathcal{R}_i$ be a partition of $\mathcal{R}$ such that $|\mathcal{R}_i| = \frac{|\mathcal{R}|}{k}$ for every $i \in [k]$. Let $\mathbf{f}$ be a partially-defined random function from the domain $\mathcal{D}$ to the range $\mathcal{R}$. Then the random variable $\mathbf{g}$ defined as follows is a partially-defined random random variable with domain $\mathcal{S}$ and range $[k]$. To sample from $\mathbf{g}$ first sample $f \xleftarrow{\$} \mathbf{f}$, let $g(a) = j$ iff $a \in \mathcal{S}$ and $f(a) \in \mathcal{R}_i$.*

Let $\mathbf{f}_0$ be the partial (randomized) function defined by $(\mathcal{L}, \mathsf{S}_0)$ and $\mathbf{f}_1$ be that of $(\mathcal{L}, \mathsf{S}_1)$. Let $\mathbf{f} \leftarrow \mathsf{Comb}(f_0, f_1)$ be the randomized combination of f_0 and f_1. It is easy to see that $\widetilde{\mathbf{f}}$ as defined in Construction 4.5 is the same as the random extension of $\mathbf{f}$ which we (intentionally) also chose to denote as $\widetilde{\mathbf{f}}$. Here we are interested in the behavior of $\widetilde{\mathbf{f}}$ over the $[n^2] \subseteq \{0,1\}^n$ and would like to see if there is any $x \in [n^2]$ such that $\widetilde{\mathbf{f}}(x) \in \mathcal{S}_T$. If there is any such mapping, then $\widetilde{\mathbf{f}}$ hits T. Fix any (co-nondeterministic) circuit T of size n with a corresponding input set $\mathcal{S}_T$ of density at least $1/2$ (which we can remove elements from $\mathcal{S}_T$ to make its density equal to $1/2$). For $i \in \{0,1\}$, let $\mathbf{g}_i$ be the randomized boolean function defined defined only over the domain $[n^2]$ according to: $\mathbf{g}_i(x) = 1$ iff $\mathbf{f}_i(x) \in \mathcal{S}_T$. By Lemma 5.9 $\mathbf{g}_0$ and $\mathbf{g}_1$ are partially-defined Boolean random functions defined over $[n^2]$. It is also easy to see that $\mathbf{g} \leftarrow \mathsf{Comb}(\mathbf{g}_0, \mathbf{g}_1)$ is the projection of $\mathbf{f} \leftarrow \mathsf{Comb}(\mathbf{f}_0, \mathbf{f}_1)$ and that the random extension $\widetilde{\mathbf{g}}$ of $\mathbf{g}$ is identical to the projection of the random extension $\widetilde{\mathbf{f}}$ of $\mathbf{f}$ to the domain $[n^2]$ and range $\{0,1\}$. Claim 5.3 follows from the following claim (whose proof appears in the following section).

Claim 5.10. *For $i \in \{0,1\}$, let $\mathbf{g}_i$ be the randomized boolean function defined defined only over the domain $[n^2]$. Let $\mathbf{g} \leftarrow \mathsf{Comb}(\mathbf{g}_0, \mathbf{g}_1)$ be their randomized combination and let $\widetilde{\mathbf{g}}$ be the (Boolean) random extension of $\mathbf{g}$ to the domain $[n^2]$. Then with probability at least $1 - O(2^{-2n})$ over the choice of $\widehat{g} \xleftarrow{\$} \widetilde{\mathbf{g}}$ it holds that $\sum_{x \in [n^2]} \widehat{g}(x) > 0$.*

Lemma 5.10 shows that the probability that $\widetilde{\mathbf{g}}$ does *not* hit a fixed circuit T of size s is at most $O(2^{-2s})$ and therefore by a union bound, with overwhelming probability $1 - O(2^s \cdot 2^{-2s})$, $\widetilde{\mathbf{g}}$ hits all the circuits of size $s = n^{\Theta(1)}$. This finishes the proof of Claim 5.3 and Theorem 1.3.

5.2.3 Proving Lemma 5.10—Concentrations Bounds for Partially-Defined Random Functions

In this section we prove Lemma 5.10. For that purpose we need to develop some concentration bounds for partially-defined random functions.

Lemma 5.11 (Restating Lemma ??)**.** *Let $p_\delta(k)$ denote the probability that k independent unbiased Boolean random variables have summation at most $(1/2 - \delta) \cdot k$. Also let $\mathbf{g}$ be a partially-defined random function with domain $\mathcal{D} = [m]$ and range $\mathcal{R} = \{0,1\}$. Then for every $k \in [m]$ and $0 \le \delta \le 1/2$ it holds that*

$$\Pr_{g \xleftarrow{\$} \mathbf{g}} \left[|\mathcal{Q}(g)| \ge k \text{ and } \sum_{x \in \mathcal{Q}(g)} g(x) \le (\tfrac{1}{2} - \delta) \cdot k \right] \le \frac{p_\delta(m)}{p_\delta(m-k)}.$$

Proof. Let $\widetilde{\mathbf{g}}$ be the random extension of $\mathbf{g}$ to the whole domain $[m]$. We suppose on the contrary that when we sample $g \xleftarrow{\$} \mathbf{g}$, with probability more than $\frac{p_\delta(m)}{p_\delta(m-k)}$ it holds that $|\mathcal{Q}(g)| \ge k$ *and* $\sum_{x \in \mathcal{Q}(g)} g(x) \le (1/2 - \delta) \cdot k$. Now, after sampling $g \xleftarrow{\$} \mathbf{g}$, we also sample the rest of $\widetilde{\mathbf{g}}$ which involves sampling $m - |\mathcal{Q}(g)|$ more random unbiased Boolean random variables. Let $\bar{\mathbf{g}}$ be the partial function that we sample when extending $\mathbf{g}$ to $\widetilde{\mathbf{g}}$. Since $|\mathcal{Q}(\bar{\mathbf{g}})| \le m - k$, even condition on any fixed g such that $|\mathcal{Q}(g)| \ge k$, with probability *at least* $p_\delta(m-k)$ over the choice of $\bar{g} \xleftarrow{\$} \bar{\mathbf{g}}$, it holds that $\sum_{x \in \mathcal{Q}(\bar{g})} \bar{g}(x) \le (1/2 - \delta) \cdot (m-k)$. Therefore, with probability more than $\frac{p_\delta(m)}{p_\delta(m-k)} \cdot p_\delta(m-k) = p_\delta(m)$ over the choice of $\widetilde{g} \xleftarrow{\$} \widetilde{\mathbf{g}}$ it would hold that $\sum_{x \in \mathcal{Q}(\widetilde{g})} \widetilde{g}(x) \le (1/2 - \delta)k + (1/2 - \delta)(m-k) = (1/2 - \delta) \cdot m$ which contradict the definition of $p_\delta(m)$. □

Lemma 5.12 (Implied by Lemma A.2.2 in [AS08])**.** *Suppose $\mathbf{x}_1, \dots, \mathbf{x}_m$ are m independent unbiased Boolean random variables with summation $\mathbf{x} = \sum_i \mathbf{x}_i$ and let δ be such that $\omega(\sqrt{m}) \le \delta \cdot m \le o(m)$. Then it holds that $\Pr[\mathbf{X} > (1/2 + \delta)m] = \Pr[\mathbf{X} < (1/2 - \delta)m] = e^{-(2+o(1))\delta^2 m}$.*

The upper-bound of $\Pr[\mathbf{X} < (1/2 - \delta)m] < e^{-2\delta^2 m}$ follows by the Chernoff bound, and Lemma 5.12 specifies that for certain range of parameters there exists an anti-concentration bound showing that the Chernoff is almost tight.

