On Seed-Incompressible Functions

Shai Halevi[1], Steven Myers[2], and Charles Rackoff[3]

[1] IBM Research shaih@alum.mit.edu
[2] Indiana University samyers@indiana.edu
[3] University of Toronto rackoff@cs.toronto.edu

Abstract. We investigate a new notion of security for "cryptographic functions" that we term *seed incompressibility* (SI). We argue that this notion captures some of the intuition for the alleged security of constructions in the random-oracle model, and indeed we show that seed incompressibility suffices for some applications of the random oracle methodology. Very roughly, a function family $f_s(\cdot)$ with $|s| = n$ is seed incompressible if given (say) $n/2$ bits of advice (that can depend on the seed s) and an oracle access to $f_s(\cdot)$, an adversary cannot "break $f_s(\cdot)$" any better than given only oracle access to $f_s(\cdot)$ and no advice.
The strength of this notion depends on what we mean by "breaking $f_s(\cdot)$". We first show that for any family f_s there exists an adversary that can distinguish $f_s(\cdot)$ from a random function using $n/2$ bits of advice, so seed incompressible pseudo-random functions do not exist. Then we consider the weaker notion of seed-incompressible correlation intractability. We show that although the negative results can be partially extended also to this weaker notion, they cannot rule it out altogether. More importantly, the settings that we cannot rule out still suffice for many applications. In particular, we show that they suffice for constructing collision-resistant hash functions and for removing interaction from Σ-protocols (3-round honest verifier zero-knowledge protocols).

1 Introduction

Identifying useful security notions of "cryptographic functions" was proposed ten years ago by Canetti [4], as a plausible way of putting random-oracle-based constructions on a firmer theoretical footing. The challenge is to find specific "random-oracle-like" properties, such that functions with these properties (a) can be realized in the standard model and (b) can be securely used in some cryptographic applications in lieu of access to a truly random function. However, very little progress along this line has been made since then, in fact the only non-obvious notion along this line that we know of is the "perfect one-way hashing" notion of Canetti [4, 7].

In this work we study a very different security notion that we term *seed incompressibility*. On a very high level, this notion is meant to capture the intuition that a random function has no structure. At a first glance, it seems hopeless to define an efficiently computable function that has no structure, since the fact that the function is computed by a small circuit is itself some structure. However, we may still hope that this small circuit is the only interesting "small property" of the function. That is, no adversary can find a *significantly smaller property* that differentiates it from your average random function. Roughly, if you do not get enough bits to describe the entire function, then you get nothing.

Toward formalizing this intuition, let $F = \{f_s\}_s$ be a family of functions with n-bit seeds, and consider an adversary that works in two phases: In the first phase the adversary gets the n-bit seed s, compresses it to (say) an $n/2$-bit string σ, and keeps only σ in memory. Then the adversary gets an oracle access to the function $f_s(\cdot)$, and it tries to "break it" (according to some notion of security). We call this the *seed compression* attack model. We say that the family F satisfies the underlying notion of security under seed-compression attack (or that it is *seed incompressible* with respect to the underlying notion of security), if breaking the function knowing σ is not any easier than breaking it without knowing σ.[4]

The choice of $n/2$ as the compression threshold is quite arbitrary. The results that we present in this paper remain unchanged whenever the threshold for the length of the compressed seed is anywhere from n^ϵ to $n - n^\epsilon$ for any fixed $0 < \epsilon < 1$. Below we stick to the $n/2$ threshold for convenience.

Following the intuition from above, we would have liked a construction where it is not possible to distinguish f_s from a random function, even given σ. Some care must be taken when defining this notion to avoid obvious pitfalls (such as σ being the first $n/2$ bits of $f_s(0)$), but this can be handled using ideas similar to the ones of Coron et al. [9] (see details in Section 3). Unfortunately, even with these ideas we show that the resulting notion cannot be realized, namely no function family can be pseudo-random under seed-compression attacks. Roughly, the reason is that the adversary can encode in σ a CS-proof [21] for the statement that f_s is computed by a small circuit. This impossibility result is somewhat disheartening, as it does not really show the existence of some property of the function that is smaller than its description; rather, it is simply a fact that convincing someone that a function has a small circuit takes much fewer bits than actually telling them what the circuit is. Further, from a security perspective the fact that the function is being computed by a small circuit is clearly information that the adversary knows.

Faced with this negative result, we investigate weaker notions of security. One direction that seems promising is the notion of *correlation-intractability* under seed-compression attacks. The notion of correlation-intractability was defined by Canetti et al. [6] as the inability of the attacker to find any input x such that the pair $(x, f_s(x))$ satisfies any "non-trivial relation" (cf. Section 4). Canetti et al. proved that correlation-intractability is not realizable when the adversary sees the entire seed s, but we point out that it may be realizable when the adversary is only given the "compressed seed" σ. We note that the negative results from [6] do not seem to extend to this model. On the other hand, our negative results for PRFs can be partially extended also to this weaker notion. However, it seems that these negative results hit an inherent limitation for some

[4] Of course, this is only meaningful if "breaking f_s" is hard without any knowledge of the seed s.

parameter settings, and we show in Section 4 that the remaining parameter settings are still useful. For example, we show that we can use them to construct collision-resistant hash functions, and perhaps more interestingly that we can use them to remove interaction from three-move public-coin honest-verifier zero-knowledge proofs.

Briefly, the primitives that can be constructed from seed-incompressible functions are those for which a "break" can be encoded with only a few bits. (For example, one can encode a collision in a hash function using only two inputs to the function.) When constructing such primitives from seed-incompressible functions, we let the seed of the function be sufficiently longer than the number of bits that are needed to encode a break, and then any adversary that breaks the resulting primitive can be converted into a "compressor" (that given the seed outputs a break), thus violating the seed-incompressibility of the underlying function.

We unfortunately were not able to find a construction that provably achieves seed-incompressible correlation intractability under a better-known computational hardness assumption. Still, one can conjecture that ad-hoc constructions such as AES or HMAC-SHA1 have this property. Such a conjecture is theoretically more appealing than using the random-oracle model since, at the very least, we do not have a proof that it is false, while still providing a conjecture open to disproof. We explain below our intuition for why one might conjecture that AES and SHA type constructions satisfy our definitions.

