The Curious Case of Non-Interactive Commitments – On the Power of Black-Box vs. Non-Black-Box Use of Primitives

Mohammad Mahmoody and Rafael Pass*

Cornell
{mohammad,rafael}@cs.cornell.edu

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.

* 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.

R. Safavi-Naini and R. Canetti (Eds.): CRYPTO 2012, LNCS 7417, pp. 701–718, 2012.

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 [44] 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., [12,21,22,46,51,62,64][1]). Reingold, Trevisan, and Vadhan [60] 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 [38, 43, 57]; 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 [8, 41, 45, 61, 63][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 [11, 25, 26, 42, 49, 66] and statistical zero-knowledge arguments as well as statistically-hiding commitments [13, 14, 18, 23, 31, 35, 37, 39, 40, 55, 56].

[1] A closely related line of research aimed at proving lower-bounds on the *efficiency* of black-box constructions (e.g., [6, 17, 19, 34, 47, 50]).

[2] Our notion of black-box construction here corresponds to the notion of *fully* black-box construction as defined in [60] where we also include the security parameter; see Definition 4.

[3] The results implicit in [8, 41, 63] show that there is no fully black-box construction of one-way permutations from one-way functions (see [51] for an exposition of this argument). This results extends even to separating one-way functions from injective one-way functions. Rudich [61] 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 [45]. See [61] for more discussions.

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 [48] 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:

> **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 [15,16,32,36,65], oblivious transfer from semi-honest oblivious transfer [33], constant-round zero-knowledge arguments and trapdoor commitments from one-way functions [58], 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 [28,30] to secure computations [27]. 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 [54] and Håstad, Impagliazzo, Luby and Levin [42] 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 [43] establish that the existence of commitment schemes implies the existence of one-way functions (in a black-box way).

In this work 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 [9, 26] 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). In this work we study the following natural question left open by previous work: *Is there a black-box construction of non-interactive commitments from one-way functions?* We provide a negative answer:

Theorem 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 [7] 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 3) against co-non-deterministic circuits (see Definition 6 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.

[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 [1, 52, 59] and this guarantees f_p is indeed a permutation.

Conditional Separation of the Power of Black-Box and Non-Black-Box Constructions. The result of [7] together with our Theorem 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). 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.

Non-Interactive Commitments from Hitting One-Way Functions. Inspired by the work of [7], 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-nondeterministic circuit of size n which accepts at least half of its inputs, there exists at least one input $x \in [1, \ldots, n^2] \subseteq \{0,1\}^n$ 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 8). Furthermore, we show that there exists a non-black-box construction of non-interactive commitments from hitting one-way functions as follows. Following [7], we derandomize Naor's commitment scheme by evaluating the hitting one-way function f on the inputs $1, \ldots, n^2$ (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 whose proof can be found in the full version of the paper.[5]

Theorem 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.

Theorem 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 [5] and Goldreich-Krawczyk [24] 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.

Extensions to 3*-Message Zero-Knowledge and Instance-Based Commitments.* A major application of commitment schemes is to construct zero-knowledge proofs for NP. We also directly study the constructions of 3-message zero-knowledge

[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.

proofs based on one-way functions and also the type of non-interactive (instance-based) commitments that are useful to construct such zero-knowledge proofs. We extend our impossibility result (of black-box constructions from one-way functions) also for these primitives, but in a *conditional* way. Namely, our separations hold assuming that the complexity class NP does not have "program checkers" [10]. For these results we refer the reader to the full version of the paper.

2 Separation from One-Way Functions

Here we outline the proof of Theorem 1. Due to lack of space, 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 refer the reader to full version of the paper for the proof of the general case.

We start by formalizing the notion of black-box constructions by following the paradigm of [60] and incorporating the security parameter. 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 4. *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 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, we employ the methodology formally described in the following lemma (which is also used in the previous works of [6,17]). See [17] for a proof of a stronger version of this lemma.

Lemma 5. *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* $\text{COM}^{\mathbf{O}}$ *is violated by a* $\text{poly}(n)$*-query attack.*
2. $\mathbf{O}$ *is strongly one-way in the sense that no* $\text{poly}(n)$*-query computationally-unbounded adversary can invert* $\mathbf{O}$ *over* $\mathbf{O}(\mathbf{U}_n)$ *with probability* $\geq 1/\text{poly}(n)$.

In the following we describe how to find a distribution for the randomized oracle $\mathbf{O}$ so that we can apply Lemma 5 to prove Theorem 1.