Corollary 5.13 (Restating Theorem 2.13)**.** *let $\mathbf{g}$ be a partially-defined random function with domain $\mathcal{D} = [m]$ and range $\mathcal{R} = \{0, 1\}$. Then for every $k \in [m]$ and $\omega(\sqrt{m}) < \delta \cdot m < o(m)$ it holds that*

$$\Pr_{g \xleftarrow{\$} \mathbf{g}} \left[|\mathcal{Q}(g)| \geq k \text{ and } \sum_{x \in \mathcal{Q}(g)} g(x) \leq (\tfrac{1}{2} - \delta) \cdot k\right] \leq e^{-(2+o(1))\delta^2 k}.$$

Proof. By Lemma 5.11 we get the upper-bound of $\frac{p_\delta(m)}{p_\delta(m-k)}$. By Lemma 5.12 it holds that $p_\delta(t) = e^{-(2+o(1))\delta^2 t}$, and thus we get the upper-bound of

$$\frac{p_\delta(m)}{p_\delta(m-k)} = \frac{e^{-(2+o(1))\delta^2 m}}{e^{-(2+o(1))\delta^2 (m-k)}} = e^{-(2+o(1))\delta^2 k}.$$

$\square$

As is clear from the proof of Corollary 5.13, any anti-concentration bound that lower-bounds $p_\delta(m-k)$ for an arbitrary δ (out of the range specified in Lemma 5.11) leads to some upper-bound over $\Pr_{g \xleftarrow{\$} \mathbf{g}}[|\mathcal{Q}(g)| \geq k \text{ and } \sum_{x \in \mathcal{Q}(g)} g(x) \leq (\frac{1}{2} - \delta) \cdot k]$.

In the following lemma one can use the domain size to be as small as $\omega(s)$, but we will prove it only for the more relaxed case of $[n^2]$ which is sufficient for us.

Lemma 5.14. *Let $\mathbf{g}_0$ and $\mathbf{g}_1$ be two (possibly correlated) partially-defined random functions with domain $[n^2]$ and range $\{0, 1\}$, and let $\mathbf{g} \leftarrow \mathsf{Comb}(\mathbf{g}_0, \mathbf{g}_1)$ be their (randomized) combination. Suppose that for all instances $g_0 \xleftarrow{\$} \mathbf{g}_0$ and $g_1 \xleftarrow{\$} \mathbf{g}_1$ it holds that $\mathcal{Q}(g_0) \cup \mathcal{Q}(g_1) = [n^2]$ (i.e., the combination $\mathbf{g}$ is always a total function). Then it holds that $\Pr_{g \xleftarrow{\$} \mathbf{g}}[\sum_{x \in [n^2]} g(x) = 0] < 2^{-2n}$.*

Concluding Lemma 5.10. Before proving Lemma 5.14 we show how to conclude Lemma 5.10 from Lemma 5.14. The difference between the two lemmas is that in Lemma 5.14 the combination of two functions $\mathbf{g}_0$ and $\mathbf{g}_1$ is a total function whereas in Lemma 5.10 we need to take a random extension at the end to make the function total. In Lemma 5.10 let $\mathbf{g}'_0$ be a "partial" random extension of $\mathbf{g}_0$ as follows. The partial function $\mathbf{g}'_0$ is a sub-function of $\widehat{\mathbf{g}}$ (where $\widehat{\mathbf{g}}$ is the random extension of the combinations of $\mathbf{g}_0$ and $\mathbf{g}_1$) which does *not* include $\mathcal{Q}(\mathbf{g}_1) \setminus \mathcal{Q}(\mathbf{g}_0)$. Namely, $\mathbf{g}'_0$ is the maximal extension of $\mathbf{g}_0$ that is consistent with $\widehat{g}$ but does not intersect with the queries whose answers are determined by $\mathbf{g}_1$ (alone). Similarly define $\mathbf{g}'_1$ based on $\mathbf{g}_0$, $\mathbf{g}_1$ and $\widetilde{\mathbf{g}}$. It is easy to see that **(1)** both of $\mathbf{g}'_0$ and $\mathbf{g}'_0$ are partially-defined random oracles, and **(2)** $\mathcal{Q}(\mathbf{g}'_0) \cup \mathcal{Q}(\mathbf{g}'_1) = [n^2]$, and **(3)** the combination of $\mathbf{g}'_0$ and $\mathbf{g}'_0$ is identically distributed as the random extension of the combination of $\mathbf{g}_0$ and $\mathbf{g}_1$. Therefore Lemma 5.10 implies Lemma 5.10.

Proof of Lemma 5.14. Let $\mathcal{B}$ be the event that $\sum_x g(x) = 0$, and for $i \in \{0, 1\}$ let $\mathcal{E}_i$ be the event that $\sum_{x \in \mathcal{Q}(g_i)} g_i(x) \geq n^2/3$.

First we note that $\Pr[\mathcal{B} \mid \sum_{x \in \mathcal{Q}(g_0)} g_0(x) > t] < 2^{-t}$. That is because, whenever there are t samples $\{x_1, \dots, x_t\}$ in g_0 that are mapped to 1, then in order to get $\sum_x g(x) = 0$, for all $i \in [t]$ we shall have: **(1)** $x_i \in \mathcal{Q}(g_1)$ and **(2)** $g_1(x_i) = 0$ and **(3)** choose $g(x_1) = g_1(x_1)$ when combining g_0 and g_1. For each $i \in [i]$, we will choose $g(x_i) = g_1(x_i)$ only with probability $1/2$, and so we will choose $g(x_i) = g_1(x_i)$ for *all* $i \in [t]$ only with probability at most 2^{-t}. A similar argument shows that $\Pr[\mathcal{B} \mid \sum_{x \in \mathcal{Q}(\mathbf{g}_1)} \mathbf{g}_1(x) > t] < 2^{-t}$, and therefore $\Pr[\mathcal{B} \mid \mathcal{E}_0 \vee \mathcal{E}_1] \le 2^{1-n^2/3}$.