1.1 Seed-Incompressibility and Contemporary Block-Ciphers

Here is one way to use the intuition behind DES and AES like block-cipher constructions to possibly construct seed-incompressible functions. We will use AES to denote any similar composition-of-round based block-cipher construction. Each AES function is expressed as the composition of r rounds of permutations $p_{k_1} \circ \ldots \circ p_{k_r}$, where each n-bit k_i is determined by the key. For this theoretical presentation, instead, we assume that all the "round keys" k_i are chosen randomly and independently. Consider now using not r rounds but nr (independent) rounds. It seems unlikely that a key of this new construction can be "compressed" to only (say) $n/2$ bits. The intuition (at least for the case where the compressed seed consists of actual key bits), is that if the compressed seed is so short then there must be r consecutive rounds for which the key is completely undetermined, which in some intuitive sense is as strong as a standard r-round construction of an AES like construction. One issue for block-cipher like constructions is their invertibility, but by choosing a compressed seed of only $n/2$ bits, then it is not enough to help inverting the AES function. For example giving $n/2$ bits from the pre-image of zero does not appear to help when the AES permutation is mapping over an n-bit domain.

1.2 Related Work

The Random-Oracle Model. Following the Fiat-Shamir heuristic for transforming public-coin identification protocols into signature schemes [14] and several other uses of random-oracles in the literature, although sometimes used in different contexts (e.g., [18]), Bellare and Rogaway formalized the random-oracle heuristic as a "general-purpose" design methodology for cryptographic schemes [2] and emphasized the need to develop formal proofs of security within it. The methodology requires that one first design an ideal system in which all parties (including the adversary) have oracle access to a truly random function and prove the security of this ideal system. (The proof is called "a security proof in the random-oracle model.") Next, one replaces the random oracle by a "cryptographic hash function" (such as SHA), where all parties (including the adversary) have a succinct description of this function. Thus, one obtains an implementation of the ideal system in a "real-world" where random oracles do not exist.

The random-oracle methodology has been used quite extensively since then, often resulting in very efficient and seemingly secure schemes. A drawback of this methodology, however, is that it is not at all clear what security properties are needed from the cryptographic hash function in order for a specific scheme to be secure. In fact, Canetti et al. demonstrated that this methodology is not sound in general, in that there exist secure "ideal schemes" that have no secure implementation in the "real world" [6]. A similar negative result was later proved by Goldwasser and Kalai also for the original Fiat-Shamir heuristic [16].

Still, there are many cryptographic schemes whose only known security proof is in the random-oracle model, some of which withstood substantial cryptanalysis and are widely implemented and deployed. Seeking to provide some theoretical footing to the security of such schemes, we would like to be able to describe "random-oracle like" properties that are (a) well-defined, (b) realizable, and (c) sufficient for the security of some instances of the random-oracle methodology. As we mentioned above, a first step in this direction was taken a decade ago by Canetti et al. with the notion of perfect one-way hashing [4, 7].

With respect to the realizability of such notion, one thing that we could have hoped for is to prove its existence based on a more standard cryptographic assumption (e.g., the hardness of factoring). We point out, however, that at least as important is that there will be some hope (or intuition) that typical cryptographic functions such as SHA or AES actually fulfill this notion (as it is these functions that are used in actual implementations of protocols proven correct in the Random Oracle model).

Conditional Entropy Hash Functions. Barak et al. [1] conjectured the existence of families of hash functions h_s for which no attacker can generate an input that has a predictable output. Specifically, a keyed function $h_s(x)$ is said to "ensure conditional entropy e" if for every attacker A that takes as input the key s and produces as output an input x to h, it holds that the conditional entropy $H(h_s(A(s))|A(s)) \geq e$. This notion appears to be very close to the notion of correlation intractable functions of [6] (since if $h_s(x)$ has "high entropy" then it is unlikely to hit any evasive relation).

Barak et al. show that such functions are sufficient to implement the Fiat-Shamir heuristic to Σ-protocols, as is done herein with seed incompressible functions. (This result from [1] casts doubt on the informal claim made in [6] that correlation intractability was insufficient for such constructions.) Yet, other than via the connection to correlation-intractability, their notion seems unrelated to seed-incompressibility. For example, we do not see a obvious way in which conditional entropy hashes imply collision-resistance, as we show seed-incompressibility does in Section 5.1.

Exposure-Resilient Functions. The notion that we investigate in this work can be seen as an enhancement of exposure-resilient functions (ERFs) as defined and constructed by Canetti et al. [5]. Recall that an ERF is a function whose output looks random even when some of the input bits are known. It is easy to see that if we restrict the seed compression attack to only output some of the bits of the seed, then a "seed incompressible PRF" can be constructed from an ERF and a standard PRF (by first applying the exposure-resilient function to the seed).

The Bounded-Retrieval Model. Our seed-compression attack model can also be seen as an instance of the "bounded-retrieval model" that was introduced by Dziembowski and by Di Crescenzo et al. [13, 11]. In this model, an adversary installs a virus on a target machine; the virus can observe all the secrets on the target machine, but only has a limited available bandwidth with which to communicate these secrets back to its "home base". The works of Dziembowski, Di Crescenzo et al. and Cash et al. [13, 11, 8] investigate obtaining secure key-exchange and authentication protocols in this model.

The current work can be thought of as trying to obtain primitives similar to pseudo-randomness in the same setting. (However, our focus is quite different, we view this model merely as a tool in order to establish primitives that can be used in other more standard models.)

Compressibility of NP Languages. A different notion of "compressibility" with applications to cryptography was recently proposed by Harnik and Naor [17]. In their notion, we are given an NP language and a word that is potentially in that language, and we try to produce a shorter word that is in the language if and only if the original is. For example, we are given a CNF formula ϕ and we try to compress it to a shorter ϕ' such that ϕ' is satisfiable if and only if ϕ is.[5] Harnik and Naor proved that if SAT is compressible then collision-resistant hashing can be constructed from one-way functions.

Our notion of compression seems quite different from the one of Harnik and Naor: roughly the difference is that they consider compressing the instance, whereas we are interested in compression of the witness (i.e., the secret seed of the function in our case).

[5] The length of ϕ' should be poly-logarithmic in the length of ϕ, but can be polynomial in the number of variables of ϕ.

2 Notations and CS Proofs

Notations. We define some notation used throughout the paper. Given a bit b, we use b^ℓ to denote the bit-string of ℓ bits b. Concatenation of bit strings is denoted with $||$. Given an ℓ bit-string $s = s_1, ..., s_\ell$ and $c \leq \ell$ we denote its first c significant bits $s_1, ..., s_c$ by $\lfloor s \rfloor_c$. We use $a \in_\mathcal{R} S$ to denote choosing uniformly at random an element a from a set S. We use $\mathsf{negl}(n)$ to denote some function f, such that for all c and sufficiently large n $f(n) \leq 1/n^c$ and $\mathrm{poly}(n)$ denotes some polynomial function $p \in \mathcal{O}(n^d)$ for some constant d.

