Instantiating Random Oracles via UCEs

Mihir Bellare[1], Viet Tung Hoang[2], and Sriram Keelveedhi[1]

[1] Dept. of Computer Science & Engineering, University of California San Diego
[2] Dept. of Computer Science, University of California Davis

Abstract. This paper provides a (standard-model) notion of security for (keyed) hash functions, called UCE, that we show enables instantiation of random oracles (ROs) in a fairly broad and systematic way. Goals and schemes we consider include deterministic PKE; message-locked encryption; hardcore functions; point-function obfuscation; OAEP; encryption secure for key-dependent messages; encryption secure under related-key attack; proofs of storage; and adaptively-secure garbled circuits with short tokens. We can take existing, natural and efficient ROM schemes and show that the instantiated scheme resulting from replacing the RO with a UCE function is secure in the standard model. In several cases this results in the first standard-model schemes for these goals. The definition of UCE-security itself is quite simple, asking that outputs of the function look random given some "leakage," even if the adversary knows the key, as long as the leakage does not permit the adversary to compute the inputs.

1 Introduction

The core contribution of this paper is a new notion of security for (keyed) hash functions called UCE (Universal Computational Extractor). UCE-security is the first well-defined, standard-model security attribute of a hash function shown to permit the latter to securely instantiate ROs across a fairly broad spectrum of schemes and goals.

Under the random-oracle paradigm of Bellare and Rogaway (BR93) [14], a "real-world" or instantiated scheme is obtained by implementing the RO of the overlying ROM scheme via a cryptographic hash function. The central (and justified) critique of the paradigm [36] is that the instantiated scheme has only heuristic security. This paper offers *proven* security for the (standard model) instantiated schemes. The proof is based on the (standard-model) assumption that the instantiating function is UCE-secure.

UCE of course does not *always* work. But we show that it works across a fairly large, diverse and interesting spectrum of schemes and goals including deterministic PKE; message-locked encryption; hardcore predicates; point-function obfuscation; encryption of key-dependent messages; encryption secure under related-key attack; OAEP; correlated-input secure hashing; adaptively-secure garbled circuits; and proofs of safe storage. In all these cases we can use UCE to obtain standard-model solutions, in most cases instantiating known, natural and

R. Canetti and J.A. Garay (Eds.): CRYPTO 2013, Part II, LNCS 8043, pp. 398–415, 2013.

efficient schemes, and in several cases getting the first standard-model schemes for the goals in question.

UCE is quite simple and natural, yet powerful. The basic intuition is that the output of a UCE-secure function looks random even given the key and some "leakage," as long as the inputs are not computable from the leakage. Let us now step back to provide some background and then return to our contributions.

Background. The random-oracle paradigm of BR93 [14] has two steps: (1) Design your scheme, and prove it secure, in the ROM, where the scheme algorithms and adversary have access to a RO denoted RO (2) Instantiate the RO to get the standard model scheme that is actually implemented and used. We will consider instantiation via a family of functions H, which means that the instantiated scheme is obtained by replacing RO calls of the ROM-scheme algorithms by evaluations of the deterministic function $\mathsf{H.Ev}(hk, \cdot)$ specified by a key $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$, where λ is the security parameter. The key hk is put in the public key of the instantiated scheme if the latter is public key, else enters in some scheme-dependent way. The suggestion of BR93 was that if H "behaved like a RO," the instantiated scheme would be secure in the standard model. They suggested to obtain such instantiations, heuristically, via cryptographic hash functions. The fundamental subsequent concern has been the lack of a proof of security for the instantiated scheme. Canetti, Goldreich and Halevi (CGH98) [36] show that this lack in some cases cannot be overcome because there exist schemes secure in the ROM but which no family of functions can securely instantiate. Advocates for the defense counter by pointing out that the counter-example schemes are artificial, and in-use instantiations of "natural" ROM schemes are unbroken. This has led to examples that are in one way or another less artificial [7, 37, 42, 51, 60, 64].

It is not the purpose of this paper to take sides in this debate. We want instead to make a scientific contribution towards better grounding the security of instantiated ROM schemes.