In the following we will show that $\Pr[\overline{\mathcal{E}_0 \vee \mathcal{E}_1}] \le 2^{-(1+o(1))n^{3/2}}$, which will imply that

$$\Pr[\mathcal{B}] \le \Pr[\overline{\mathcal{E}_0 \vee \mathcal{E}_1}] + \Pr[\mathcal{B} \mid \mathcal{E}_0 \vee \mathcal{E}_1] \le 2^{-(1+o(1))n^{3/2}} + 2^{1-n^2/3} = 2^{-(1+o(1))n^{3/2}}.$$

By using the parameters $m = n^2$, $k = n^2/2$ and $\delta = n^{-1/10}$ in Corollary 5.13 (which satisfy the condition $\omega(n) < \delta \cdot n^2 < o(n^2)$) for $i \in \{0,1\}$ we get that

$$\Pr_{g_i \xleftarrow{\$} \mathbf{g}_i} [|Q(g_i)| \ge n^2/2 \text{ and } \sum_{x \in \mathcal{Q}(g_i)} g_i(x) \le (\tfrac{1}{2} - n^{-1/10}) \cdot n^2] \le e^{-(2+o(1))\delta^2 n^2} < 2^{-(1+o(1))n^{3/2}}.$$

On the other hand since $\mathcal{Q}(g) = [n^2]$, we know that $\Pr[|Q(g_0)| \ge n^2/2 \text{ or } |Q(g_1)| \ge n^2/2] = 1$, and therefore $\Pr[\forall\ i \in \{0,1\}, \sum_{x \in \mathcal{Q}(g_i)} g_i(x) < (\frac{1}{2} - n^{-1/10}) \cdot n^2] < 2 \cdot e^{-(2+o(1))\delta^2 n^2} < 2^{-(1+o(1))n^{3/2}}$. Finally, since $(\frac{1}{2} - n^{-1/10}) \cdot n^2 > n^2/3$, we get that $\Pr[\overline{\mathcal{E}_0} \wedge \overline{\mathcal{E}_1}] \le 2^{-(1+o(1))n^{3/2}}$. □

6 The Checkability Barrier

In this section we prove Theorems **??** and 1.4.

6.1 Proof Systems and Program Checkers

Definition 6.1 (Interactive Proofs [GMR89])**.** A proof system (P, V) for a language L is a pair of interactive algorithms such that V runs in time $\mathrm{poly}(|x|)$ where x is the common input, and the following holds:

- c**-Completeness:** V accepts the interaction with P over any $x \in L$ with probability $\ge c(|x|)$.
- $(1-s)$**-Soundness:** No matter what strategy a cheating prover employs, the verifier accepts the interaction over any $x \notin L$ with probability at most $s(|x|)$ (which is called the soundness error).
- **Non-negligible Gap:** It holds that $c(n) - s(n) > 1/\mathrm{poly}(n)$.

An *argument* system is defined similarly, but the soundness is guaranteed only against $\mathrm{poly}(n)$-sized circuits cheating provers.[18]

Definition 6.2 (Honest-Verifier Zero-Knowledge)**.** Let $\mathsf{View}\langle P, V\rangle(x)$ be the view of a verifier V in an interaction interaction with a prover P over the input x. A proof (or argument) system (P, V) for a language L is called honest-verifier zero-knowledge HVZK, if there exists an efficient simulator $\textsc{Sim}$ such that the ensembles $\{\textsc{Sim}(x)\}_{x \in L}$ and $\{\mathsf{View}\langle P, V\rangle(x)\}_{x \in L}$ are computationally indistinguishable.

[18] A k-prover proof system is defined similarly with the restriction that the provers can not communicate with each other during the interaction (and only talk to the verifier). The completeness is defined the same as before while the soundness should only hold when considering cheating prover strategies that do not communicate during the interaction with the verifier.

Definition 6.3 (Checkability)**.** A language L is (black-box) checkable if there exists an efficient algorithm A (called the program checker) such that given any oracle π, the following holds.

- **Completeness:** Whenever $\pi(x) = L(x)$ for every x, then for every x it also holds that $\Pr[A^\pi(x) = L(x)] = 1 - \mathrm{negl}(n)$.
- **Soundness:** For every x (regardless of whether π solves L always correctly or not), it holds that $\Pr[A^\pi(x) \in \{L(x), \perp\}] = 1 - \mathrm{negl}(n)$. ($\perp$ denotes "finding a bug" in the "program" π.)

Lemma 6.4 ([BK95])**.** *If there are a (single prover) proof system for both of the languages L and $\overline{L}$ in which the provers can be implemented efficiently given access to an L-oracle (i.e., implemented in $\mathsf{BPP}^{\mathsf{NP}}$), then L has a black-box program checker.*[19]

Remark 6.5. Since all languages in NP are trivially provable with a prover of complexity P^{NP}, the (black-box) checkability of NP is equivalent to the existence of a proof system for coNP with provers in $\mathsf{BPP}^{\mathsf{NP}}$.[20]

6.2 Lower-Bounds on Instance-Based Commitments

In this section we prove Theorem **??**.

In fact we prove something stronger than Theorem **??**:

Theorem 6.6. *If there exists a construction of instance-based non-interactive commitments for the language L through a black-box construction based on one-way functions, then there exists a* single prover *proof system for $\overline{L}$ whose prover complexity is in $\mathsf{BPP}^{\mathsf{NP}}$.*

For the case of $L = \mathrm{SAT}$, the theorem above implies a proof system for coNP with prover complexity $\mathsf{BPP}^{\mathsf{NP}}$. By Lemma 6.4 and Remark 6.5, the latter implies the checkability of SAT (and all of NP).

We will prove Theorem 6.6 for the case of one-way functions, and the generalization to FCRHs is straightforward. In the following we will assume that a black-box construction of instance-based commitments based on one-way function f exists, and we feed the construction with a *random* oracle $f \equiv \mathbf{RO}$.

The formal description of the protocol to prove $\overline{L}$ is as follows.

Construction 6.7. This protocol is based on a black-box construction (S, R) of non-interactive commitments for $\mathcal{W} = \{0, 1\}$ from one-way functions. For this assumed construction, let $\delta = 1/\mathrm{poly}(n)$ be chosen small enough so that $\delta' < 1/2$ in Lemma 4.2. The prover P and the verifier V both get access to x (which P claims to be $x \notin L$). The length of the input $|x| = n$ serves as the security parameter (i.e., both parties run in $\mathrm{poly}(n)$ time). The prover has access to an NP oracle and its goal is to prove that $x \notin L$.

1. The verifier V chooses a random seed $\mathbf{r}_S$ and a random bit $b \xleftarrow{\$} \{0, 1\}$. Then it executes the sender's algorithm S (of the commitment scheme) to generates the commitment string $C(b)$. During this execution the verifier chooses the answers to the oracle queries of the sender S at random (and saves the answers to use them in case of asking the same query again). The verifier sends $C(b)$ to the prover.

[19] The statement would be "if and only if" in case of using a k-prover proof system for any $k \geq 2$.

[20] The existence of a proof system for coNP with a *single* prover of complexity $\mathsf{BPP}^{\mathsf{NP}}$ is potentially stronger than just the checkability of NP (since the checkability is equivalent to the existence of *multi*-prover proof systems).

2. Then the parties engage in $10m/\delta^2$ rounds of interaction. In each round the prover sends an oracle query q to the verifier. The verifier looks up the query q and if the answer $f(q)$ is already chosen, it sends the answer to the prover. In case $f(q)$ is not chosen yet, the verifier V chooses $f(q) \xleftarrow{\$} \{0,1\}^{|q|}$ at random and returns the answer to the prover. The way the prover chooses his queries is by executing the cheating receiver algorithm $\widehat{R}$ of Lemma 4.2 with the parameter δ (and prover's NP oracle is used to execute the learning algorithm efficiently). Note that, the learning algorithm of $\widehat{R}$ will ask at most $10m/\delta^2$ oracle queries. Thus there is enough number of rounds so that the prover can ask its queries from the verifier.