CS Proofs. Our negative results use CS-proofs as constructed by Micali [20] (using techniques from Kilian [19]), as well as a variant of them due to Naor and Nissim [22]. Below, we briefly recall the definition. For our purposes, we view a CS-proof system as consisting of a prover, $\textsc{Prv}$, that wants to convince a verifier, $\textsc{Ver}$, of the validity of an assertion $x \in L$ where L is some NP-language and $\textsc{Prv}$ is in possession of a witness w for x.[6] In our context, we use non-interactive CS-proofs that work in the Random Oracle Model; that is, both the prover and verifier have access to a common random oracle. The prover generates an alleged proof that is examined by the verifier.

Definition 1 (Non-interactive CS proofs in the Random Oracle Model). *A CS-proof system for a language $L \in NP$ (with relation R_L), consists of two deterministic polynomial-time oracle machines, a prover* $\textsc{Prv}$ *and a verifier* $\textsc{Ver}$*, operating as follows:*

- *On input $(1^k, x, w)$ such that $(x, w) \in R_L$ and access to an oracle $\mathcal{O}$, the prover computes a proof $\pi = \textsc{Prv}^\mathcal{O}(1^k, x, w)$ such that $|\pi| \leq \mathrm{poly}(k, \log(|x| + |w|))$.*
- *On input $(1^k, x, \pi)$ and access to $\mathcal{O}$, the verifier decides whether to accept or reject the proof π* (i.e., $\textsc{Ver}^\mathcal{O}(1^k, x, \pi) \in \{\mathtt{accept}, \mathtt{reject}\}$).

The proof system satisfies the following conditions, where the probabilities are taken over the random choice of the oracle $\mathcal{O}$:

Perfect completeness: *For any $(x, w) \in R_L$ and for any k,*

$$\Pr_{\mathcal{O}}\left[\pi \leftarrow \textsc{Prv}^\mathcal{O}(1^k, x, w),\ \textsc{Ver}^\mathcal{O}(1^k, x, \pi) = \mathtt{accept}\right] = 1.$$

Computational soundness: *For any polynomial time oracle machine* $\textsc{Bad}$ *and any input $x \notin L$ it holds that*

$$\Pr_{\mathcal{O}}\left[\pi \leftarrow \textsc{Bad}^\mathcal{O}(1^k, x),\ \textsc{Ver}^\mathcal{O}(1^k, x, \pi) = \mathtt{accept}\right] \leq \mathsf{negl}(k).$$

We sometimes also require a stronger soundness condition by replacing the negligible function $\mathsf{negl}(k)$ with an exponentially small function $\frac{poly(k+|x|)}{2^k}$. (This stronger condition can still be proven in the random-oracle model.)

[6] Micali defined CS-proofs more generally, but we do not need this extra generality for our purposes.

3 Seed-Incompressible Pseudo-Random Functions

Following the intuition as presented in the introduction, we would have liked to have a construction $F = \{f_s\}$ such that f_s looks random even when given a "compressed version of s." Formalizing this takes some care, since this "compressed version of s" could be, for example, the first $|s|/2$ bits of $f_s(0)$ (which would make it easy to distinguish f_s from an unrelated random function). This technicality can be solved by borrowing the technique used by Coron et al. (in the context of domain extenders for random oracles) [9]. Namely, the second phase of the adversary gets either the compressed seed σ and access to $f_s(\cdot)$, or access to a random function $f(\cdot)$ and a "simulated compressed seed" that was generated by a simulator S^f (where S has access to the same random f).

In the formal definition below, we fix some polynomially-bounded length functions ℓ_1, ℓ_2 and consider function families from $\ell_1(n)$ bits to $\ell_2(n)$ bits with n-bit seeds. We denote by $\mathcal{F}_n$ the set of all functions $f : \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}$.

Definition 2 (Seed-Incompressible PRFs). *Let $\{F_n\}_{n\in\mathtt{N}}$ be a family of functions such that $F_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}$ can be efficiently computed, and denote $f_s(\cdot) \equiv F(s,\cdot)$.*

The family $\{F_n\}$ is pseudo-random under seed-compression attacks if for every two-phase efficient adversary $\mathtt{Adv} = (A, B)$ there exists an efficient simulator S and a negligible function negl such that

$$\left| \begin{array}{l} \Pr\left[s \in_{\mathcal{R}} \{0,1\}^n,\ \sigma \leftarrow A(s)\ :\ |\sigma| \le n/2 \textit{ and } B^{f_s}(\sigma) = 1\right] \\ -\ \Pr\left[f \in_{\mathcal{R}} \mathcal{F}_n,\ \sigma \leftarrow S^f(1^n)\ :\ |\sigma| \le n/2 \textit{ and } B^{f}(\sigma) = 1\right] \end{array} \right| \le \mathsf{negl}(n)$$

Seed-Incompressible PRFs would have been very useful, but unfortunately they do not exist, as will be shown below. While we show that SI-PRFs do not exist, a related concept will be introduced later, and therefore the discussion is useful for this later topic.

Theorem 1. *Seed-Incompressible PRFs as defined in Definition 2 do not exist.*

Proof. Let $\{F_n\}_{n\in\mathtt{N}}$ be a family of functions as in Definition 2. We show a two-phase adversary $\mathtt{Adv} = (A, B)$ for which no simulator exists. Fix some n and let $\ell_1 = \ell_1(n)$ and $\ell_2 = \ell_2(n)$. Let $j = \lceil 2n/\ell_2 \rceil$ (i.e., the output of f_s on $1, 2, \ldots, j$ contains at least $2n$ bits).

The first phase of the adversary, A, gets as input a seed $s \in \{0,1\}^n$. It computes $y_i = f_s(0||i)$ for $i = 1, 2, \ldots, j$, and then prepares a CS-proof π for the true NP statement

$$\text{"there exists a seed } s' \text{ such that } y_i = f_{s'}(0||i) \text{ for } i = 1, 2, \ldots, j\text{"} \qquad (\star)$$

The proof is prepared relative to the oracle $\mathcal{O}(\cdot) = f_s(1||\cdot)$ and security parameter $k = \sqrt{n}$. Then A outputs the proof π as the "compressed seed", to be used by the second phase B. (Notice that the length of this proof is $k\cdot\text{polylog}(n) < n/2$.)

The second phase B, on input π, first uses its oracle f to compute $y_i = f(0||i)$ for $i = 1, 2, \ldots, j$, thereby recovering the statement that π is supposed to be a CS-proof for. Then B attempts to verify the proof π relative to $f(1||\cdot)$ (where f is the provided oracle) and security parameter $k = \sqrt{n}$. It accepts if the proof is valid and rejects otherwise. By the perfect completeness of CS-proofs, B accepts with probability one when given the proof that A generated and access to the same f_s for which that proof was generated.

