Extractable Perfectly One-way Functions

Ran Canetti[1]⋆ and Ronny Ramzi Dakdouk[2]⋆⋆

[1] IBM T. J. Watson Research Center, Hawthorne, NY.
canetti@watson.ibm.com
[2] Yale University, New Haven, CT.
dakdouk@cs.yale.edu

Abstract. We propose a new cryptographic primitive, called **extractable perfectly one-way (EPOW) functions**. Like perfectly one-way (POW) functions, EPOW functions are probabilistic functions that reveal no information about their input, other than the ability to verify guesses. In addition, an EPOW function, f, guarantees that any party that manages to compute a value in the range of f "knows" a corresponding preimage.

We capture "knowledge of preimage" by way of algorithmic extraction. We formulate two main variants of extractability, namely non-interactive and interactive. The noninteractive variant (i.e., the variant that requires non-interactive extraction) can be regarded as a generalization from specific knowledge assumptions to a notion that is formulated in general computational terms. Indeed, we show how to realize it under several different assumptions. The interactive-extraction variant can be realized from certain POW functions.

We demonstrate the usefulness of the new primitive in two quite different settings. First, we show how EPOW functions can be used to capture, in the standard model, the "knowledge of queries" property that is so useful in the Random Oracle (RO) model. Specifically, we show how to convert a class of CCA2-secure encryption schemes in the RO model to concrete ones by simply replacing the Random Oracle with an EPOW function, without much change in the logic of the original proof. Second, we show how EPOW functions can be used to construct 3-round ZK arguments of knowledge and membership, using weaker knowledge assumptions than the corresponding results due to Hada and Tanaka (Crypto 1998) and Lepinski (M.S. Thesis, 2004). This also opens the door for constructing 3-round ZK arguments based on other assumptions.

1 Introduction

The Random Oracle methodology [15, 4] consists of two steps. The first step involves designing a protocol and proving security in an idealized model called the Random Oracle (RO) model. In the RO model, all parties have oracle access to a public random function, O. The oracle answers are uniform and independent with only one constraint, specifically, that all answers to the same query are identical. The second step involves "moving" the protocol from this idealized model to the real world. This is done by "replacing" the RO with a cryptographic hash function such as SHA1 [16] or MD5 [26]. In other words, every oracle call is replaced by a function call to some publicly

⋆ Supported by NSF grant CFF-0635297 and US-Israel Binational Science Foundation Grant 2006317.

⋆⋆ Work supported by NSF grant #0331548.

known cryptographic hash function. This transformation is known as an instantiation of Random Oracles.

Although the first step of the RO methodology is rigorous, the second step remains a heuristic for the most part. While most results in this area provide proofs in the RO model, they lack even informal justification as to why the instantiated protocols may be secure. Such justification is of dire need given the fact that the RO methodology is not sound in general. Specifically, it was shown that there are schemes secure in the RO model without any secure instantiations [9, 24, 17]. Furthermore, there exist natural primitives that are realizable in the RO model but can not be realized at all in the standard model, regardless of the computational assumptions used [25].

Given the general impossibility results mentioned above, one may resort to considering a proof in the RO model as a "stepping stone" towards a proof in the standard model. However, there is a severe flaw with this point of view: When it comes to security properties, proofs in the RO model use the Random Oracle somewhat like a Swiss Army knife. Random Oracles satisfy many cryptographic properties including collision resistance (it is hard to find two queries with the same RO answer), uniformity (the answer to any query is uniformly distributed), unpredictability or correlation intractability [9], programmability [25] and knowledge of queries (any machine that computes $O(q)$ "knows" q). Furthermore, works that use the RO methodology do not often highlight the specific properties of Random Oracles that are used or needed for the current proof. This makes translating a proof from the RO model to the standard model a harder task. And indeed, proofs in the RO model usually follow different lines from the corresponding ones in the standard model. This is contrary to the intuition behind the RO methodology, which is to use the randomness in the RO model to come up with simple proofs and then replace the Random Oracle by an appropriate function while maintaining the overall proof structure.

In light of the above discussion, it is interesting to identify specific properties of Random Oracles that are essential for the security of specific protocols. Once these properties are identified, it may then be possible to capture them with concrete functions that can be used to replace Random Oracles. Such an approach motivated the introduction of perfectly one-way (POW) functions in [7] as functions that capture the hiding property of Random Oracles and are then used to instantiate Random Oracles in a semantically-secure encryption scheme.[3] In another attempt, Boldyreva and Fischlin [6] introduce a strong variant of pseudorandom generators geared towards instantiating OAEP.