3. In the last round of the protocol, the prover sends his guess about the bit b by outputting the bit which is more likely to be used by the sender conditioned on $(C(b), \mathcal{L})$.

4. The verifier accepts if and only if the prover's last message is equal to the bit she used in the commitment.

Claim 6.8. *Suppose Construction 6.7 uses a black-box construction of instance based non-interactive commitment scheme (S, R) for the language L based on one-way functions with a black-box proof of security, then:*

- **Completeness:** *If $x \in \overline{L}$, then the verifier accepts with probability at least $(1+\delta)/2$.*
- **Soundness:** *If $x \notin \overline{L}$, then no matter what an unbounded cheating prover $\widehat{P}$ does, it can not make the verifier accept with probability more than $1/2 + \mathrm{negl}(n)$.*

Proof of Claim 6.8.

Soundness. This property follows from the black-box proof of hiding for the commitment scheme (in case $x \in L$) and Lemma 3.13. Note that the prover has no way to ask more than $10m/\delta^2 \leq \mathrm{poly}(n)$ oracle queries from the oracle f, simply because it is the verifier who is simulating f and answers only $10m/\delta^2$ many queries in $10m/\delta^2$ many rounds. By Lemma 3.13 no cheating receiver who asks up to $\mathrm{poly}(n)$ oracle queries is able to guess the committed bit by more than $1/2+\mathrm{negl}(n)$ (otherwise the black-box proof of hiding cannot exist).

Completeness. Similarly to the case of soundness, but by this time by by the black-box proof of security for the *binding* property of the commitment scheme, and due to Lemma 3.13, we conclude that there is no (efficient query) cheating sender $\widehat{S}$ (together with a partially-fixed random oracle fixed over a $\mathrm{poly}(n)$-sized domain) who is $1/\mathrm{poly}(n)$-successful according to the definition of Lemma 4.2. But this is exactly what we want here, because Lemma 4.2 implies that either such a $1/\mathrm{poly}(n)$-successful cheating sender exists, or that $\widehat{R}$ will be a δ-successful cheating receiver who is able to δ-distinguish between the commitments 0 and 1. But the black-box proof of security for binding asserts that such $\widehat{S}$ can not exist, therefore it is the δ-successful $\widehat{R}$ which exists. In particular, the prover can use this successful cheating receiver's strategy $\widehat{R}$ to guess the random bit b correctly with probability at least $(1+\delta)/2$. Also note that the prover has enough number of rounds to ask all of its oracle queries (to emulate $\widehat{R}$) from the verifier who controls the access to the oracle f. $\square$

6.3 Lower-Bounds on Honest-Verifier Zero-Knowledge

In the rest of this section we prove Theorem 1.4.

Definition 6.9 (k-Bit Verifiers)**.** In the following by a "k-bit verifier" we denote a verifier V in a 3-message public-coin protocol who sends k random bits in the second message of the protocol. The verifier V is also allowed to toss one more round of coins after receiving the second message of the prover and use them in her final decision.

We first prove Theorem 1.4 for the easier case of 1-bit verifiers. This simple case, even without the ending coin tosses by the verifier, includes protocols such as Blum's zero-knowledge protocol for Hamiltonicity of graphs [Blu87] as special case. After that we show how to extend the proof to the more general case of $O(\log n)$-bit verifiers which includes the zero-knowledge protocol of [GMW87] for 3-coloring of graphs as special case. In both cases we essentially reduce the problem to the case of instance-based commitments which is already handled by Theorem **??**. Our reduction, however, starting from a zero-knowledge protocol, constructs a *weakly* binding scheme (in which the scheme is only $(1/\operatorname{poly}(n), 1/\operatorname{poly}(n))$-binding, but the proof of Theorem **??** in fact handles the weakly-binding case directly (because the cheating sender $\widehat{S}$ succeeds with probability $1-\delta'$ which can be chosen to be $1-\delta' > 1-1/\operatorname{poly}(n)$).

6.3.1 1-bit Verifiers

Here we describe reduction due to [KMS07] from instance-based non-interactive bit-commitment schemes for the language L to any 3-message public-coin honest-verifier zero-knowledge argument system for L with a 1-bit verifier as defined in Definition 6.9. This would prove Theorem 1.4 for the case of 1-bit verifiers, since this new black-box construction for commitment can be used to get a program checker for NP by Construction 6.7. We present a more general construction starting from $\operatorname{poly}(n)$-bit verifiers, and prove its properties formally later because the more general construction will be used in the proof of $\log n$-bit verifiers as well.

Construction 6.10 (Commitment from k-Bit Verifiers)**.** Let (P, V) be zero-knowledge argument system for the language L with a k-bit verifier V (as defined in Definition 6.9) and simulator $\textsc{Sim}$. A non-interactive instance-based commitment (S, R) for the same language L can be constructed as follows: (the construction might *not* be secure in general).

- **Commitment:** Suppose $x \in L$ and $w \in [2^k]$ is the sender's private input. The sender S runs the simulator over the input x to get $(a_1, v, a_2, r) \leftarrow \textsc{Sim}(x)$ where (a_1, a_2) are the simulated prover messages, v is the verifier's k-bit message, and r is the verifier's final coin tosses. The sender S sends the commitment $C(w) = (a_1, v + w = v')$ to the receiver.

- **Decommitment:** The sender sends (b, a_2) as the decommitment value. The receiver chooses r' at random and runs the verifier over the transcript $(a_1, v' + w, a_2, r')$ and rejects the decommitment if this verification fails.

The following lemma specifies the properties of of Construction 6.10 when the verifier is 1-bit, and might be of independent interest. The work of Ong and Vadhan [OV07] has already proved this lemma without the non-interactive feature.

Lemma 6.11 (Bit-Commitment from 1-Bit Verifiers)**.** *If one uses an argument system (P, V) with completeness $1 - \mathrm{negl}(n)$, soundness δ, and a 1-bit verifier in Construction 6.10 (i.e., $|v| = k = 1$), then the result will be a non-interactive $\sqrt{\delta}$-binding bit-commitment scheme.*

We postpone the proof of Lemma 6.11 to the proof of Lemma 6.13 which includes Lemma 6.11 as a special case.

6.3.2 $O(\log n)$-Bit Verifiers

Now we go over the general case of $O(\log n)$-bit verifiers. Unfortunately, we do not know how to construct standard commitment schemes from 3-message zero-knowledge protocols with k-bit verifiers for $k > 1$), so we will take another tour. We will define a new primitive, called a "somewhere-binding" commitment: a commitment scheme that the sender is *not* able to decommit to *all* the possible values. We show that 3-message zero-knowledge protocols with a k-verifier will imply a somewhere-binding commitment scheme with message space of size 2^k. We then show how to extend Theorem **??** to somewhere-binding commitments schemes of message space $|\mathcal{W}| = \mathrm{poly}(n)$.