On the other hand, the soundness of CS-proofs implies that no simulator can make B accept with non-negligible probability. Indeed, when f is a random function in $\mathcal{F}_n$ then $y_1, \ldots, y_j$ consist of at least $2n$ random bits, hence the probability that the statement ($\star$) from above is true is at most 2^{-n}. And if the statement is not true, then no efficient simulator with access to a random f can generate a valid proof for it with probability better than $\text{poly}(n) \cdot 2^{-\Theta(k)} = \mathsf{negl}(n)$.

4 Seed-Incompressible Correlation Intractability

Canetti et al. [6] introduced the concept of Correlation Intractability to capture the intuition that the adversary cannot "hit" any rare input-output relation. Roughly, an *evasive relation* R is one where it is hard to find an input x such that $(x, f(x)) \in R$ for a random function f, and a function family F is *correlation intractable* if for any evasive relation R it is hard to find $(x, f(x)) \in R$ for a random member $f \in F$. These notions can be extended to $2p$-ary relations in the obvious way (see below).

Canetti et al. proved that correlation-intractable function families do not exist, in that an adversary that knows the short description of $f \in F$ can always find some $(x, f(x)) \in R_F$ for a particular relation R_F that depends on F. In our case, however, we are interested in an adversary that does not see the entire description of $f \in F$ but only gets a "compressed description". We provide the formal definitions below, and then discuss the extent to which the negative results from [6] and from Section 3 do or do not extend to this new notion. Below we again fix some polynomially bounded length functions ℓ_1, ℓ_2, and denote by $\mathcal{F}_n$ the set of all functions $f : \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}$.

Definition 3 (Evasive Relations). *A $2p$-ary relation R is evasive if for any efficient adversary A, there is a negligible function* negl *such that for all sufficiently large n*

$$\Pr_{f \in \mathcal{F}_n}[\langle x_1, \ldots, x_p\rangle \leftarrow A^f(1^n) \;:\; \langle x_1, \ldots, x_p, f(x_1), \ldots, f(x_p)\rangle \in R] \le \mathsf{negl}(n).$$

Sometimes we are interested only in *efficient* relations, namely relations R for which the membership problem $\langle x_1, \ldots, x_p, y_1, \ldots, y_p\rangle \overset{?}{\in} R$ can be efficiently decided (i.e., in polynomial time).

Definition 4 (Seed-Incompressible Correlation Intractability). *Let $\{F_n\}_{n\in\mathbb{N}}$ be a family of functions where $F_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}$ can be efficiently computed, and denote $f_s(\cdot) \equiv F(s, \cdot)$.*

For some polynomial $p = p(n)$, we say that the family $\{F_n\}$ is correlation intractable under seed-compression attacks with respect to $2p$-ary relations if for every $2p$-ary evasive relation R, and for every two-phase efficient adversary $\mathtt{Adv} = (A, B)$*, there is a negligible function* negl *such that for all sufficiently large n*

$$\Pr\left[\begin{array}{l} s \in_{\mathcal{R}} \{0,1\}^n,\ \sigma \leftarrow A(s),\ \ \langle x_1,\ldots,x_p\rangle \leftarrow B^{f_s}(\sigma)\ : \\ |\sigma| \le n/2 \textit{ and } \langle x_1,\ldots,x_p, f_s(x_1),\ldots,f_s(x_p)\rangle \in R \end{array}\right] \le \mathsf{negl}(n).$$

We also call such function families seed-incompressible correlation-intractable (with respect to $2p$-ary relations)*, or SI-CorInt($2p$), for short.*

In the case that we restrict the above quantification on all evasive relations to only efficient *evasive relations, we say that the family $\{F_n\}$ is* weakly *seed-incompressible correlation-intractable with respect to $2p$-ary relations (wSI-CorInt($2p$)).*

4.1 Do SI Correlation Intractable Functions Exist?

The first question to answer with respect to the seed-incompressible correlation intractability as defined above is whether we can extend the impossibility result from Theorem 1 (or from [6]) to show that it too cannot be realized.

One first observes that for some setting of parameters, an attacker in the seed-compression model is just as powerful as an attacker that has the full uncompressed seed. Specifically, if the seed is more than $2p$ times the length of the input to h, then the first phase of an attacker in the seed-compression model can output the vector that breaks the correlation intractability as the "compressed seed". However, the impossibility results from [6] do not extend to very long seeds, so this simple observation does not appear to shed new light on the existence of SI-CorInt functions. Below we show, however, that the technique from Theorem 1 can be extended for some settings of parameters:

As opposed to the case of Theorem 1, here the adversary needs not only to distinguish f_s from random (which can be done with CS-proofs), but also to compute some "unpredictable relation". The idea that we exploit here is that the CS-proof itself can be thought of as an "unpredictable relation." Roughly, we have a relation of the form

$$\left\{\begin{array}{l} \langle(1,\ldots,t,v_1,\ldots,v_m),(x_1,\ldots,x_t,y_1,\ldots,y_m)\rangle\ \ : \\ \qquad\qquad\qquad \text{The CS-proof } (v_1,\ldots,v_m) \text{ is valid for the instance } (x_1 = f(1), \\ \qquad\qquad\qquad \ldots, x_t = f(t)) \text{ w.r.t. } V \text{ receiving oracle answers } (y_1,\ldots,y_m). \end{array}\right\}$$

Tracing through the various parameters we see that to use Micali's construction for CS-proofs with a relation such as above we need $t = (n+\omega(\log n))/\ell_2(n)$ and $m = \mathit{polylog}(n)$. Hence we get an impossibility result for $2p$-ary relations where $p = n/\ell_2(n) + \mathit{polylog}(n)$. Moreover, if we assume the existence of collision-resistant hashing then we can use the variant of CS-proofs with few oracle calls due to Naor and Nissim [22], and then we can get by with a relation that only depends on $m = O(n/\ell_1(n))$ of the v_i's. Hence for function families with n-bit seeds and input/output bit lengths of size $\ell_1(n), \ell_2(n) = \Omega(n)$ we also obtain an impossibility result for $2p$-ary relations where $p = O(1)$ (we can get as low as $p = 3$ when $\ell_1(n), \ell_2(n) > n$.)

Lemma 1. *An efficiently computable functions family $\{F_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}\}$ cannot be weakly correlation intractable under seed-compression attacks with respect to $2p$-ary relations, for any $p \geq \lceil (n + \omega(\log n))/\ell_2(n) \rceil + \lceil polylog(n)/\ell_2(n) \rceil$.*

Moreover, if collision resistant hash function family $\{H_n : \{0,1\}^n \times \{0,1\}^{\ell_4(n)} \to \{0,1\}^{\ell_5(n)}\}$ exist, for polynomials $\ell_4(n) > \ell_5(n)$, then no family as above can be correlation intractable under seed-compression attacks with respect to $2p$-ary relations where $p \geq \lceil (n + \omega(\log n))/\ell_2(n) \rceil + \lceil n^\epsilon/\ell_2(n) \rceil + \lceil n^\epsilon/\ell_1(n) \rceil$ (for any $\epsilon > 0$).