However, attempts at direct instantiation of encryption schemes secure against chosen ciphertext attacks (IND-CCA2) in the RO model have failed. It seems that one main problem is to translate a central property of Random Oracles, namely knowledge of queries, to the standard model. This property proves essential for the security proof in the RO model but it has not been previously formalized and captured by concrete functions.

1.1 Our Work

We formalize the "knowledge of queries" property mentioned above and cast it on a concrete object in the standard model. We call the new object an extractable perfectly

[3] Informally, POW functions are probabilistic functions that hide all partial information about the input.

one-way (EPOW) function. Then, we use EPOW functions not only to instantiate such schemes but also use a proof of security that follows similar logic as the original proof. The intended goal in this instantiation is not to try to achieve a more efficient construction than the existing ones in the literature but rather identify and realize the needed properties of the random oracle so that the proof of security remains the same in the standard model in both its logic and simplicity. In addition, we show that EPOW functions are useful in other contexts. We go into more detail shortly.

Extractable perfectly one-way functions. In the RO model, the knowledge of queries property means that any machine that computes an RO answer, $O(q)$, "knows" q. Even though such a property is easy to formalize and satisfy in the RO model if the range of O is sparse, defining it in the standard model while maintaining hiding properties is tricky. Towards this end, we build on the notion of perfect one-wayness presented in [7] to introduce a new class of functions called **extractable perfectly one-way** (EPOW) functions. These are functions that hide all information about the input but any machine that computes a valid image, "knows" a corresponding preimage. We also require a similar property to hold with respect to auxiliary information which may include other images. The corresponding statement is any machine that computes a *new* valid image, even in the presence of other images, knows a corresponding preimage. Although using extractability with a weaker hiding property may be sufficient for certain applications, it is of particular interest when combined with POW functions since it gives a better approximation of the properties expected from a Random Oracle.

From one angle, extractability can be interpreted as saying that the only way to produce a point in the range of this function is by taking a point in the input domain and then applying the algorithm that computes this function to the input. From another perspective, an EPOW function is an obfuscation of a point function [1, 30] with the additional property that the original source program, that computes the point function in the clear, can be extracted from the view of any potentially adversarial obfuscator. This property can in fact be defined with respect to any function family.

We define two variants of EPOW functions, namely noninteractive and interactive. Noninteractive extraction is captured by the existence of a (nonblackbox) preimage extractor. In more detail, every adversary, that tries to output a point in the range, has a corresponding extractor that gets the view of the adversary and outputs a preimage. We emphasize that the extractor gets the view of the adversary *including any private random coins.* The interactive variant is described later on.

On the relation between noninteractive EPOW functions and NIZK. Superficially, EPOW functions resemble noninteractive zero-knowledge (NIZK) arguments of knowledge [29, 28] in that an image can be viewed as a proof of preimage knowledge. However, EPOW functions and NIZK arguments of knowledge differ in several ways. First, NIZK secrecy, i.e., zero knowledge, holds over the choices of the Common Reference String (CRS) while EPOW functions require secrecy to hold without a CRS. Second, EPOW functions are not required to have efficient verification, that is deciding whether a given point belongs to the range of the function. (Not to be confused with the verification requirement on POW functions, where it is easy to check that a given output is an image of a given input.) We mention that our noninteractive EPOW constructions satisfy a weaker form of verification, which seems to be needed for the ZK application but not for our Random Oracle instantiation. On the other hand, our interactive EPOW con-

structions are not known to satisfy this form of verification. Third, NIZK arguments of knowledge require a *universal blackbox* extractor to recover a witness with the help of auxiliary information about the CRS. On the other hand, EPOW functions only require a nonblackbox extractor for every adversary. However, this extractor has to recover a preimage from the view of the adversary *without any extra information that is not given to the adversary*. The latter formulation may better capture our intuition about knowledge because it clearly demonstrates that an adversary "knows" a preimage by recovering it from its view alone.

On the relation between noninteractive EPOW functions and other knowledge assumptions. From another angle, extractable functions look similar to other knowledge assumptions such as the knowledge of exponent (KE) assumption [12, 20] and the proof of knowledge (POK) assumption [23]. In fact, we view *extractable functions as an abstraction away from specific knowledge assumptions, much like a one-way function is an abstraction of specific one-way assumptions, such as the discrete logarithm (DL) assumption*. In other words, the DL assumption gives us a one-way function but it may even give us more, e.g., a one-way permutation in certain group or certain algebraic properties. However, we abstract away from these particularities and identify the essential property needed. Likewise, we use extractable functions as a step towards capturing the abstract knowledge assumption - it provides a relatively simple primitive that is defined only in terms of its general computational properties, that seems to be useful in a number of places, and that can be realized by a number of different assumptions. (We show later that either the KE or the POK assumption, when combined with a hardness assumption such as the DDH assumption, is sufficient for constructing EPOW functions).