The core problem and previous work. The lack of a proof of security for the instantiated scheme is, we submit, a consequence of an even more fundamental lack, namely that of a *definition*, of what it means for a family of functions to "behave like a RO," that could function as an assumption on which to base the proof. The PRF definition [50], which has worked so well in the symmetric setting, is inadequate here because PRF-security relies on the adversary not knowing the key. And collision-resistance (CR) is far from sufficient in any non-trivial usage of a RO.

Canetti [34] was the first to articulate this position and seek a standard-model primitive sufficient to capture some usages of a RO. Notions such as Perfectly One-Way Probabilistic Hash Functions (POWHFs) [34,35,39] and non-malleable hash functions [19] have however proven of limited applicability [21]. Another direction has been to try to instantiate the RO in particular schemes like OAEP [15], again with limited success [21, 22] or under strong assumptions on RSA [59].

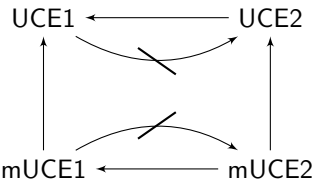

Fig. 1. Relations between UCE security notions. Letting S denote the set of all families H that are S-secure, an arrow $\mathsf{A} \to \mathsf{B}$ represents $\mathsf{A} \subseteq \mathsf{B}$, meaning any H that is A-secure is also B-secure. A barred arrow $\mathsf{A} \not\to \mathsf{B}$ represents $\mathsf{A} \not\subseteq \mathsf{B}$, meaning there is an H that is A-secure but not B-secure. (Assuming of course that some A-secure H exists.)

Our position is philosophically different from that of [34, 39]. These works aimed for security notions that they could achieve under standard assumptions. Expectedly, applicability was limited. We aim to maximize applicability and are willing to see our notion (UCE) as an assumption rather than something to achieve under other assumptions.

UCE. Our definition considers an adversary S, called the source, who is given an oracle HASH, the latter being $\mathsf{H.Ev}(hk, \cdot)$ for key $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ if the challenge bit b is 1, and a RO otherwise. If security now asks that S not figure out b, then, if we deny it hk, we would be back to PRFs, and if we give it hk, security would be unachievable. So we don't ask S to figure out b. Instead, it must pass to an accomplice adversary D, called the distinguisher, some information L called the leakage. The distinguisher *is given the key* hk and must figure out b.

Clearly, security is not achievable for arbitrary leakage. (The source could include in L a point x and the result $y = \text{HASH}(x)$ of its oracle on x, and D, having hk, can test whether or not $y = \mathsf{H.Ev}(hk, x)$.) We put an extra condition on the source that we call unpredictability. It requires that it be computationally infeasible for a predictor adversary P, given the leakage produced by the source in the *random* ($b = 0$) game, to find any of the inputs queried by the source to its oracle. Note that unpredictability is a property of the source, not of the family of functions H, the latter not figuring in the definition at all.

Security, finally, requires that for any PT *unpredictable* source S, and any PT distinguisher D, the advantage of S, D in figuring out b is negligible. See Section 4 for a formal definition of this notion that we call UCE1. A variant called UCE2, introduced in [11], preserves the source-distinguisher framework of UCE1 but replaces the unpredictability condition with a weaker condition we call reset-security. ("Weaker" because any unpredictable source is reset-secure. This makes UCE stronger: any UCE2-secure family is UCE1-secure.) Both UCE1 and UCE2 involve a single hashing key. We define natural multi-key extensions mUCE1 and mUCE2 as well.

In [11] we examine the relation between UCE and standard security notions for families of functions such as PRF-security and collision-resistance (CR). We show that UCE (of whatever form) neither implies, nor is implied by, any