$\mathbf{O}$ *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[6] which is indeed sufficient for constructing non-interactive commitments in a black-box way [9].

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 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 planted in an adversarial way against the construction COM and that is why the distribution of $\mathbf{O}$ depends on COM.[7]

It is easy to see that a partially-fixed random oracle is still hard to invert using $\text{poly}(n)$-query attacks. We show how to define the the distribution of $\mathbf{O}$ such that, either of the binding or hiding properties of $\text{COM}^{\mathbf{O}}$ will be violated through a $\text{poly}(n)$-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 $\text{COM}^{\mathbf{O}_R}$ is broken by a $\text{poly}(n)$-query malicious receiver $\widehat{R}$.
2. The binding of $\text{COM}^{\mathbf{O}_S}$ is broken by a $\text{poly}(n)$-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 COM is broken by some $\text{ADV} \in \{\widehat{R}, \widehat{S}\}$.

[6] 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.

[7] 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. [22].

The 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 $1/2 + 1/\operatorname{poly}(n)$, 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.

2.1 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:

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

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 [6] shows that such learning algorithm would (on average) ask at most $\operatorname{poly}(n/\varepsilon) = \operatorname{poly}(n)$ 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 $\widehat{R}$.

2.2 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 $1/\operatorname{poly}(n)$ over the executing of the learning algorithm, at the end there is a value $b \in \{0,1\}$ such that $\Pr[b \text{ is the committed bit } \mid (C, \mathcal{L})] > 1/2 + 1/\operatorname{poly}(n)$. 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 $1/2 + 1/\operatorname{poly}(n)$.

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 [54] 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).

3 Separation from Hitting One-Way Functions

In this Section we outline the proof of Theorem 3. Before doing so we need to develop the notion of a hitting one-way function.

3.1 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 $\mathrm{poly}(n)$ 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 [29] 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 [3,4]. Here we are interested in the notion of hitting set generators against co-nondeterministic circuits defined as follows.

Definition 6 (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 7 (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 $\{f(1)|_m, \ldots f(n^2)|_m\} \cap \mathcal{S}_T \neq \varnothing$ where $1, 2, \ldots, n^2$ 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 (details in the full version) 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 [53].[8]

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 4 and Definition 7. 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).

3.2 Outline of the Proof of Theorem 3

In order to prove Theorem 3, we rely on the proof of Theorem 1 outlined in Section 2. 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 3 as a direct extension of the proof of Theorem 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.[9] We first show that a random oracle is also a hitting function with overwhelming probability (and so it will be hitting one-way).

[8] 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 $w_1, \ldots, w_{n^2}$ such that $T(f(i)|_m, w_i) = 1$ for all $i \in [n^2] \subset \{0,1\}^n$.

[9] Recall that our random oracle chooses its randomness after the adversary is fixed and is different from the settings of [20,44] who *fix* the random oracle after sampling it once and for all.

Lemma 8. *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[10] n is at most $2^{O(n \log n)}$, the lemma follows by a union bound.

Lemma 8 implies that for large enough n a random function from $\{0,1\}^n$ to $\{0,1\}^n$ is hitting with overwhelming probability. Therefore Lemma 8 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 Sec (of any potential black-box construction Com) can output to refute the hitting property of f. Therefore, in this case the security reduction Sec might as well just try to invert the random oracle (with the help of the adversary). This means that, 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 where the security reduction only tries to invert f, but we have already settled this case!

Generalization. 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 [44] 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, 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

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

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 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 *among the queries 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 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 3. The following three propositions imply Theorem 3.

Proposition 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.

Proposition 11 ($\mathbf{O}_S$ Remains One-Way). *No* $\mathrm{poly}(n)$*-query adversary* ADV *can invert* $\mathbf{O}_S(\mathbf{U}_n)$ *with probability* $1/\mathrm{poly}(n)$*, even when* 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 3 is found in the following proposition. Due to lack to space, we only provide a very high-level outline.

Proposition 12. $\mathbf{O}_S$ *is hitting with overwhelming probability.*

Proof (Outline). 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.

Formalizing this intuition, however, is far from easy and the challenge stems from the fact that, as we said before, we want the probability ρ to be extremely close to one. Because of this, we can not afford the $1/\mathrm{poly}(n)$ probability that queries the sampled views V_0 an V_1 might have collisions (out of $\mathcal{L}$) and resample $\mathbf{O}_S$ again. Therefore, we define the oracle $\mathbf{O}_S$ in a randomized way, *even* when such collisions happen. To prove that the sample oracle $\mathbf{O}_S$ is hitting with very high probability, we develop and employ new concentration bounds to control the probability that $\mathbf{O}_S$ is not hitting. We refer the reader for the full version the paper for the full proof.