Definition 6.12 (Somewhere-Binding Commitments)**.** A *somewhere-binding* commitment scheme for message space $\mathcal{W} = \mathcal{W}_n$ (where n is the security parameter) is a two party protocol between a sender S and a receiver R defined similarly to Definition 3.3 with the following difference:

- **Sender's Input:** The sender receives a private input vector $w \in \mathcal{W}_n$.
- **α-Binding:** For every malicious efficient sender $\widehat{S}$ who plays the role of S in the commitment phase, receives $w \in \mathcal{W}$, and outputs a decommitment D_w, with probability at least α over the choice of C, there exists at least one value $w \in \{0, 1\}$ such that $\Pr[R(C, w, D_w) \text{ accepts}] \le 1 - \alpha$ where the probability is over the randomness of the verification $\mathbf{r}_V$ and the remaining randomness of $\widehat{S}$ in generating D_w based on w. We simply call the (somewhere-binding) commitment scheme binding if it is α-binding for $\alpha = 1 - \mathrm{negl}(n)$, and call it weakly-binding if it is α-binding for $\alpha = 1/\mathrm{poly}(n)$.

Note that for the case of $\mathcal{W} = \{0, 1\}$, the somewhere-binding and regular commitments become the same objects. The following lemma shows that if we feed an argument system with a k-bit verifier to Construction 6.10, it gives us a somewhere-binding commitment for message space $[2^k]$.

Lemma 6.13 (Somewhere-Binding Commitment from k-Bit Verifiers)**.** *If one uses an argument system (P, V) with completeness $1 - \mathrm{negl}(n)$, soundness δ, and a k-bit verifier in Construction 6.10 (i.e., $|v| = k$), then the result will be a non-interactive $\sqrt{\delta}$-binding somewhere-binding commitment scheme for message space $\mathcal{W} = [2^k]$.*

Proof of Lemma 6.13.

Completeness. The completeness of the commitment scheme (S, R) is inherited from that of the proof system (P, V) and the quality of its simulator Sim. More formally, we define the following random variables.

- $\mathbf{T}_1$: denoting the transcript (a_1, v, a_2, r) of an actual execution of (P, V) over x.
- $\mathbf{T}_2$: (a_1, v, a_2, r') where the last component of $\mathbf{T}_1$ is substituted with a fresh randomness.

- $\mathbf{T}_3$: denoting the output of the simulator $(a_1, v, a_2, r) \leftarrow \textsc{Sim}(x)$.
- $\mathbf{T}_4$: (a_1, v, a_2, r') where the last component of $\mathbf{T}_3$ is substituted with a fresh randomness.

By the completeness of the argument system $\Pr[V(\mathbf{T}_1) = \text{ accept}] = 1 - \text{negl}(n)$. In the following that $\mathbf{T}_1$ and $\mathbf{T}_4$ are computationally indistinguishable $\mathbf{T}_1 \approx_c \mathbf{T}_4$ which will show that $\Pr[V(\mathbf{T}_4) = \text{ accept}] = 1 - \text{negl}(n)$ as well, proving the completeness of the commitment scheme. The reason is that by the quality of the simulation we have $\mathbf{T}_1 \approx_c \mathbf{T}_3$, and so if we substitute the last message of $\mathbf{T}_1$ and $\mathbf{T}_3$ with a fresh randomness they remain indistinguishable $\mathbf{T}_2 \approx_c \mathbf{T}_4$ (because it is an efficient transformation). But $\mathbf{T}_1$ and $\mathbf{T}_2$ are simply the same distributions, and thus $\mathbf{T}_1 \equiv \mathbf{T}_2 \approx_c \mathbf{T}_4$.

Hiding. The hiding property of the commitment scheme relies on the quality of the simulation and the randomness of the verifier's second message. We will show that commitments to every two messages are computationally indistinguishable. We will prove it only for messages 0^k and 1^k, but the same arguments for every two messages $w, w' \in [2^k]$. Now we consider the following random variables:

- $\mathbf{C}_1$: denoting the partial transcript (a_1, v) of an actual execution of (P, V) over x.
- $\mathbf{C}_2$: $(a_1, v + 1^k)$ where the second component of $\mathbf{C}_1$ is flipped.
- $\mathbf{C}_3$: denoting the first two messages (a_1, v) simulated by the simulator $\textsc{Sim}(x)$.
- $\mathbf{C}_4$: $(a_1, v + 1^k)$ where the second component of $\mathbf{C}_3$ is flipped.

We have that **(1)** $\mathbf{C}_1 \equiv \mathbf{C}_2$ because v is a random message and that **(2)** $\mathbf{C}_1 \approx_c \mathbf{C}_3, \mathbf{C}_2 \approx_c \mathbf{C}_4$ both due to the quality of the simulator. Thus we get $\mathbf{C}_3 \approx_c \mathbf{C}_1 \equiv \mathbf{C}_2 \approx_c \mathbf{C}_4$ proving the hiding.

Binding. The binding property follows from the soundness of the proof system (P, V) and the quality of the simulation. More formally let $\widehat{S}$ be a cheating sender that with probability at least $1 - \sqrt{\delta}$ can generate a commitment $C = (a_1, v')$ such that for every $w \in [k]$, it can generate D_w such that with probability more than $\sqrt{\delta}$, (C, w, D_w) passes the verification of the receiver. Then we show a closely related $\widehat{P}$ that is able to convince the verifier (at least) with probability δ about the claim $x \in L$ (which is not possible if $x \notin L$). The cheating prover $\widehat{P}$ simply runs $\widehat{S}$ to get the commitment $C = (a_1, v')$ and sends a_1 as the first message. Then given the verifier's message v, the cheating prover asks $\widehat{R}$ to generate the decommitment D_w for $w = v + v'$, and sends the second message $a_2 = D_w$. Note that if the verifier accepts the decommitment $(a_1, v' + w, D_w = a_2)$ for $w = v + v'$, it is in fact accepting the transcript (a_1, v, a_2). $\square$

In the following we show how to extend Theorem **??** to the case of somewhere-binding commitments of message length $O(\log n)$.

Theorem 6.14. *If there exists a black-box construction of instance-based non-interactive somewhere-binding commitment for an* NP*-complete language and message space* $\mathcal{W}$ *of size* $|\mathcal{W}| = \text{poly}(n)$ *from one-way functions then* NP *is checkable.*

Proof. We employ a similar approach to the proof Theorem **??** by giving a proof system for the language L assuming the black-box somewhere-binding commitment for the message space $|\mathcal{W}| = \mathrm{poly}(n)$.

This time we will use Lemma 4.2 in its full-fledged proven form in a slightly modified version of Construction 6.7. This time, whenever the prover clams $x \notin L$, then by Lemma 4.2 and by the black-box proof of binding, there should be a pair of messages $(w_0, w_1) \in \mathcal{W}^2$ such that the malicious receiver $\widehat{R}$ is able to δ distinguish commitments to w_0 and w_1. In this extended version of the protocol, we simply let the honest prover to send (w_0, w_1) to the verifier, and the verifier commits to a random message from the space $\{w_0, w_1\}$ rather than $\{0, 1\}$. The analysis of the soundness and the completeness remains exactly the same.