Goal	Result	UCE
D-PKE	Instantiation of the ROM EwH scheme of [6] to obtain the first standard model deterministic PKE scheme providing full IND [9] and PRIV [6] security.	UCE1
MLE	Instantiation of the ROM convergent encryption scheme of [12, 43], showing this in-use message-locked encryption scheme meets the IND$-CDA goal of [12].	UCE1
HC	Any UCE1-secure family is hardcore for any one-way function and allows for extraction of any number of hardcore bits.	UCE1
BR93 PKE	Instantiation of a natural ROM PKE scheme from BR93 [14] showing it is IND-CPA-secure.	UCE1
PFOB	Instantiation of a ROM point-function obfuscation scheme of [38] to obtain a secure standard-model scheme.	mUCE1
KDM	Instantiation of the ROM BRS scheme [18] to get an efficient and natural standard-model symmetric scheme for encryption of key-dependent messages.	mUCE1
RKA	An efficient standard-model symmetric encryption scheme providing best-possible security against related-key attacks.	mUCE1
CIH	Construction from UCE1 of correlation-intractable hash functions meeting the strongest notion of [54].	UCE1
STORE	Instantiation of a natural ROM proof of storage scheme from [67].	UCE1
OAEP	IND-CPA-KI security of OAEP [15] assuming partial one-wayness (with UCE1) or one-wayness (with UCE2) of the underlying trapdoor function.	UCE1/2
GB	Standard-model adaptively secure garbling with short tokens.	UCE2

Fig. 2. Applications of UCE: We summarize results for different goals, the last column indicating the form of UCE used

of these. We also investigate the relations between the different forms of UCE we have introduced. Our findings are summarized in Fig. 1. As indicated there, UCE2 implies UCE1 but not vice versa, and analogously mUCE2 implies mUCE1 but not vice versa. Of course mUCE1 implies UCE1 and mUCE2 implies UCE2. We do not know whether UCE1 implies mUCE1, and analogously for UCE2 and mUCE2.

Applications. Fig. 2 summarizes the applications we now discuss.

1. Deterministic PKE. The EwH deterministic PKE (D-PKE) ROM scheme of BBO07 [6] encrypts message m under public key ek by applying the RO to $ek\|m$ to get coins r and then encrypting m with an IND-CPA PKE scheme under ek and coins r. They showed that this achieved their PRIV notion of security in the ROM. Our instantiation adds $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ to

the public key and then replaces the RO with $\mathsf{H.Ev}(hk, \cdot)$. We show that if H is UCE1-secure then this instantiated D-PKE scheme is PRIV-secure in the standard model. This is not only the first standard-model PRIV-secure scheme (previous standard-model D-PKE schemes achieve only restricted notions of blocksource-PRIV-security [9, 20, 32, 47]) but also the most practical. Our proof makes crucial use of the equivalence between PRIV and an indistinguishability-style notion IND of D-PKE security [9].

2. Message-locked encryption. In convergent encryption (CE) [12, 43], message m is encrypted using a deterministic symmetric encryption scheme with the key derived, via a RO, from the message itself. CE is the most natural and prominent embodiment of message-locked encryption (MLE) and is in current use by commercial cloud-storage providers to provide secure deduplicated storage. The scheme is shown in [12] to meet, in the ROM, a formal notion of MLE-security called PRV$-CDA. We instantiate with a UCE1-family, putting the key in public parameters, and show that the resulting MLE scheme is PRV$-CDA in the standard model.
3. Hardcore functions. A RO is an ideal hardcore function, with $\mathsf{RO}(x)$ returning any number of bits that remain pseudorandom given $f(x)$ where f is one-way. UCE1 families can securely instantiate the RO here, meaning are secure hardcore functions for any one-way function, able to extract as many bits as desired.
4. BR93 PKE. A simple and natural ROM IND-CPA PKE scheme from [14] encrypts m by picking random x and returning $(f(x), \mathsf{RO}(x) \oplus m)$ where f is a trapdoor function in the public key. We show that instantiating the RO with a UCE1-secure family preserves the IND-CPA security.
5. Point-function obfuscation. A *point function* has non-$\perp$ output on just one point. Canetti, Kalai, Varia, and Wichs [38] give a ROM point-function obfuscation scheme. We mUCE1-instantiate their construction to obtain a standard-model point-function obfuscation scheme.
6. KDM-secure SE. Black, Rogaway and Shrimpton (BRS) [18] showed that the following simple and efficient symmetric encryption (SE) scheme is KDM-secure in the ROM: to encrypt message m under key K, pick a random r and return $(r, \mathsf{RO}(r \| K) \oplus m)$. We instantiate by letting the random value r in the BRS scheme take on the role of a fresh hash key, so that, to encrypt m, we pick $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ and return $(hk, \mathsf{H.Ev}(hk, K) \oplus m)$. We prove that if H is mUCE1-secure then this instantiated scheme is KDM secure in the standard model. (We achieve non-adaptive KDM security, but this includes popular cases such as key-cycles.) This scheme is more practical than other standard-model KDM-secure encryption schemes such as [1, 2, 4, 31, 62].
7. RKA-secure SE. Symmetric encryption schemes secure against related-key attack (RKA) must preserve security even when encryption is performed under keys derived from the original key by application of a key-deriving function. Previous schemes [3, 13] provided security for algebraic key-deriving functions such as linear or polynomial functions over a keyspace that is a particular group depending on the scheme. We provide a scheme that has