On the constructions. We give three simple constructions of EPOW functions. The first one uses a POW function and a "strong" notion of NIZK proof of preimage knowledge. In addition, we provide another construction from the POW construction in [7] and the KE assumption. At a high level, the KE assumption guarantees preimage extraction, while hiding can be based on a strong variant of the DDH assumption. The third construction is similar to the second one but it uses the POK assumption (with the same DDH assumption mentioned above). However, none of these constructions satisfies all of our requirements (see [8] for more details). Thus, we turn our attention to EPOW functions with interactive extraction.

Interactive EPOW Functions. These are POW functions with interactive extraction. Informally, interactive extraction means that if a party interacts *consistently* with a challenger, then it "knows" a preimage. Interaction between the prover and the challenger is restricted to Arthur-Merlin games. Furthermore, the messages sent by the prover are restricted to images of the interactive EPOW function. For instance, in a 3-round game of this type, the prover computes hashes of the preimage using different random coins for the EPOW function, $\mathbb{H}$, chosen by the challenger. In more detail, the prover sends $y = H_k(x, r_0)$ in the first round, the challenger then responds with a uniform string, r_1, and the prover sends the corresponding image, $H_k(x, r_1)$, in the last round. Here, extractability means that if the images in the first and third round share a common preimage, then the prover knows it. Similar to the noninteractive setting, knowledge of preimage is captured by the existence of a preimage extractor.

We show how to transform POW functions to interactive EPOW functions. Informally, our transformation imposes a structure on the new function so that a preimage can be recovered from any two "related" images. For clarity, consider a toy construction to recover the first bit only. Specifically, if $\mathbb{H}$ is the old POW function and x is the input, then $\mathbb{H}'$ is defined as $H'_k(x,(r_1,r_2)) = H_k((x,1),r_1), H_k((x,x_1),r_2)$, where x_1 is the first bit of x. To recover x_1, the extractor asks the prover to compute $H'_k(x,(r_1,r_2))$, then it rewinds the protocol, and forces the prover to compute $H'_k(x,(r'_1,r_1))$ using the same r_1 as before. Note that x_1 can be recovered (by simple comparison) from $H_k((x,1),r_1)$ and $H_k((x,x_1),r_1)$ computed in the first and second game respectively.

We remark that a slightly weaker notion of interactive EPOW functions can be constructed from any POW function and a corresponding Σ-protocol [5, 11] for proving preimage knowledge.

1.2 Applications

Using EPOW functions to instantiate Random Oracles in Encryption Schemes. As mentioned before, POW functions are used in [7] to capture and realize CPA-security of the encryption scheme in [4]. However, this is not sufficient for CCA2-security as POW functions may not guarantee extractability. So, an EPOW function provides the missing link, namely extractability, for replacing a Random Oracle by a POW function. Here, we use EPOW functions to instantiate the second encryption scheme in [4] (recalled shortly), and translate the proof to the standard model in a straightforward way. This scheme uses a trapdoor permutation, M, and two Random Oracles, O_1, O_2, to encrypt a message, m, as $c = (M(r), O_1(r) \oplus m, O_2(r,m))$, where r is uniform. At a high level, it is CCA2-secure because the hiding property of Random Oracles gives us semantic security while knowledge of queries gives us knowledge of plaintext (the latter property is what enables proving CCA2-security). Thus, if we replace the Random Oracle by an EPOW function in the previous scheme we get a CCA2-secure encryption scheme in the standard model. This scheme can be either noninteractive or 3-round depending on whether the EPOW function is noninteractively or interactively extractable.

This approach can be utilized to realize other encryption schemes in the RO model. In particular, we show how to instantiate some schemes that provably cannot be instantiated using the standard instantiation prescribed in the RO methodology [9, 24], where each RO query is replaced with a call to a specific function. Thus, the aforementioned instantiation is different from the standard one and does not contradict the impossibility results mentioned above. A detailed presentation of this result appears in [8].