The only remaining point is the complexity of the prover in how to find (w_0, w_1). But, since $|\mathcal{W}| = \mathrm{poly}(n)$, the honest prover can simply try all possible pairs (w_0, w_1), and simulate the commitment to a random message among them and run $\widehat{R}$ to see whether it guesses the message correctly or not. The prover does this simulation n times for each pair, and for any pair (w_0, w_1), at least $1/2 + \delta/3$ fraction of the guesses were correct, the prover chooses this pair. It is easy to see that by Chernoff bound, unless with negligible probability $\mathrm{negl}(n)$, the prover chooses a pair over(w_0, w_1) which it can be $(\delta/6)$-successful (and note that $\delta/6 > 1/\mathrm{poly}(n)$ is a still sufficiently large gap for the protocol). □

Theorem 6.14 together with Lemma 6.13 prove Theorem **??**.

7 A Note on Non-Interactive Somewhere-Binding Commitments

Recall that the proof of Theorem 4.1 was heavily based on Lemma 4.2. Also recall that Lemma 4.2 was proved in a general form that handles not only standard commitments, but also somewhere-binding commitments. Therefore we get the following stronger separation.

Theorem 7.1. *Suppose there exists a secure implementation of some primitive $\mathcal{P}$ from partially-fixed random oracles (see Definition 3.2) where $\mathcal{P}$ is security-transparent. Then there exists no black-box construction of non-interactive somewhere-binding commitments with a message space $\mathcal{W}$ of polynomial size $|\mathcal{W}| = \mathrm{poly}(n)$ from $\mathcal{P}$.*

It is easy to to see that partially-fixed random oracles, not only imply (super-polynomially) secure one-way functions, but also exponentially (i.e., $2^{\Omega(n)}$)-hard one-way functions. This means that Theorem 7.1 separates non-interactive somewhere-binding commitments for $O(\log n)$-bit message from $2^{\Omega(n)}$-hard one-way functions. In the following we show that this result is almost optimal by presenting a black-box construction of non-interactive somewhere-binding commitments for $\omega(\log n)^2$-bit messages based on the existence of $2^{\Omega(n)}$-hard one-way functions, and discuss how it could potentially be improved to the optimal case of $\omega(\log n)$-bit messages.

Theorem 7.2. *Suppose there exists a $2^{\Omega(n)}$-hard one-way function, then there exists a non-interactive somewhere-binding commitment scheme for $\omega(\log n)^2$-bit messages.*

Proof. Haitner et al. [HHR06] showed (through a black-box construction) that if there exists a $2^{c \cdot m}$-hard one-way function $f \colon \{0,1\}^m \mapsto \{0,1\}^m$, then there exists a pseudorandom generator $g \colon \{0,1\}^k \mapsto \{0,1\}^{k+1}$ for $k = O(m^2)$ which is secure against $2^{c' \cdot m}$-time adversaries where c' is a constant depending on the constant c.

By setting $m = \omega(\log n)$ we get a pseudorandom generator $g\colon \{0,1\}^k \mapsto \{0,1\}^{k+1}$ of seed length $k = O(m^2) = \omega(\log n)^2$ which is secure against $n^{\omega(1)}$-time distinguishers. Our non-interactive somewhere-binding commitment scheme is as follows: Given the message $w \in [2^{k+1}]$, the sender chooses $r \xleftarrow{\$} [2^k]$ at random and sends the commitment $C(w) = w + f(r)$. To decommit, the sender simply reveals (w, r). The hiding of the scheme is due to the pseudorandomness of $g(\mathbf{U}_k)$. The somewhere-binding binding property also holds because there are at most 2^k preimages to any image of g, and so the sender is not able to decommit any commitment value to more than half of the possible messages. $\square$

It is clear from the proof of Theorem 7.2 that any improvement on the seed length of pseudorandom generators from one-way functions would improve the message length of our somewhere-binding commitment scheme. In fact, any "security preserving" construction of pseudorandom generators from one-way functions and with a linear seed length (which also preserves the exponential hardness) would imply a non-interactive somewhere-binding commitment with an optimal $\omega(\log n)$-bit message length. Whether such security preserving pseudorandom generators exist or not is in fact a major open question.

8 Open Questions

In this work we proved a black-box separation of non-interactive commitments from one-way functions. Thus non-interactive commitments are shown to be a natural cryptographic primitive that can be constructed from one-way permutations (or one-to-one one-way functions) but not from general one-way functions. We extended our separation to include one-way functions that are also hitting set generators against co-nondeterministic circuits. We observed that the work of [BOV03] can be interpreted as a non-black-box construction of non-interactive commitments from hitting one-way functions. Thus our separation of non-interactive commitments from hitting one-way functions settles the first pair of cryptographic primitives between which a black-box separation holds while there is a non-black-box construction. To prove the above results we employed the notion of partially-fixed random oracles as a key concept and introduced the notion of partially-defined random oracles and proved some basic concentration bounds for these basic probabilistic objects which we believe to be of independent interest.

Finally we studied the type of non-interactive commitments that can be used in three-message zero-knowledge proofs or arguments (i.e., instance-based non-interactive commitments). We proved that constructing such non-interactive commitments for NP-complete languages based on a black-box use of one-way functions requires finding program checkers for SAT. We also studied three-message honest-verifier zero-knowledge proofs for NP-complete languages directly, and we prove that such proof systems with $O(\log n)$-bit public-coin verifiers (which already include the existing protocols such as the scheme of Goldreich, Micali, and Wigderson [GMW91] and the scheme of Blum [Blu87]) based on a black-box use of one-way functions also requires constructing program checkers for SAT. Whether SAT (i.e., the whole class NP) is checkable or not has been open for more than two decades.

The following are some of the interesting questions remaining open for further research.

1. Are there other natural cryptographic primitives that establish a separation between the power of one-way permutations and one-way functions?

2. Are there more natural pairs of cryptographic primitives where the power of black-box versus non-black-box constructions are different?

3. Are there stronger implausibility consequences, such as the collapse of the polynomial-time hierarchy, assuming that there is a black-box construction of *instance-based* non-interactive commitments from one-way functions? Recall that complexity assumptions are *necessary* for refuting such constructions. Using a *round-efficient* learning algorithm of [MMV11] it can be shown that as long as the sender asks only a constant number of queries, it is possible to get a constant-round protocol in Theorem 6.6 which implies the collapse of the polynomial-time hierarchy, but going beyond this case seems challenging.

4. Can one construct private-coin three-message zero-knowledge proofs for NP based on a black-box construction from one-way functions? Using our techniques one can extend our result about the public-coin case to where the verifier's message is $n^{o(1)}$ bits, assuming the (non-standard) assumption that SAT does not have a program checker of sub-exponential time. However, going beyond the case of $n^{o(1)}$-bit verifiers seems to require new ideas (or assumptions).

Acknowledgment. We thank the anonymous referees of Crypto 2012 for their valuable comments. In particular, we thank Marc Fisclin for a through reading of the paper and valuable comments. We thank Noga Alon for pointing out the anti-concentration bound of Lemma A.2.2 in [AS08] to us (used in the full proof of Proposition **??**). Finally we thank Yuval Ishai for encouraging us to work on the question of separating the power of black-box versus non-back-box cryptographic constructions.