"best possible" security, in that key-deriving functions are arbitrary subject only to a condition necessary for security, namely to have unpredictable outputs. Furthermore, in our scheme, keys are binary strings rather than group elements, so we cover the most common practical attacks, such as XORing a constant to the key. We assume only a mUCE1-secure family of functions.

8. Correlation-intractable secure hashing. Goyal, O'Neill and Rao (GOR) introduced the notion of correlated-input hash (CIH) function families [54] and proposed several notions of security for them. GOR provided constructions achieving limited CIH security from the q-DHI assumption of [25] and from RKA-secure blockciphers, but achieving full CIH security in the standard model has remained open. We solve this problem, showing that UCE1-secure function families are selective (pseudorandomness) CIH secure in the terminology of GOR.

9. Secure storage. Ristenpart, Shacham and Shrimpton [67] give a ROM protocol allowing a client to check that a server is storing its file in its entirety, its interest being that constructions indifferentiable from a RO [63] may fail to securely replace the RO. In contrast, we show that UCE1 instantiation succeeds.

10. OAEP. OAEP [15] has been a benchmark for RO instantiation [21, 22, 59]. We instantiate OAEP by adding $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ to the public key and then implementing both the ROs via $\mathsf{H.Ev}(hk, \cdot)$. Under UCE1, we get IND-CPA-KI security under the partial-domain one-wayness, and hence by [46] under standard one-wayness,of RSA; under UCE2 we get it directly under standard one-wayness. IND-CPA-KI is IND-CPA when challenge messages are not allowed to depend on the public key. (This limitation arises because in UCE the strings being hashed by the source cannot depend on the hashing key. We note that this UCE feature does not *always* prevent us from achieving full IND-CPA. Indeed, we do achieve it for the BR93 PKE scheme, because there the inputs to the RO do not depend on the messages.) Kiltz, O'Neill and Smith (KOS) [59] show that RSA-OAEP is IND-CPA-secure if its two ROs are replaced with t-wise independent hash functions and RSA is Φ-hiding [33]. In comparison our results for RSA are under the standard one-wayness assumption.

11. Adaptively-secure garbling. Verifiable outsourcing [48], as well as one-time programs [52], call for garbling schemes that are adaptively secure [10]. Standard-model adaptively-secure garbling has however so far been at the cost of large tokens, meaning ones as large as the circuit being garbled [10, 53]. This is not only inefficient but makes the resulting verifiable outsourcing "trivial" in that the client does as much work as the server. We provide a UCE2-based garbling scheme that is adaptively secure and has short tokens. This is the first standard-model garbling scheme with these properties and it results in the first non-trivial instantiation of the outsourcing scheme of [48]. Our garbling scheme is obtained by instantiating a ROM garbled circuit construction of [66].

Constructing UCE-secure families. We provide a ROM construction of a family of functions shown to achieve both mUCE1 and mUCE2. (And thereby UCE1 and UCE2.)

This at first may seem like a step backwards; wasn't the purpose of UCE to avoid the ROM? As explained in more depth in Section 2, it is a step forward because the security we require from families of functions in implementations has moved from something heuristic and vague, namely to "behave like a RO," to something well defined, namely to be UCE-secure.

In practice we would aim to instantiate UCE-secure families via blockciphers or cryptographic hash functions. We explain that direct instantiation with a blockcipher (e.g. AES) is not secure due to the invertibility of the blockcipher. Cryptographic hash functions, being unkeyed, do not directly provide instantiations either. We suggest instead to use HMAC [5, 8].