A proof of this lemma will be in the full version of this paper.

Smaller relations. We speculate that current techniques cannot be used to rule out relations with arity less than 6. This is because with the current technique of using CS-proofs, you would need at least one oracle call to specify the function instance, at least one oracle call for the CS-proof, and you would have to use at least one more v_i to describe the CS-proof itself. Thus it is still plausible that seed-incompressible correlation intractable function families exist with respect to such low-arity relations.

5 Implications of Seed-Incompressible Correlation Intractability

We demonstrate the usefulness of seed-incompressible functions by showing how they can be used to easily construct two primitives: specifically collision-resistant hash functions and (single-theorem) NIZK systems via the Fiat-Shamir methodology. More generally, the primitives that can be constructed from seed-incompressible functions are those for which a "break" can be encoded with only a few bits. (For example, one can encode a collision in a hash function using only two inputs to the function. Similarly, for a NIZK that was derived from a 3-move Σ-protocol, one can encode a false proof using only the two messages that the prover sends.) When constructing such primitives from seed-incompressible functions, we let the seed of the function be sufficiently longer than the number of bits that are needed to encode a break, and then any adversary that breaks the resulting primitive can be converted into a "compressor" (that given the seed outputs a break), thus violating the seed-incompressibility of the underlying function.

5.1 Collision Resistant Hashing

We show that seed-incompressible correlation-intractable function with respect to quaternary relations must be "essentially collision-resistant." We view this feature as a minimal requirement for any primitive that one hopes to use in lieu of a random-oracle, since heuristic implementations of random-oracle-based constructions always use collision-resistant hash functions such as SHA to replace the oracle. Note that it is not true that any seed-incompressible function is also collision-resistant (for example, seed-incompressible functions need not be length decreasing). Rather, we show below that any seed-incompressible function must have "embedded in it" a collision-resistant function.

Specifically, given a seed-incompressible correlation-intractable function $f_s(\cdot)$, we consider shortening the inputs and outputs of f_s so that the inputs are shorter than one quarter of the seed and the outputs are shorter than the inputs. We then observe that an algorithm that finds collisions in the resulting (length-decreasing) function can be (trivially) converted to a "compressor" that breaks the seed-incompressibility of f_s: the "compressor" only needs to output the collision.

We formally state the definition of collision-resistance for completeness and then state the theorem with proof.

Definition 5 (Collision Resistant Hash Functions (CRHF)). *Fix polynomially-bounded length functions $\ell_2(n) < \ell_1(n)$. A function generator $\{H_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \rightarrow \{0,1\}^{\ell_2(n)}\}_{n\in\mathtt{N}}$ is collision resistant if for every probabilistic polynomial time adversary A there is a negligible functions negl such that for all sufficiently large n:*

$$\Pr[s \leftarrow \{0,1\}^n, (x_1, x_2) \leftarrow A(s) \mid x_1 \neq x_2 \wedge h_s(x_1) = h_s(x_2)] \leq \mathsf{negl}(n).$$

Theorem 2. *If there exists a function family $\{F_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \rightarrow \{0,1\}^{\ell_2(n)}\}$ with super-logarithmic length functions $\ell_1, \ell_2 = \omega(\log n)$, which is correlation intractable under seed-compression attacks with respect to quaternary relations, then collision resistant hash functions exist.*

Proof. Let ℓ_1, ℓ_2 be super-logarithmic length functions, $\ell_1, \ell_2 = \omega(\log n)$, and assume that a family $F = \{F_n : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \rightarrow \{0,1\}^{\ell_2(n)}\}_{n\in\mathtt{N}}$ is correlation intractable under seed-compression attacks with respect to quaternary relations.

Consider a modification of the family F to operate on potentially shorter inputs and outputs. Namely, let $\ell_1' = \min(\ell_1, n/4)$ and $\ell_2' = \min(\ell_2, \ell_1' - 1)$, and consider the family $H = \{H_n : \{0,1\}^n \times \{0,1\}^{\ell_1'(n)} \rightarrow \{0,1\}^{\ell_2'(n)}\}_{n\in\mathtt{N}}$ which if defined as follows: on seed s of length n and input x' of length ℓ_1', first append zeros to x' up to length of ℓ_1 bits, then apply $F(s, \cdot)$ to the result, and finally take only the first ℓ_2' bits of the outcome.

$$H(s, x') \stackrel{\text{def}}{=} \lfloor F(s, x) \rfloor_{\ell_2'}, \text{ where } x = x' || 0^{\ell_1(|s|) - \ell_1'(|s|)}.$$

We prove that if F is SI-CorInt with respect to quaternary relations then H is collision-resistant. In particular, consider the relation $R \subset \{0,1\}^{\ell_1} \times \{0,1\}^{\ell_1} \times \{0,1\}^{\ell_2} \times \{0,1\}^{\ell_2}$:

$$R \stackrel{\text{def}}{=} \{(x_1, x_2, y_1, y_2) \mid x_1 \neq x_2 \text{ and } \lfloor y_1 \rfloor_{\ell_2'} = \lfloor y_2 \rfloor_{\ell_2'}\}.$$

This relation is clearly evasive, as every polynomial-tie adversary has probability at most $poly(n) \cdot 2^{-\ell'_2} = \mathsf{negl}(n)$ of outputting x_1, x_2 for which $(x_1, x_2, f(x_1), f(x_2)) \in R$ where f is a random function (since ℓ'_2 is super-logarithmic).

Assume for contradiction that the family H from above is not collision resistant, and let C be a collision finding adversary that given a random $s \in \{0,1\}^n$ outputs a pair of strings $x_1, x_2 \in \{0,1\}^{\ell'_1}$ such that $H(s, x_1) = H(s, x_2)$ with probability at least $1/n^c$. Then, define the adversary $\mathtt{Adv} = (A, B)$ for the underlying SI-CorInt family as follows: The "compressor" $A(s)$ simply outputs the collision $(x_1, x_2) = C(s)$, and note that $|A(s)| = n/2$. The second phase of the attack is just translate H-inputs into F-inputs by appending zeros, namely $B(x_1, x_2)$ outputs (x'_1, x'_2) where $x'_i = x_i || 0^{\ell_1 - \ell'_1}$. By definition if $H(s, x_1) = H(s, x_2)$ (which happens with non-negligible probability) then $\lfloor F(s, x'_1) \rfloor_{\ell'_2} = \lfloor F(s, x'_2) \rfloor_{\ell'_2}$, and therefore $(x'_1, x'_2, F(s, x'_1), F(s, x'_2)) \in R$, contradicting the security of F.