References

[ACP98] Alexander E. Andreev, Andrea E. F. Clementi, and Jose D. P.Rolim, *A new general derandomization method*, JACM: Journal of the ACM **45** (1998). 9

[ACPT99] Alexander E. Andreev, Andrea E. F. Clementi, Jose D. P.Rolim, and Luca Trevisan, *Weak random sources, hitting sets, and BPP simulations*, SICOMP: SIAM Journal on Computing **28** (1999). 9

[AK01] Arvind and Kobler, *On pseudorandomness and resource-bounded measure*, TCS: Theoretical Computer Science **255** (2001). 9

[AKS02] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena, *PRIMES is in P*, Report, Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur-208016, India, August 2002. 3

[AS08] Noga Alon and Joel H. Spencer, *The probabilistic method*, third ed., Wiley, New York, 2008. 32, 42

[Bar01] Boaz Barak, *How to go beyond the black-box simulation barrier.*, Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS), 2001, pp. 106–115. 4

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau, *Minimum disclosure proofs of knowledge*, Journal of Computer and System Sciences **37** (1988), no. 2, 156–189. 1

[BCKT94] Bshouty, Cleve, Kannan, and Tamon, *Oracles and queries that are sufficient for exact learning*, COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers, 1994. 12

[BCY91] Gilles Brassard, Claude Crépeau, and Moti Yung, *Constant-round perfect zero-knowledge computationally convincing protocols*, Theoretical Computer Science **84** (1991), no. 1, 23–52. 1

[BFL90] László Babai, Lance Fortnow, and Carsten Lund, *Non-deterministic exponential time has two-prover interactive protocols*, FOCS, 1990, pp. 16–25. 5

[BI87] Blum and Impagliazzo, *Generic oracles and oracle classes*, FOCS: IEEE Symposium on Foundations of Computer Science (FOCS), 1987. 1

[BK95] Manuel Blum and Sampath Kannan, *Designing programs that check their work*, J. ACM **42** (1995), no. 1, 269–291. 4, 5, 35

[Blu81] Manuel Blum, *Coin flipping by telephone*, CRYPTO, 1981, pp. 11–15. 2, 6

[Blu87] Manuel Blum, *How to prove a theorem so no one else can claim it*, Proceedings of the International Congress of Mathematicians, 1987, pp. 1444–1451. 4, 37, 41

[BM82] Manuel Blum and Silvio Micali, *How to generate cryptographically strong sequences of pseudo random bits*, 1982, pp. 112–117. 1

[BM07] Boaz Barak and Mohammad Mahmoody, *Lower bounds on signatures from symmetric primitives*, FOCS: IEEE Symposium on Foundations of Computer Science (FOCS), 2007. 1, 6, 8, 21, 23

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky, *Perfect zero-knowledge in constant rounds*, Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC), ACM Press, 1990, pp. 482–493. 17

[BOV03] Boaz Barak, Shien Jin Ong, and Salil Vadhan, *Derandomization in cryptography.*, Advances in Cryptology – CRYPTO 2003, Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 299–315. 3, 4, 9, 11, 41

[BPR$^+$08] Boneh, Papakonstantinou, Rackoff, Vahlis, and Waters, *On the impossibility of basing identity based encryption on trapdoor permutations*, FOCS: IEEE Symposium on Foundations of Computer Science (FOCS), 2008. 1

[BR93] M. Bellare and P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, ACM Conference on Computer and Communications Security, November 1993, pp. 62–73. 16

[CDSMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee, *Black-box construction of a non-malleable encryption scheme from any semantically secure one*, TCC (Ran Canetti, ed.), Lecture Notes in Computer Science, vol. 4948, Springer, 2008, pp. 427–444. 2

[CDSMW09] ———, *Simple, black-box constructions of adaptively secure protocols*, TCC (Omer Reingold, ed.), Lecture Notes in Computer Science, vol. 5444, Springer, 2009, pp. 387–402. 2

[DPP98] Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann, *Statistical secrecy and multibit commitments*, IEEE Transactions on Information Theory **44** (1998), no. 3, 1143–1151. 1

[DSLMM11] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin, *On black-box complexity of optimally-fair coin-tossing*, Theory of Cryptography Conference - TCC 2011, 2011. 1, 6, 21

[FRS88] Lance Fortnow, John Rompel, and Michael Sipser, *On the power of multi-prover interactive protocols*, Theoretical Computer Science, 1988, pp. 156–161. 5

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan, *Bounds on the efficiency of generic cryptographic constructions*, SIAM Journal on Computing **35** (2005), no. 1, 217–246. 1

[GK92] Oded Goldreich and Hugo Krawczyk, *Sparse pseudorandom distributions*, Random Structures & Algorithms **3** (1992), no. 2, 163–174. 4

[GK96] Oded Goldreich and Ariel Kahan, *How to construct constant-round zero-knowledge proof systems for NP*, Journal of Cryptology **9** (1996), no. 3, 167–190. 1

[GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby, *On the existence of pseudorandom generators*, SIAM Journal on Computing **22** (1993), no. 6, 1163–1175. 1

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan, *The relationship between public key encryption and oblivious transfer*, Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000. 1

[GL89] Oded Goldreich and Leonid A. Levin, *A hard-core predicate for all one-way functions*, Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), 1989, pp. 25–32. 1, 2

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM Journal on Computing **17** (1988), no. 2, 281–308, Preliminary version in *FOCS'84*. 1

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, *The knowledge complexity of interactive proof systems*, no. 1, 186–208, Preliminary version in *STOC'85*. 2, 34

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold, *On the impossibility of basing trapdoor functions on trapdoor predicates*, FOCS, 2001, pp. 126–135. 1, 7

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson, *How to play any mental game or a completeness theorem for protocols with honest majority*, 1987, pp. 218–229. 2, 4, 37

[GMW91] ———, *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems*, Journal of the ACM **38** (1991), no. 1, 691–729, Preliminary version in *FOCS'86*. 2, 41

[Goy11] Vipul Goyal, *Constant round non-malleable protocols using one way functions*, 2011. 2

[GT00] Rosario Gennaro and Luca Trevisan, *Lower bounds on the efficiency of generic cryptographic constructions*, Proceedings of the 41st Annual Symposium on Foundations of Computer Science, 2000, pp. 305–313. 12

[GTS07] Dan Gutfreund and Amnon Ta-Shma, *Worst-case to average-case reductions revisited*, APPROX-RANDOM (Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, eds.), Lecture Notes in Computer Science, vol. 4627, Springer, 2007, pp. 569–583. 12

[GV08] Dan Gutfreund and Salil P. Vadhan, *Limitations of hardness vs. randomness under uniform reductions*, APPROX-RANDOM, 2008, pp. 469–482. 12

[GW99] Oded Goldreich and Avi Wigderson, *Improved derandomization of bpp using a hitting set generator*, Proceedings of the RANDOM 99 Conference, 1999, pp. 131–137. 9

[Hai08] Iftach Haitner, *Semi-honest to malicious oblivious transfer - the black-box way*, Theory of Cryptography, Fourth Theory of Cryptography Conference, TCC 2008, 2008, pp. 394–409. 2

[HHK+05] Iftach Haitner, Omer Horvitz, Jonathan Katz, Chiu-Yuen Koo, Ruggero Morselli, and Ronen Shaltiel, *Reducing complexity assumptions for statistically-hiding commitment*, Advances in Cryptology – EUROCRYPT 2005, 2005, See also preliminary draft of full version, `www.wisdom.weizmann.ac.il/~iftachh/papers/SCfromRegularOWF.pdf`, pp. 58–77. 1