On the connection to other approaches and CCA2 schemes. We remark that generic transformations from any semantically-secure scheme to a CCA2-secure one have been studied before [14, 27]. Also, the KE assumption has been used to prove that certain encryption schemes are plaintext-aware, which when coupled with semantic security gives CCA-secure schemes [3, 13]. Moreover, Katz [22] used the notion of proofs of plaintext knowledge to construct efficient 3-round CCA2-secure schemes. We emphasize that the contributions of this work are not in giving better or more efficient constructions than existing ones in the literature, but rather in the methodology of replacing Random Oracles as described above.

Using EPOW functions to construct 3*-round ZK protocols.* We give one more application of EPOW functions in the context of Zero-Knowledge (ZK) systems. Current 3-round ZK arguments and proofs use strong and very specific number theoretic assumptions [20, 21, 23, 3]. On the other hand, we construct 3-round ZK arguments of knowledge and membership assuming only the existence of a variant of EPOW functions and noninteractive witness-indistinguishable (WI) arguments [2, 19]. This allows for abstracting from specific number theoretic assumptions and opens the door for basing 3-round ZK arguments on other assumptions sufficient for constructing this variant of EPOW functions. On the one hand, the existence of EPOW functions is an assumption that is stated in general computational terms without resorting to specific algebraic constructs. On the other hand, the assumption seems rather basic and in particular less specific than current knowledge assumptions.

As a concrete example, we use our second EPOW construction to build such ZK arguments. We remark that the KE assumption used here is weaker than the corresponding knowledge assumptions used for constructing 3-round ZK arguments in [20, 21, 3]. Specifically, we eliminate the need for the second KE assumption in [21] and later updated in [3]. We note that both simulation and extraction are nonblackbox.

Organization. We introduce and define extractable functions in Section 3. We then highlight one noninteractive and one interactive constructions in Sections 4 and 5. The last two sections discuss applications to Random Oracle instantiation and 3-round ZK protocols, respectively. A more detailed presentation, common definitions, and proofs appear in [8].

2 Preliminaries

A function, μ, is called negligible if it decreases faster than any inverse polynomial. Formally, for any polynomial p, there exists an N_p such that, for all $n \geq N_p$: $\mu(n) < \frac{1}{p(n)}$. We reserve μ to denote negligible functions. A distribution is called **well-spread** if it has superlogarithmic min-entropy, i.e., $max_k Pr[X_n = k]$ is a negligible function in n. A probabilistic function family is a set of efficient probabilistic functions having common input and output domains. Formally, $\mathbf{H}^n = \{H_k\}_{k \in K_n}$ is a function family with key space K_n and randomness domain R_n if, for all $k \in K_n, H_k : I_n \times R_n \rightarrow O_n$. A probabilistic function family has **public randomness** if for all k, $H_k(x, r) = r, H'_k(x, r)$ for some deterministic function H'_k. A family ensemble is a collection of function families, i.e., $\mathbb{H} = \{\mathbf{H}^n\}_{n \in \mathbb{N}}$. An uninvertible function, f, with respect to a well-spread distribution, $\mathbb{X}$, is an efficiently computable function that is hard to invert on $\mathbb{X}$. Formally, for any PPT, A, $Pr[x \leftarrow X_n,\ A(f(x)) = x] < \mu(n)$. If f is uninvertible with respect to *any* well-spread distribution, then it is called uninvertible.

Perfectly One-way Probabilistic Functions. A perfectly one-way (POW) function is a probabilistic function that satisfies collision resistance and hides all information about its input. Due to its probabilistic nature, such a function is coupled with an efficient **verification** scheme that determines whether a given string is a valid hash of some given input [10].

One formulation of information hiding requires hardness of indistinguishability between hashes of the same input and hashes of different inputs [10], where the former

is taken from a well-spread distribution and the latter inputs are uniform and independent. We also consider the presence of auxiliary information, which is represented as an uninvertible function of the input. A notable special case of indistinguishability is pseudorandomness, i.e., hashes of the same input are indistinguishable from uniform. Moreover, the **statistical** version of both definitions can be obtained by dropping the requirements of auxiliary information and efficiency of the adversary. The formal definitions appear in [8].

3 Extractable Functions

An extractable function is one for which any machine that "computes" a point in the range, "knows" a corresponding preimage. As a starting point, we can formulate this notion by requiring any efficient machine that computes an image *without auxiliary input* to "know" a preimage. Although, this requirement seems reasonable, it is not sufficient for applications where auxiliary information is present. On the other hand, formulating this notion in the presence of auxiliary information is tricky. As a toy example, A can be a machine that receives an image as an input and copies it to its output. Moreover, A may still receive an image hidden in its auxiliary input in a subtle way but can be efficiently extracted from it. Yet, we do not think that this captures our intuition because A does not really compute the function, rather it decodes the image syntactically from its input. Thus, we need a meaningful way of telling apart "copying" an image from "computing" an image.