5.2 From Σ-Protocols to NIZK Arguments

One of the "signature uses" of the random-oracle heuristic is to remove interaction from zero-knowledge protocols using the Fiat-Shamir heuristic. Specifically, given a public-coin honest verifier zero-knowledge proof system (known as Σ-protocols in the case where the number of rounds is three), it is possible to transform it into a non-interactive protocol by replacing the verifier's messages with the output of a "cryptographic hash function", applied to the transcript up to that point.

It is well known that if the original protocol has negligible soundness error, then the resulting non-interactive protocol can be proven in the random-oracle model to be a non-interactive zero-knowledge argument system, and can also be used as a secure signature scheme. On the other hand, Goldwasser and Kalai proved in [16] that there exist interactive Σ-protocols with negligible soundness for which their resulting protocols are not secure signature schemes in the standard model, no matter what function family is used to replace their interaction.

However, that negative result still leaves the possibility of a function family that will convert the interactive protocol into a NIZK argument system for a single theorem. Indeed, below we show that the latter is possible if seed-incompressible correlation-intractable functions exist.

We begin by recalling the definitions of Σ-protocols and NIZK argument systems. Below let L be an NP-language and let R_L be a binary relation that defines L, namely $L = \{x \ : \ \exists w \ s.t. \ (x, w) \in R_L\}$ (where the witness w has length polynomial in $|x|$).

Σ-Protocols. For a pair (P, V) of interacting protocols, we denote by $(P, V)(x, w)$ a run in which P has input $(x, w) \in R_L$, V has input x, and P attempts to convince V of the validity of the assertion $x \in L$. For a three-move protocols as above (with P going first), denote by α, β, and γ the three messages that are exchanged in the protocol, and let P_1, and P_2 be the randomized functions that the prescribed prover uses to compute its two messages, namely we have

$$(\alpha, \text{state}) \leftarrow P_1(x, w), \text{ and } \gamma \leftarrow P_2(x, w, \text{state}, \alpha, \beta).$$

We also denote by V^* the function that the verifier employs to decide whether to accept or reject the proof,

$$V^*(x, \alpha, \beta, \gamma) \in \{\text{accept,reject}\}.$$

Typically, the first flow α is called a *commitment*, the second flow β is called a *challenge*, and the third flow γ is called a *response*.

Definition 6. *A 3-move protocol (P, V) as above is a Σ protocol for a language L if it satisfies the following properties:*

Public-coin verifier. *The message β sent by V is always a sequence of $t(|x|)$ uniformly chosen random bits (for some length function t).*

Perfect completeness. *For every $(x, w) \in R$:*

$$\Pr\left[\begin{array}{c}(\alpha, state) \leftarrow P_1(x, w);\ \beta \in_{\mathcal{R}} \{0,1\}^{t(|x|)};\ \gamma \leftarrow P_2(x, w, state, \alpha, \beta)\ : \\ V^*(x, \alpha, \beta, \gamma) = accept\end{array}\right] = 1,$$

where the probability is over the random choices of P_1, P_2 and β.

Soundness. *There is a negligible function* negl *such that for every $x \notin L_R$, for every pair of adversarial (computationally unlimited) prover circuits P_1^*, P_2^*:*

$$\Pr\left[\alpha \leftarrow P_1^*(x);\ \beta \in_{\mathcal{R}} \{0,1\}^{t(|x|)};\ \gamma \leftarrow P_2^*(x, \beta) : V^*(x, \alpha, \beta, \gamma) = acpt\right] = \frac{1}{2^{t(|x|)}},$$

where the probability is over the random choice of β. [7] *We note a property of Σ-protocols of interest to our later arguments: For every Σ-protocol, simple parallel repetition and padding arguments allow the length $t(|x|)$ of the verifier's challenge to be set to an arbitrary positive integer.* [8] *For the remainder of the paper we therefore assume that $t \in \omega(\log n)$, and that the adversaries probability of success, as stated above, is negligible in $|x|$.*

Zero-knowledge. *There exists a polynomial-time simulator S that on input x and $\beta' \in \{0,1\}^{t(|x|)}$ outputs $(\alpha', \beta', \gamma')$ (we have S outputting its input β' to simplify the notations somewhat).*
We require that for every $(x, w) \in R_L$ and every $\beta' \in \{0,1\}^{t(|x|)}$, the distribution on $(\alpha', \beta', \gamma') = S(x, \beta')$ is identical to the conditional distribution on the transcript $(\alpha, \beta, \gamma) = (P, V)(x, w)$, when conditioned on $\beta = \beta'$.

---

[7] Traditionally Σ-protocols are defined with a stronger soundness condition called extractability that clearly implies the current soundness definition.

[8] We refer the reader to [10] for a complete discussion on this and other properties of Σ-protocols.

NIZK Arguments We remind the reader that the common reference string (CRS) model is one in which all participants and the adversary have access to a polynomial sized common reference string chosen by a trusted third party from a pre-specified distribution (which we denote $\mathcal{D}$).

Definition 7. *A pair of efficient probabilistic algorithms (P, V) are a single-theorem NIZK argument system for a language L (specified by a binary relation R_L) in the CRS model, if it satisfies:*

Completeness. $\forall (x, w) \in R$, $\Pr_{\mathsf{crs},P,V}[\sigma \leftarrow P(\mathsf{crs}, x, w) : V(\mathsf{crs}, x, \sigma) = 1] = 1$.

Computational Soundness. For every (possibly cheating) efficient probabilistic prover P^* there is a negligible function negl such that for all $x \notin L$:

$$\Pr_{\mathsf{crs},P^*,V}[\sigma \leftarrow P^*(\mathsf{crs}, x) : V(\mathsf{crs}, x, \sigma) = 1] \leq \mathsf{negl}(|x|).$$

Zero-Knowledge. There exists an efficient simulator S such that for every $(x, w) \in R$, the output of the following two experiments are (computationally, statistically, perfectly)-indistinguishable.

$\mathsf{Exp}_1(x, w)$	$\mathsf{Exp}_2(x)$
$\mathsf{crs} \leftarrow \mathcal{D}$	$(\mathsf{crs}', \sigma') \leftarrow S(x)$
$\sigma \leftarrow P(\mathsf{crs}, x, w)$	
Output (crs, σ)	Output (crs', σ')

The Fiat-Shamir Transformation Fiat-and Shamir described in [14] a transformations that turns Σ-protocols into non-interactive argument systems. Specifically, instead of having the verifier choose a random challenge β, one computes β by applying a hash function to the input x and the commitment α, setting $\beta = f(x, \alpha)$. The non-interactive proof σ then consists of the elements α, γ of the Σ-protocol (i.e., the commitment and the response). Given the input x and (α, γ), the verifier computes $\beta = f(x, \alpha)$ and checks that $V^*(x, \alpha, \beta, \gamma)$ =accept. It is easy to show that when the hash function f is modeled as a random oracle then the resulting protocol is still computationally sound (since the challenge β is still a string of random bits that the adversary cannot control, other than attempting to select a polynomial number of them). Moreover, if the simulator can program the random oracle then this protocol also remains zero-knowledge.