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold, *Efficient pseudorandom generators from exponentially hard one-way functions*, Automata, Languages and Programming, 24th International Colloquium, ICALP, 2006. 40

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev, *Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments*, Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, 2007. 1

[HHry] Juris Hartmanis and Lane A. Hemachandra, *One-way functions, robustness, and the non-isomorphism of NP-complete sets*, Tech. Report 86-796, Department of Computer Science, Cornell University, 1987, January. 1

[HIK+11] Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank, *Black-box constructions of protocols for secure computation*, SIAM J. Comput **40** (2011), no. 2, 225–266. 2

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby, *A pseudorandom generator from any one-way function*, SIAM Journal on Computing **28** (1999), no. 4, 1364–1396, Preliminary versions in *STOC'89* and *STOC'90*. 1, 2, 18

[HNO+07] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan, *Statistically-hiding commitments and statistical zero-knowledge arguments from any one-way function*, SIAM Journal on Computing, November 2007. 1

[HO11] Iftach Haitner and Eran Omri, *Coin flipping with constant bias implies one-way functions*. 1

[HR04] Hsiao and Reyzin, *Finding collisions on a public road, or do secure hash functions need secret coins?*, CRYPTO: Proceedings of Crypto, 2004.

[HR07] Iftach Haitner and Omer Reingold, *Statistically-hiding commitment from any one-way function*, Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, 2007. 1

[HRVW09] Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee, *Inaccessible entropy*, 2009. 1

[IL89] Russell Impagliazzo and Michael Luby, *One-way functions are essential for complexity based cryptography*, Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS), 1989, pp. 230–235. 1, 2, 19

[IOS97] Itoh, Ohta, and Shizuya, *A language-dependent cryptographic primitive*, JCRYPTOL: Journal of Cryptology **10** (1997). 17

[IR89] Russell Impagliazzo and Steven Rudich, *Limits on the provable consequences of one-way permutations*, Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), ACM Press, 1989, pp. 44–61. 1, 12, 13, 16

[KMS07] Bruce Kapron, Lior Malka, and Venkatesh Srinivasan, *A characterization of non-interactive instance-dependent commitment-schemes (NIC)*, Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Lecture Notes in Computer Science, Springer, 2007. 37

[KSS00] Jeff Kahn, Michael Saks, and Cliff Smyth, *A dual version of Reimer's inequality and a proof of Rudich's conjecture*, 15th Annual IEEE Conference on Computational Complexity, 2000, pp. 98–103. 1

[KST99] Jeong Han Kim, Daniel R. Simon, and Prasad Tetali, *Limits on the efficiency of one-way permutation-based hash functions*, FOCS, 1999, pp. 535–542. 1

[KSY11] Katz, Schrder, and Yerukhimovich, *Impossibility of blind signatures from one-way permutations*, TCC: Theory of Cryptography Conference, 2011. 1

[KvM02] Adam Klivans and Dieter van Melkebeek, *Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses*, SIAM J. Comput **31** (2002), no. 5, 1501–1526. 9

[Lev87] Leonid A. Levin, *One-way functions and pseudorandom generators*, Combinatorica **7** (1987), 357–363. 1

[LHWL93] Arjen K. Lenstra and Jr. Hendrik W. Lenstra (eds.), *The development of the number field sieve*, Lecture Notes in Mathematics, vol. 1554, Springer-Verlag, Berlin, 1993. 1

[LTW05] Lin, Trevisan, and Wee, *On hardness amplification of one-way functions*, Theory of Cryptography Conference (TCC), LNCS, vol. 2, 2005. 1

[Mil76] Gary L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences **13** (1976), no. 3, 300–317. 3

[MM11] Matsuda and Matsuura, *On black-box separations among injective one-way functions*, TCC: Theory of Cryptography Conference, 2011. 1

[MMV11] Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan, *Time-lock puzzles in the random oracle model*, CRYPTO (Phillip Rogaway, ed.), Lecture Notes in Computer Science, vol. 6841, Springer, 2011, pp. 39–50. 42

[MP] Mohammad Mahmoody and Rafael Pass, *Black-box separations; handing non-uniform proofs of security*, Submitted. 21

[MV05] Miltersen and Vinodchandran, *Derandomizing arthur-merlin games using hitting sets*, CMPCMPL: Computational Complexity **14** (2005). 9

[Nao91] Moni Naor, *Bit commitment using pseudorandomness*, Journal of Cryptology **4** (1991), no. 2, 151–158, Preliminary version in *CRYPTO'89*. 2, 9, 18

[Nao03] Naor, *On cryptographic assumptions and challenges*, CRYPTO: Proceedings of Crypto, 2003. 10

[NOV06] Minh-Huyen Nguyen, Shien Jin Ong, and Salil Vadhan, *Statistical zero-knowledge arguments for NP from any one-way function*, Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS), 2006, pp. 3–14. 1

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung, *Perfect zero-knowledge arguments for NP using any one-way permutation*, Journal of Cryptology **11** (1998), no. 2, 87–108, Preliminary version in *CRYPTO'92*. 1

[OV07] Shien Jin Ong and Salil Vadhan, *Zero knowledge and soundness are symmetric*, Advances in Cryptology – EUROCRYPT 2007, 2007, pp. 187–209. 37

[OW93] Rafail Ostrovsky and Avi Wigderson, *One-way fuctions are essential for non-trivial zero-knowledge*, ISTCS, 1993, pp. 3–17. 1

[PW09] Rafael Pass and Hoeteck Wee, *Black-box constructions of two-party protocols from one-way functions*, TCC (Omer Reingold, ed.), Lecture Notes in Computer Science, vol. 5444, Springer, 2009, pp. 403–418. 2

[Rab80] Michael O. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory **12** (1980), no. 1, 128–138. 3

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan, *Notions of reducibility between cryptographic primitives.*, Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Lecture Notes in Computer Science, vol. 2951, Springer, 2004, pp. 1–20. 1, 5, 18

[Rud88] Steven Rudich, *Limits on the provable consequences of one-way functions*, Ph.D. thesis, U.C. Berkeley, 1988. 1

[Sim98] Daniel Simon, *Finding collisions on a one-way street: Can secure hash functions be based on general assumptions?*, Advances in Cryptology – EUROCRYPT '98, Lecture Notes in Computer Science, vol. 1403, Springer, 1998, pp. 334–345. 1

[Tar89] Gábor Tardos, *Query complexity, or why is it difficult to seperate NP^{A} cap co NP^{A} from P^{A} by random oracles A?*, Combinatorica **9** (1989), no. 4, 385–392. 1

[Vah10] Yevgeniy Vahlis, *Two is a crowd? A black-box separation of one-wayness and security under correlated inputs*, TCC (Daniele Micciancio, ed.), Lecture Notes in Computer Science, vol. 5978, Springer, 2010, pp. 165–182. 1

[Wee10] Hoeteck Wee, *Black-box, round-efficient secure computation via non-malleability amplification*, FOCS, IEEE Computer Society, 2010, pp. 531–540. 2

[Yao82] Andrew C. Yao, *Theory and applications of trapdoor functions*, 1982, pp. 80–91. 1