This extended abstract. Due to space limitations, this extended abstract will provide only the UCE1 definition and detail only one application from Fig. 2. We refer the reader to our full paper [11] for definitions of mUCE1, UCE2 and mUCE2 and for the 10 omitted applications.

2 Perspective and Discussion

We explain why UCE is step forward even if we can (currently) only achieve it in the ROM, and how UCE relates to other assumptions.

Layered cryptography. Currently, RO-based design *directly* proves schemes (for end goals) secure in the ROM. We are instead advocating and using what we call a *layered* approach. In this approach, *base primitives* with standard-model security definitions are validated in the ROM. End goals are then reached from the base primitives purely in the standard model, the ROM being entirely dispensed with in the second step. This is illustrated in Fig. 3. We are showing that UCE can function as such a base primitive, and a powerful one at that, since many goals may be reached from it.

In implementations, we would continue to instantiate families assumed UCE-secure via appropriately-keyed cryptographic hash functions, but we claim this layered approach is still an important advance on direct ROM-based design. This is because the property we desire from the object (family of functions) actually being used in the implementation has moved from something heuristic and vague ("behave like a random oracle") to something precise and meaningful (be UCE-secure). Cryptanalytic validation of UCE security, even if difficult, is at least meaningful, while cryptanalytic evaluation of "behaving like a RO" is not even meaningful because the phrase in quotes is not well defined.

We make an analogy with pairing-based cryptography. Here we have seen the proposal of a large number of standard-model assumptions, including BDH [28], DLIN [27], SDH [27], BDHE [26] and SD (Subgroup Decision) [30] to name just a small fraction. These assumptions are (ubiquitously) validated in the generic-group model, end goals then reached from the assumptions in the standard

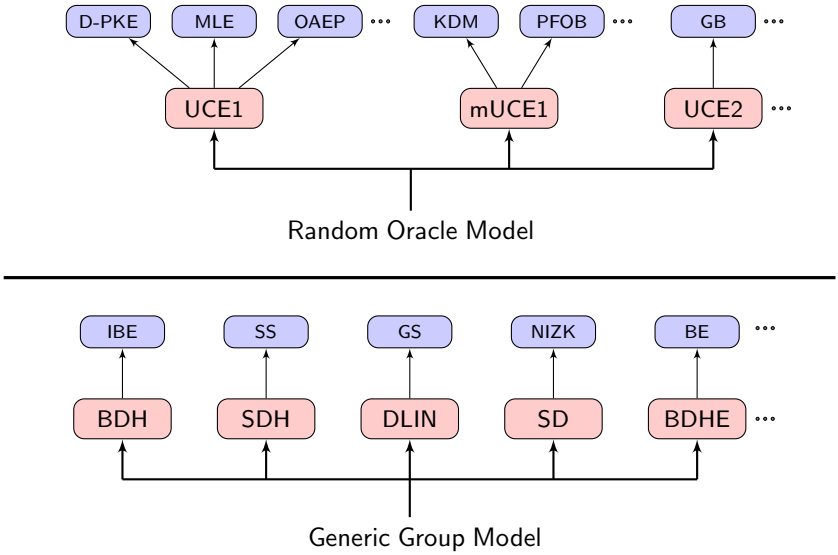

Fig. 3. The layered-cryptography paradigm for the ROM (left) and for pairing-based cryptography (right). Assumptions are validated in the idealized model and then used to attain end goals entirely in the standard model. SS refers to the short signatures of [24]; BE refers to the broadcast encryption scheme of [29]; NIZK refers to the NIZK arguments of [55]. See text for other abbreviations.

model. But the generic-group model is subject to issues, critiques and counter-examples analogous to those for the ROM, if not worse [41,44]. We believe that the (deserved) success and acceptance of pairing-based cryptography, and that it has not come under as much fire as ROM-based cryptography, are due in part to what, in our terminology, is its layered approach (again illustrated in Fig. 3). Namely, schemes for end goals, rather than being directly validated in the generic model (the un-layered or direct approach), are based on standard-model assumptions that are themselves validated in the generic-group model and amenable to cryptanalysis.