We next show that using SI-CorInt families (with respect to quaternary relations), we can construct function families for which the Fiat-Shamir transformation yields a single-theorem NIZK argument system in the CRS model.

- Let (P, V) be a Σ-protocol in which the commitment, challenge, and response are of lengths $|\alpha| = t_1(|x|)$, $|\beta| = t_2(|x|)$, and $|\gamma| = t_3(|x|)$, respectively.
- Also, let $\{F : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}\}_{n \in \mathbb{N}}$ be a function family, where ℓ_1, ℓ_2 are polynomially bounded from above and below. (That is $n^{1/c} \leq \ell_1(n), \ell_2(n) \leq n^c$ for some constant $c > 1$ and every sufficiently large n).

For inputs of length $|x| = m$, we choose the security parameter n (which defined the seed-length for F) as $n = \max\{\ 2(m + t_1(m) + t_3(m)),\ \ell_1^{-1}(m + t_1(m)),\ \ell_2^{-1}(t_2(m))\ \}$. Namely, n is chosen large enough so that $m + t_1(m) + t_3(m) \le n/2$ and also $\ell_1(n) \ge m + t_1(m)$ and $\ell_2(n) \ge t_2(m)$. Note that since ℓ_1, ℓ_2 and the t_i's are polynomially-bounded, then n is polynomial in m. Below we view n, ℓ_1, ℓ_2 as functions of the input length m. We then reset the input and output length to be exactly $\ell_1' = m + t_1(m)$ and $\ell_2' = t_2(m)$ by setting

$$F'(s, x') \stackrel{\text{def}}{=} \lfloor F(s, x) \rfloor_{\ell_2'}\,, \text{ where } x = x' || 0^{\ell_1 - \ell_1'}$$

Finally we define $H : \{0,1\}^{n+\ell_2'} \times \{0,1\}^{\ell_1'} \to \{0,1\}^{\ell_2'}$ as

$$H(\langle s, z\rangle, x') \stackrel{\text{def}}{=} z \oplus F'(s, x') \;=\; z \oplus \left\lfloor F\left(s,\ x' || 0^{\ell_1 - \ell_1'}\right) \right\rfloor_{\ell_2'} \qquad (1)$$

We are now ready to describe the NIZK argument system. The CRS consists of a pair (s, z) where $s \in \{0,1\}^n$ is a seed for the underlying function F and z is a random string of length $\ell_2' = t_2(m)$ (so together $\langle s, z\rangle$ are a seed for the function H from Eq. (1)). The NIZK argument system is obtained by applying the Fiat-Shamir transformation to the original Σ-protocol (P, V) using the function $H_{s,z}$.

Namely, on input $(x, w) \in R_L$ with $|x| = m$ and $\mathsf{crs} = (s, z)$, the prover sets $(\alpha, \text{state}) = P_1(x, w)$, $\beta = H_{s,z}(x, \alpha)$ and $\gamma = P_2(x, w, \text{state}, \alpha, \beta)$. The proof is the string $\sigma = (\alpha, \gamma)$. Given x, $\mathsf{crs} = (s, z)$, and the proof $\sigma = (\alpha, \gamma)$, the verifier computes $\beta = H_{s,z}(x, \alpha)$ and checks that $V^*(x, \alpha, \beta, \gamma) =$accept.

Theorem 3. *Let (P, V) be a three round Σ-protocol for the language L, defined by the NP relation R_L, and let $F = \{F : \{0,1\}^n \times \{0,1\}^{\ell_1(n)} \to \{0,1\}^{\ell_2(n)}\}_{n\in\mathbb{N}}$ be a function family with polynomially-bounded length functions.*

If (P, V) has a negligible soundness error and F is correlation intractable under seed-compression attacks with respect to quaternary relations, then applying the Fiat-Shamir transformation to (P, V) using the function family H from Eq. (1) yields a single theorem NIZK argument system for L in the CRS model.

Proof. The perfect completeness of the resulting NIZK system follows immediately from the perfect completeness of the Σ-protocol.

For the zero-knowledge property, the simulator S^* for the NIZK system uses the simulator S given by the Σ-protocol. It first chooses a random value $\beta' \in \{0,1\}^{t_2(|x|)}$ and uses S to compute $S(z) = (\alpha', \beta', \gamma')$. It then chooses a random seed $s \in \{0,1\}^n$ for the function F and computes

$$z = \beta' \oplus F'(s, \langle x, \alpha'\rangle) \;=\; \beta' \oplus \left\lfloor F\left(s,\ \ \langle x, \alpha'\rangle || 0^{\ell_1 - \ell_1'}\right) \right\rfloor_{\ell_2'}$$

Note that by definition, we have $H(\langle s, z\rangle,\ \ \langle x, \alpha'\rangle) = \beta'$. The simulator S^* outputs the CRS $\langle s, z\rangle$ and proof $\langle \alpha', \gamma'\rangle$. Clearly, the distribution on the output of S^* is identical to the distribution on the real pairs of CRS and proof.

It is left to prove computational soundness. Suppose for contradiction that there existed an efficient cheating prover $\widehat{P}'$ such that for a constant $c > 0$ and infinitely many $x \notin L$ it holds that:

$$\Pr_{s,z}\left[\sigma \leftarrow \widehat{P}'((s,z),x) : \widehat{V}((s,z),x,\sigma) = \text{accept}\right] \geq |x|^{-c}. \tag{2}$$

We then show a *non-uniform* seed-compression adversary $\texttt{Adv} = (A, B)$ that breaks the correlation-intractability of the underlying family F.

For each $x \notin L$, denote by $z(x)$ the auxiliary string z that maximizes the success probability of $\widehat{P}'$. That is,

$$z(x) = \operatorname*{argmax}_{z}\left\{\Pr_{s}\left[\sigma \leftarrow \widehat{P}'((s,z),x) : \widehat{V}((s,z),x,\sigma) = \text{accept}\right]\right\}.$$

An easy averaging argument implies that whenever Eq. (2) holds:

$$\Pr_{s}\left[\sigma \leftarrow \widehat{P}'((s,z(x)),x) : \widehat{V}((s,z(x)),x,\sigma) = \text{accept}\right] \geq |x|^{-c}.$$

On the other hand, for any $x \notin L$ the relation

$$\hat{R}_x = \left\{ (x_1,x_2,y_1,y_2) \;\middle|\; \begin{array}{l} x_1 = \left(\langle x,\alpha\rangle \,||0^{\ell_1-\ell_1'}\right) \text{ and} \\ V^*\left(x,\alpha,\left(z(x)\oplus \lfloor y_1\rfloor_{\ell_2'}\right),x_2\right) = \text{accept} \end{array} \right\}$$

is evasive by the soundness of the original Σ-protocol. (Note, that $z(x)$ is a constant in this relation, so XOR-ing it to $\lfloor y_1\rfloor_{\ell_2'} = \lfloor f(x_1)\rfloor_{\ell_2'}$ has no effect on soundness when f is a random function.) It follows that for our evasive relations R_x $\hat{R} = \bigcup_{x\notin L}\hat{R}_x$ is also an evasive relation.