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 [2] to us (used in the full proof of Proposition 12). 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

1. Agrawal, M., Kayal, N., Saxena, N.: PRIMES is in P. Report, Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur-208016, India (August 2002)
2. Alon, N., Spencer, J.H.: The probabilistic method, 3rd edn. Wiley, New York (2008)
3. Andreev, A.E., Clementi, A.E.F., Rolim, J.D.P.: A new general derandomization method. JACM: Journal of the ACM 45 (1998)
4. Andreev, A.E., Clementi, A.E.F., Rolim, J.D.P., Trevisan, L.: Weak random sources, hitting sets, and BPP simulations. SICOMP: SIAM Journal on Computing 28 (1999)
5. Barak, B.: How to go beyond the black-box simulation barrier. In: Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 106–115 (2001)
6. Barak, B., Mahmoody, M.: Lower bounds on signatures from symmetric primitives. In: FOCS: IEEE Symposium on Foundations of Computer Science (2007)
7. Barak, B., Ong, S.J., Vadhan, S.: Derandomization in Cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg (2003)
8. Blum, Impagliazzo: Generic oracles and oracle classes. In: FOCS: IEEE Symposium on Foundations of Computer Science (1987)
9. Blum, M.: Coin flipping by telephone. In: CRYPTO, pp. 11–15 (1981)
10. Blum, M., Kannan, S.: Designing programs that check their work. J. ACM 42(1), 269–291 (1995)
11. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo random bits, pp. 112–117 (1982)
12. Boneh, Papakonstantinou, Rackoff, Vahlis, Waters: On the impossibility of basing identity based encryption on trapdoor permutations. In: FOCS: IEEE Symposium on Foundations of Computer Science (2008)
13. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences 37(2), 156–189 (1988)
14. Brassard, G., Crépeau, C., Yung, M.: Constant-round perfect zero-knowledge computationally convincing protocols. Theoretical Computer Science 84(1), 23–52 (1991)
15. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Black-Box Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 427–444. Springer, Heidelberg (2008)
16. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, Black-Box Constructions of Adaptively Secure Protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 387–402. Springer, Heidelberg (2009)
17. Dachman-Soled, D., Lindell, Y., Mahmoody, M., Malkin, T.: On the Black-Box Complexity of Optimally-Fair Coin Tossing. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 450–467. Springer, Heidelberg (2011)

18. Damgård, I.B., Pedersen, T.P., Pfitzmann, B.: Statistical secrecy and multibit commitments. IEEE Transactions on Information Theory 44(3), 1143–1151 (1998)
19. Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM Journal on Computing 35(1), 217–246 (2005)
20. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 305–313 (2000)
21. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (2000)
22. Gertner, Y., Malkin, T., Reingold, O.: On the impossibility of basing trapdoor functions on trapdoor predicates. In: FOCS, pp. 126–135 (2001)
23. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology 9(3), 167–190 (1996)
24. Goldreich, O., Krawczyk, H.: Sparse pseudorandom distributions. Random Structures & Algorithms 3(2), 163–174 (1992)
25. Goldreich, O., Krawczyk, H., Luby, M.: On the existence of pseudorandom generators. SIAM Journal on Computing 22(6), 1163–1175 (1993)
26. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), pp. 25–32 (1989)
27. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority, pp. 218–229 (1987)
28. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(1), 691–729 (1991); Preliminary version in FOCS 1986
29. Goldreich, O., Wigderson, A.: Improved Derandomization of BPP Using a Hitting Set Generator. In: Hochbaum, D.S., Jansen, K., Rolim, J.D.P., Sinclair, A. (eds.) RANDOM-APPROX 1999. LNCS, vol. 1671, pp. 131–137. Springer, Heidelberg (1999)
30. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989); Preliminary version in STOC 1985
31. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988); Preliminary version in FOCS 1984
32. Goyal, V.: Constant round non-malleable protocols using one way functions (2011)
33. Haitner, I.: Semi-honest to Malicious Oblivious Transfer—The Black-Box Way. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 412–426. Springer, Heidelberg (2008)
34. Haitner, I., Hoch, J.J., Reingold, O., Segev, G.: Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In: Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS). IEEE Computer Society (2007)
35. Haitner, I., Horvitz, O., Katz, J., Koo, C.-Y., Morselli, R., Shaltiel, R.: Reducing Complexity Assumptions for Statistically-Hiding Commitment. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 58–77. Springer, Heidelberg (2005), See also preliminary draft of full version `www.wisdom.weizmann.ac.il/~iftachh/papers/SCfromRegularOWF.pdf`
36. Haitner, I., Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions of protocols for secure computation. SIAM J. Comput. 40(2), 225–266 (2011)