Following [18], we consider two types of auxiliary information. The first one, called **independent auxiliary information**, consists of auxiliary information computed before a function is sampled from a family ensemble, $\mathbb{H}$. We stress that this input is independent of the particular function currently used. This prevents hiding images in this type of input. The second type, called **dependent auxiliary information**, is restricted to images under $\mathbb{H}$. This is a restricted form of dependent auxiliary information but it is sufficient for our applications. Given these two types of inputs, we require that no adversary can come up with a *new* image without knowing a corresponding preimage. We capture knowledge of a preimage by requiring for every A, that computes a new image, a corresponding extractor, $\mathcal{K}_A$, that has access to the private input of A and computes a preimage. We emphasize that $\mathcal{K}_A$ has to compute the preimage from the view of A without any additional information.

For clarity, we first formalize this notion in the presence of independent auxiliary information alone before addressing the general case.

Definition 1. *Let $\mathbb{H} = \{\mathbf{H}^n\}_{n\in\mathbb{N}}$ be any verifiable family ensemble (with verifier $V_{\mathbb{H}}$). Then, $\mathbb{H}$ is called **noninteractively extractable** if for any PPT, A (with private random coins denoted by r_A), and polynomial, p, there exists a PPT, $\mathcal{K}_A$, such that for any auxiliary information, z:*

$$Pr[k \leftarrow K_n,\ y = A(k, z, r_A),\ x \leftarrow \mathcal{K}_A(k, z, r_A) : V_{\mathbb{H}}(x, y) = 1 \text{ or } (\forall x', V_{\mathbb{H}}(x', y) \neq 1)]$$

$$> 1 - \frac{1}{p(n)} - \mu(n).$$

Note that we allow a noticeable extraction error. The constructions from the KE or POK assumption have a negligible error. However, the error in our interactive constructions is not known to be negligible. So, for uniformity, we adopt the weaker notion.

There are two possible ways to introduce dependent auxiliary information into Definition 1. One can allow this auxiliary information to be images of any input while the more restrictive way forces the images to correspond to inputs chosen from well-spread distributions. Even though the former is more general, the latter is sufficient for our applications. Thus, we use the latter notion in this work. The formal definitions are not presented here due to space constraints.

Interactive extraction. In the interactive setting, we force an adversary, A, to compute not only one image but a large fraction of the images of x (recall, the function is probabilistic). We say that if A can do so, then x is extractable. We achieve the first property by forcing A to use random coins for the probabilistic function that are chosen by an external challenger. In more detail, we define a 3-round game between A and a challenger (or knowledge extractor). [4] At the end of the game, if the interaction is consistent (we say shortly what this means) then extraction is possible.

The game starts with A sending an image, y_0. The challenger sends uniform strings, $r_1, ..., r_n$, and A has to answer with images, $y_1, ..., y_n$, using $r_1, ..., r_n$ as random coins for $\mathbb{H}$. We call an interaction consistent if there is a common preimage, x, of $y_0, ..., y_n$ with $r_1, ..., r_n$ as random coins for the last n images. We then say $\mathbb{H}$ is interactively extractable if for any adversary that plays this game consistently, there is a corresponding extractor that recovers a common preimage of $y_0, ..., y_n$. We also allow A to receive an auxiliary input that can depend, in an arbitrary way, on the choice of the function from the ensemble, $\mathbb{H}$.

4 A Noninteractive EPOW Construction

Before we present the EPOW construction, we show a simpler construction that achieves extractability but satisfies a weaker notion of computational hardness, namely one-wayness. Both constructions use the KE assumption to satisfy extractability. Informally, the KE assumption says that it is hard to compute, on input p, q, g, g^a, a pair of elements (g^r, g^{ra}) without knowing r, where p and q are primes, $p = 2q + 1$, and g is a generator for the quadratic residue group modulo p. This assumption can be formulated with or without independent auxiliary information (it can be shown that it does not hold with respect to auxiliary information that depends on (p, q, g, g^a)).