Since our choices of parameters imply that $|x|+|\alpha|+|\gamma| \leq n/2$, then we can use $\widehat{P}'$ to construct a seed-compression attacker $\texttt{Adv} = (A, B)$. The first part A gets as advice string the values $x, z(x)$ for some $x \notin L$ for which Eq. (2) holds, as well as the seed s for F. It uses $\widehat{P}'$ to compute α, γ, and outputs (x, α, γ) as the "compressed seed", and indeed the length of this "compressed seed" is at most $n/2$. The second part B outputs $\langle x,\alpha\rangle$ and γ, and indeed we have by definition $(\langle x,\alpha\rangle, \gamma, F(s,\langle x,\alpha\rangle), F(s,\gamma)) \in \hat{R}$.

5.3 Non-Uniformity in the Proof of The Fiat-Shamir Transform

We comment that the proof of Theorem 3 seems inherently non-uniform. We use non-uniformity in two places: one is to select $x \notin L$ for which $\widehat{P}'$ has good success probability, and the other to select the auxiliary $z(x)$. The first use can be eliminated by switching to a uniform soundness condition on the underlying Σ-protocol (i.e., when a uniform cheating prover needs to output some $x \notin L$ together with a convincing proof for it). The latter use of non-uniformity seems harder to eliminate, however. Maybe this can be done by switching to 6-ary relations and setting $z = F(s, 0)$, so we get

$$\hat{R}' = \left\{ (x_1,x_2,x_3,y_1,y_2,y_3) \;\middle|\; \begin{array}{l} x_1 = 0,\; x_2 = \langle x,\alpha\rangle,\; x_3 = \gamma, \\ x \notin L \text{ and } V^*\left(x,\;\alpha,\;\lfloor y_1\oplus y_2\rfloor_{\ell_2'},\;\gamma\right) = \text{accept} \end{array} \right\}$$

It is not hard to see that this $\hat{R}'$ is evasive, but it is not clear how to translate the success of $\widehat{P}'$ in breaking the soundness (when z is chosen at random) to success against this $\hat{R}'$ (when z is set as $z = F(s, 0)$ for a random s).

6 Future Work and Open Problems

The most intriguing open question that results from this work is whether seed-incompressible correlation-intractable functions can be constructed under more traditional computational assumptions. Given that their existence implies the existence of NIZK protocols without the aid of any apparent trapdoor feature, such a construction will likely need substantial insight. Alternately, and just as interesting, would be an argument showing that knowledge of small numbers of key-bits really does allow one to say something meaningful about composition based block-cipher and hash-function designs.

In a slightly orthogonal direction, the impossibility results presented in this paper are derived through proving exactly the one property of the function generators that we know the adversary has direct knowledge of: the function is computed by a small circuit. Any definition along the lines of seed-incompressibility that also managed to circumvent this problem would be interesting.

Finally, it would be nice if one could show that the OAEP scheme proposed by Bellare and Rogaway[3] (or a close relative of it) could be proven secure under such an assumption, as it is this random-oracle protocol that is probably used in practice on the most frequent basis (due to its inclusion in the TLS protocol [12] for secure web transactions), and thus further evidence of its security would be heartening.

References

1. B. Barak, Y. Lindell and S. Vadhan. Lower Bounds for Non-Black-Box Zero-Knowledge. In the Journal of Computer and System Sciences, 72(2), pages 321–391, 2006 (JCSS FOCS 2003 Special Issue).
2. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *1st Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.
3. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. De Santis, editor, *Advances in Cryptology- EUROCRYPT'94*, pages 92–111. Springer, Berlin,, 1994.
4. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology - CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer-Verlag, 1997.
5. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-Resilient Functions and All-or-Nothing Transforms. In *Advances in Cryptology – EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469. Springer, 2000.
6. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004. Preliminary version in STOC'98, Pages 209-218.

7. R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hashing. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 131–140, Dallas, TX, May 1998. ACM.
8. D. Cash, Y. Z. Ding, Y. Dodis, W. Lee, R. J. Lipton, and S. Walfish. Intrusion-Resilient Key Exchange in the Bounded Retrieval Model. In *4th Theory of Cryptography Conference (TCC'07)*, volume 4392 of *Lecture Notes in Computer Science*, pages 479–498. Springer, 2007.
9. J. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgrd Revisited: How to Construct a Hash Function. In *Advances in Cryptology – CRYPTO '04*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.
10. I. Damgard. On Σ-Protocols. Lecture notes for Cryptologic Protocol Theory course, Aarhus University, 2005 , `http://www.daimi.au.dk/%7Eivan/Sigma.pdf`.
11. G. DiCrescenzo, R. J. Lipton, and S. Walfish. Perfectly Secure Password Protocols in the Bounded Retrieval Model. In *3rd Theory of Cryptography Conference (TCC'06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 225–244. Springer, 2006.
12. T. Dierks and C. Allen. RFC2246:The TLS Protocol. RFC 2246, The Internet Society, Network Working Group, 1999.
13. S. Dziembowski. Intrusion-Resilience Via the Bounded-Storage Model. In *3rd Theory of Cryptography Conference (TCC'06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 207–224. Springer, 2006.
14. A. Fiat and A. Shamir. How to prove yourself. practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–189. Springer-Verlag, 1986.
15. O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, 1998.
16. S. Goldwasser and Y. T. Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *44th Symposium on Foundations of Computer Science (FOCS '03)*, pages 102–115. IEEE, 2003.
17. D. Harnik and M. Naor. On the Compressibility of NP Instances and Cryptographic Applications. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 719–728. IEEE Computer Society, 2006.
18. R. Impagliazzo and S. Rudich, Limits on the provable consequences of one-way permutations. Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pages 44–61, ACM Press, 1989.
19. J. Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, pages 723–732. ACM, May 1992.
20. S. Micali. CS proofs. In *35th Annual Symposium on Foundations of Computer Science ("FOCS'94")*, pages 436–453. IEEE, 1994.
21. S. Micali. Computationally Sound Proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.
22. M. Naor and K. Nissim. Computationally sound proofs: Reducing the number of random oracle calls. manuscript, 1999.