37. Haitner, I., Nguyen, M.-H., Ong, S.J., Reingold, O., Vadhan, S.: Statistically-hiding commitments and statistical zero-knowledge arguments from any one-way function. SIAM Journal on Computing (November 2007)
38. Haitner, I., Omri, E.: Coin flipping with constant bias implies one-way functions (2011)
39. Haitner, I., Reingold, O.: Statistically-hiding commitment from any one-way function. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC). ACM Press (2007)
40. Haitner, I., Reingold, O., Vadhan, S.P., Wee, H.: Inaccessible entropy (2009)
41. Hartmanis, J., Hemachandra, L.A.: One-way functions, robustness, and the non-isomorphism of NP-complete sets. Technical Report 86-796, Department of Computer Science, Cornell University (January 1987)
42. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999); Preliminary versions in STOC 1989 and STOC 1990
43. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography. In: Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS), pp. 230–235 (1989)
44. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), pp. 44–61. ACM Press (1989)
45. Kahn, J., Saks, M., Smyth, C.: A dual version of Reimer's inequality and a proof of Rudich's conjecture. In: 15th Annual IEEE Conference on Computational Complexity, pp. 98–103 (2000)
46. Katz, J., Schröder, D., Yerukhimovich, A.: Impossibility of Blind Signatures from One-Way Permutations. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 615–629. Springer, Heidelberg (2011)
47. Kim, J.H., Simon, D.R., Tetali, P.: Limits on the efficiency of one-way permutation-based hash functions. In: FOCS, pp. 535–542 (1999)
48. Lenstra, A.K., Lenstra Jr., H.W. (eds.): The development of the number field sieve. Lecture Notes in Mathematics, vol. 1554. Springer, Berlin (1993)
49. Levin, L.A.: One-way functions and pseudorandom generators. Combinatorica 7, 357–363 (1987)
50. Lin, H., Trevisan, L., Wee, H.: On Hardness Amplification of One-Way Functions. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 34–49. Springer, Heidelberg (2005)
51. Matsuda, T., Matsuura, K.: On Black-Box Separations among Injective One-Way Functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 597–614. Springer, Heidelberg (2011)
52. Miller, G.L.: Riemann's hypothesis and tests for primality. Journal of Computer and System Sciences 13(3), 300–317 (1976)
53. Naor, M.: On Cryptographic Assumptions and Challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)
54. Naor, M.: Bit commitment using pseudorandomness. Journal of Cryptology 4(2), 151–158 (1991); Preliminary version In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 128–136. Springer, Heidelberg (1990)
55. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way permutation. CRYPTO 1992 11(2), 87–108 (1998); Preliminary version in Brickell, E.F. (ed.): CRYPTO 1992. LNCS, vol. 740. Springer, Heidelberg (1993)

56. Nguyen, M.-H., Ong, S.J., Vadhan, S.: Statistical zero-knowledge arguments for NP from any one-way function. In: Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS), pp. 3–14 (2006)
57. Ostrovsky, R., Wigderson, A.: One-way fuctions are essential for non-trivial zero-knowledge. In: ISTCS, pp. 3–17 (1993)
58. Pass, R., Wee, H.: Black-Box Constructions of Two-Party Protocols from One-Way Functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (2009)
59. Rabin, M.O.: Probabilistic algorithm for testing primality. Journal of Number Theory 12(1), 128–138 (1980)
60. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of Reducibility between Cryptographic Primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004)
61. Rudich, S.: Limits on the Provable Consequences of One-Way Functions. PhD thesis, U.C. Berkeley (1988)
62. Simon, D.R.: Findings Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 334–345. Springer, Heidelberg (1998)
63. Tardos, G.: Query complexity, or why is it difficult to seperate NP^A cap co NP^A from P^A by random oracles A? Combinatorica 9(4), 385–392 (1989)
64. Vahlis, Y.: Two Is a Crowd? A Black-Box Separation of One-Wayness and Security under Correlated Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 165–182. Springer, Heidelberg (2010)
65. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: FOCS, pp. 531–540. IEEE Computer Society (2010)
66. Yao, A.C.: Theory and applications of trapdoor functions, pp. 80–91 (1982)