Note that the KE and discrete-log (DL) assumptions imply that the family ensemble, $\mathbb{F} = \{\{f_{p,q,g,g^a}\}_{(p,q,g,g^a)\in PQGA_n}\}_{n\in\mathbb{N}}$, where $f_{p,q,g,g^a}(x) = g^x, (g^a)^x$, is an extractable one-way (EOW) family ensemble. We strengthen the previous construction into a POW function by masking x with a uniform element r as in [7]. Formally, $H_{p,q,g,g^a}(x, r) = g^r, g^{ar}, g^{rx}, g^{arx}$.

Preimage extraction. If the KE assumption holds without auxiliary information then for any PPT, A, that outputs a valid image $(g^r, g^{ar}, g^{rx}, g^{arx})$, there are two PPT, $\mathcal{K}_1$ and $\mathcal{K}_2$, such that $\mathcal{K}_1$ extracts r and $\mathcal{K}_1$ extracts rx. Consequently, $\mathbb{H}$ is extractable. Moreover, if the KE assumption holds with respect to auxiliary information, then $\mathbb{H}$ is extractable with respect to *independent* auxiliary information. However, $\mathbb{H}$ is not extractable in the presence of dependent auxiliary information. Note that extraction occurs here with negligible error.

[4] In the full version of the paper, we define a 2-round version. However, realizing this notion seems to require stronger assumptions.

Information hiding. The secrecy of this construction is similar to that of the corresponding one in [7], specifically $H_{p,q,g}(x,r) = g^r, g^{rx}$. In particular, secrecy of both constructions is based on a stronger version of the DDH assumption. Informally, g^a, g^b, g^{ab} is indistinguishable from g^a, g^b, g^z where a is drawn from a *well-spread* distribution instead of uniform. However, these secrecy notions differ in two ways. First, the [7] construction is pseudorandom while this one is an indistinguishable POW function. Second, secrecy in [7] holds for a randomly chosen function while we use secrecy that holds for any function. While the former is sufficient in some applications, such as Random Oracle instantiation in encryption schemes, the latter is needed in the ZK protocol (Section 7). Consequently, following [20], the DDH assumption used here is assumed to hold for any (p,q,g) instead of a randomly chosen one.

5 Construction of Interactive EPOW Functions

The construction presented here is based on hardness assumptions and achieves both interactive extraction and perfect one-wayness. However, it does not achieve perfect one-wayness with auxiliary information. A second construction that satisfies the latter property appears in the full version of the paper.

The idea behind both constructions is to have pairs of related images satisfy the property that it is easy to compute a preimage if both of them are available. In more detail, we identify for every r, a related $\hat{r}$, such that $O(x,r), O(x,\hat{r})$ reveals x. However, $O(x,r), O(x,\hat{r})$ is unlikely to appear in a single execution of the extraction game. So, the extractor can recover a preimage by sending r in the second round of the game to get $O(x,r)$, rewinding A, and then sending $\hat{r}$ to get $O(x,\hat{r})$. More details appear after the construction.

Construction 1 *Let $\mathbb{H} = \{\mathbf{H}^n\}_{n\in\mathbb{N}}$ and $\mathbb{G} = \{\mathbf{G}^n\}_{n\in\mathbb{N}}$ be two family ensembles. Denote by $\mathbb{O} = \{\mathbf{O}^n\}_{n\in\mathbb{N}}$ the family ensemble defined as:*

$$O_{k=(k_1,k_2,k_3)}(x,(r_0^1,r_0^2,r_0^3,r_1...,r_n,r_G)) =$$

$$r_0^2, r_0^3, H_{k_1}(x,r_0^1), H_{k_2}(t_1,r_1), ..., H_{k_2}(t_n,r_n), G_{k_3}(x,r_G),$$

where for all i, $t_i = H_k(x,r_0^2)$ if $x_i = 1$, and $t_i = H(x,r_0^3)$ otherwise.

Primage extraction. For simplicity, and to see why Construction 1 is extractable assume that A receives only a single challenge, r^O, in the second round of the extraction game. Informally, $\mathcal{K}$ tries to make A output two "related" hashes that allows it to recover x. In more detail, $\mathcal{K}$ sends $\boldsymbol{r_0^1}, r_0^2, r_0^3, r_1, ..., r_n$ to A in the first execution of the game, where all strings are uniform. $\mathcal{K}$ then rewinds A and starts a new game. In the second game, $\mathcal{K}$ sends $u_0^1, \boldsymbol{r_0^1}, u_0^3, u_1, ..., u_n$, where $u_0^1, u_0^3, u_1, ..., u_n$ are chosen uniformly but r_0^1 (the string in bold font) is the same as the one used in the first interaction. If A answers both challenges consistently, then $\mathcal{K}$ can recover x. This is so because the message in the last round of the first game contains $t = H_{k_1}(x,r_0^1)$ in the clear, while the message in the last round in the second game contains $H_{k_2}(t,u_i)$ if and only if the ith bit of x is 1. We remark that the technical proof requires that $\mathbb{H}$ satisfies a strong form of collision resistance. The formal definition and proof of extraction appears in the full version of the paper.

Information hiding. This construction uses two functions, $\mathbb{H}$ and $\mathbb{G}$, instead of one due to the properties needed to prove perfect one-wayness and extractability. Specifically, our proof of perfect one-wayness uses the assumption that $\mathbb{H}$ is *statistically* perfectly one-way. On the other hand, extractability assumes that $\mathbb{H}$ satisfies strong collision resistance. Currently, we do not know of any class of functions that satisfies this requirement except statistically binding functions. However, no single function can be both statistically pseudorandom (hiding) and statistically binding. Therefore, we use two functions. We assume that $\mathbb{G}$ is strongly collision resistant, e.g., statistically binding, so that $\mathbb{O}$ is strongly collision resistance and consequently extractable. On the other hand, $\mathbb{H}$ is assumed to be a statistically POW function. Therefore, if $\mathbb{G}$ is computationally perfectly one-way with auxiliary information (it is sufficient that the auxiliary information be only a statistically hiding function), then $\mathbb{O}$ is a *computationally* POW function. We emphasize that $\mathbb{O}$ is a POW function but not necessarily with respect to auxiliary information. In the full version of the paper, we modify the construction to meet this requirement based on a strong POW assumption.

6 Instantiating the Second Encryption Scheme of [4]

We use EPOW functions to instantiate Random Oracles in the second encryption scheme of [4] while maintaining a similar proof of security. Extractable POW functions allow us to do so because they capture two properties of Random Oracles essential for the original proof, namely, pseudorandomness and knowledge of queries.

The original scheme uses a family ensemble of trapdoor permutations, $\mathbb{M}$, with key space PK_n and trapdoor SK_n, and two random oracles O_1 and O_2. The encryption of a message, m, is $c = M_{pk}(q), O_1(q) \oplus m, O_2(m, q)$, where q is uniform.

Informally, this scheme is IND-CCA2 because it is IND-CPA and the decryption oracle does not help the adversary, A. In more detail, without access to the decryption oracle, A has a negligible advantage because M is one-way. On the other hand, any *valid* decryption query, c_1, c_2, c_3, that A makes must be preceded by two Random Oracle queries, $M_{sk}(c_1)$ and $M_{sk}(c_1), O_1(M_{sk}(c_1)) \oplus c_2$. However, if A makes any of these two queries it can compute the plaintext on its own *without the decryption oracle*.

Interactive instantiation. In the interactive setting, each oracle query is replaced by a call to a function, $\mathbb{H}$. Moreover, to encrypt a message, m, E sends a hash of a uniform string, q, in the first round. D responds by sending random strings $r_1, ..., r_n$. In the last round, E sends n hashes of q using $r_1, ..., r_n$ as random coins for $\mathbb{H}$. E also sends the ciphertext of m using the original scheme (with $\mathbb{H}$ in place of the Random Oracle) with the same q as the one used in the first round. We note that the first two messages are independent of the plaintext and thus can be sent ahead of time.

The idea behind this instantiation is to make use of interaction to verify that the sender actually knows q. This utilizes the fact that $\mathbb{H}$ satisfies interactive preimage extraction. So that any adversary communicating with the decryption oracle knows what the plaintext is. Hence, the decryption oracle does not really help the adversary. Therefore, IND-CCA2 can be reduced to IND-CPA. Since this scheme can be shown to be IND-CPA, it is IND-CCA2 in the interactive setting.

Noninteractive instantiation. A similar relation can be drawn between the existence of noninteractive EPOW functions and noninteractive instantiation of this scheme. Specifically, if $\mathbb{M}$ is a trapdoor permutation and $\mathbb{H}$ is an extractable (with dependent auxiliary information) and pseudorandom POW function with public randomness, then the scheme, $E(m, pk' = (pk, k_1)) = r_1, M_{pk}(q), y \oplus m, H_{k_1}(q, m, r_2)$, where $H_{k_1}(q, r_1) = r_1, y$, is IND-CCA2.[5]

7 Overview of the 3-round Zero-Knowledge Protocol

EPOW functions can also be used to construct 3-round ZK argument systems. Such functions allow us to do so because of their knowledge and secrecy properties. Informally, the protocol starts with the prover sending an EPOW function. The verifier responds with a corresponding image of a uniform string. The protocol ends with the prover sending a noninteractive witness-indistinguishable (WI) proof that either the theorem is true or the prover "knows" a preimage of the verifier's message. Intuitively, this protocol is sound because the verifier's message completely hides its preimage. Thus, the (polynomially-bounded) prover does not "know" a preimage. Consequently, if the verifier accepts the conversation then by the soundness property of the WI proof, the theorem has to be true. On the other hand, this protocol is zero-knowledge because the verifier "knows" a preimage of its message. In other words, a simulator can use the extractor for the EPOW function to recover a preimage and produce a WI proof using this preimage as a witness. In more detail, the simulator sends a random EPOW function in the first round. The verifier responds with an image under this function, and the simulator uses the extractor to recover a corresponding preimage, and then uses it as a witness in computing the noninteractive WI proof.

We emphasize that when using the construction of Section 4 in the above ZK protocol, we do not use any algebraic property of the discrete log in a direct way. This opens the door for basing 3-round ZK arguments on assumptions other than the KE assumption as long as such assumptions prove sufficient for constructing such EPOW functions.

We remark that using EPOW functions with arbitrary small but noticeable extraction failure probability gives weak simulation, i.e., simulation fails with arbitrary small but noticeable probability. On the other hand, if an EPOW function, such as construction of Section 4, has negligible extraction error then simulation succeeds overwhelmingly.

Acknowledgements. We are grateful to Joan Feigenbaum for many enlightening discussions and suggestions. We also thank the anonymous referees for their helpful comments and remarks.

References

1. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *Crypto*, 2001.
2. B. Barak, S. Ong, and S. Vadhan. Derandomization in cryptography. *Crypto*, 2003.

[5] The construction in Section 4 is an indistinguishable POW function but is not known to be pseudorandom. Realizing the latter requirement with noninteractive extraction remains open.

3. M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. *Crypto*, 2004.
4. M. Bellare and P. Rogaway. Random oracles are practical:a paradigm for designing efficient protocols. *CCS*, 1993.
5. M. Blum. How to prove a theorem so no one else can claim it. *Proceedings of the International Congress of Mathematicians*, 1986.
6. A. Boldyreva and M. Fischlin. On the security of OAEP. *AsiaCrypt*, 2006.
7. R. Canetti. Towards realizing random oracles:hash functions that hide all partial information. *Crypto*, 1997.
8. R. Canetti and R. R. Dakdouk. Extractable perfectly one-way functions. *eprint*, 2008.
9. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *STOIC*, 1998.
10. R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions. *STOIC*, 1998.
11. R. Cramer, I. Damgard, and J.B. Nielsen. Multiparty computation from threshold homomorphic encryption. *EuroCrypt*, 2001.
12. I. Damgard. Towards practical public key systems secure against chosen ciphertext attacks. *Crypto*, 1992.
13. A. Dent. The cramer-shoup encryption scheme is plaintext aware in the standard model. *Eurocrypt*, 2006.
14. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30, 2000.
15. A. Fiat and A. Shamir. How to prove yourself:practical solutions to identification and signature problems. *Crypto*, 1986.
16. Federal Information Processing Standard (FIPS). Secure hash standard. *NIST*, FIPS publication 180, 1993.
17. S. Goldwasser and Y. T. Kalai. On the (in)security of the fiat-shamir paradigm. *FOCS*, 2003.
18. S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. *FOCS*, 2005.
19. J. Groth, R. Ostrovsky, and A. Sahai. Non-interactive zaps and new techniques for NIZK. *Crypto*, 2006.
20. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. *Crypto*, 1998.
21. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. *eprint*, 1999.
22. J. Katz. Efficient and non-malleable proofs of plaintext knowledge and applications. *Eurocrypt*, 2003.
23. M. Lepinski. On the existence of 3-round zero-knowledge proofs. *M.S. Thesis*, 2002.
24. U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. *TCC*, 2004.
25. J. Nielsen. Separating random oracle proofs from complexity theoretic proofs:the non-committing encryption case. *Crypto*, 2002.
26. R. Rivest. The MD5 message-digest algorithm. *IETF Network Working Group*, RFC 1321, 1992.
27. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *FOCS*, 1999.
28. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. *Crypto*, 2001.
29. A. De Santis and G. Persiano. Zero knowledge proofs of knowledge without interaction. *FOCS*, 1992.
30. H. Wee. On obfuscating point functions. *STOIC*, 2005.