It is perhaps curious that the layered approach has not been explicitly articulated and widely used for ROM-based cryptography, while it has been widely used (even if not explicitly articulated) in pairing-based cryptography. The benefits are identical in the two cases. We view our work as making layered cryptography an explicit approach for ROM-based design.

Assumption degree and achieving UCE. In the UCE definition, the adversary consists of stages (source and distinguisher) that (due to the unpredictability condition) cannot completely share state. We refer to this as a second-degree assumption, as opposed to a first-degree assumption, where the adversary is a single algorithm. Put another way, a first-degree assumption can be specified via an interaction (game) between an adversary and a challenger. (In some places [57,65] this is called a "standard" assumption, but we think this is less

clear than "first degree.") UCE cannot. This distinction is crucial to its power and to why various negative results are circumvented. Thus, Wichs [68] shows that first-degree assumptions do not suffice for PRIV-secure D-PKE, but our proof that UCE does suffice is not a contradiction because UCE is not first-degree.

A corollary is that UCE itself cannot be achieved based on first-degree assumptions. This does not necessarily mean that UCE is an implausible assumption. (A second-degree assumption does not have to be implied by a first-degree one to be true.)

Without ROs. There is a large body of work on cryptography without random oracles. (A Google Scholar search shows 286 papers with the phrase "without random oracles" in the title, and 3,640 with this phrase somewhere in the paper, as of June 6, 2013.) More often than not, the without-RO schemes of such works are completely different from, and less efficient than, RO ones. While UCE also serves, of course, to get without-RO schemes, it does more, permitting these to be obtained by actual instantiation of the RO in a ROM scheme, so that the efficiency and practicality of the starting ROM scheme is preserved.

Directions. We believe that achieving UCE under other assumptions is an interesting and important direction for future work. We suggest to begin by targeting restricted versions of UCE, for example UCE1 for block sources. This we may hope to achieve under first-degree assumptions. Hope is lent to the enterprise by the fact that D-PKE that is PRIV-secure for block sources has been achieved under standard assumptions [20,32,47]. Full UCE security would, of course, require second-degree assumptions.

UCE is a framework permitting definitional variants beyond the four we have formalized. One could define variants with extractability, which may be useful for further applications. A tempting variant is to allow some communication back from the distinguisher to the source. This opens the door to many interesting applications, but is a dangerous path to tread, for any version we, at least, have formalized, we have also broken, even for forms of communication that seemed highly restricted.

Discussion, limitations and related work. That the source adversary in UCE does not get the key is important in avoiding impossibility results like those in [36,63]. (For example, UCE does not imply correlation intractability as defined, and shown to be unachievable in the standard model, by [36].)

UCE is not a panacea in the sense that it can replace ROs everywhere. UCE helps in cases where the RO is applied to inputs hidden (at least in part) from the adversary. As far as we know, UCE will not help for tasks like instantiating the RO in FDH signatures [16]. This is consistent with impossibility results [42].

Curiously, UCE-based proofs for instantiated schemes are sometimes simpler than the proofs for the starting ROM schemes. This is the case for D-PKE. The intuition for the ROM security of the EwH scheme of [6] is simple enough, but a rigorous ROM proof is in our view less straightforward than our proofs for the UCE1-based instantiation of EwH.

The term "computational extractor" has been used for primitives that extract pseudorandomness from distributions that have computational min-entropy [40, 45, 61]. A UCE-secure family instead extracts pseudorandomness from unpredictable distributions. These may or may not have computational min-entropy in the formal sense the latter is defined [56] but we view unpredictability as we defined it as another computational relaxation of min-entropy so preserved the "extractor" name. "Universal" refers to the ability to do this from *any* starting (unpredictable) distribution.

Programmable hash functions [58] are an information-theoretic tool that in some way mimic the "programmability" of ROs and were used by [58] to build signature schemes with short signatures in the standard model. They do not serve to instantiate ROs in the kinds of applications we consider. Several works [23,49] define new security properties of hash functions tailored for their own particular applications.

3 Preliminaries

By $\lambda \in \mathbb{N}$ we denote the security parameter and by 1^λ its unary representation. We denote the number of coordinates of a vector $\mathbf{x}$ by $|\mathbf{x}|$, and the length of a string $x \in \{0,1\}^*$ by $|x|$. Algorithms are randomized unless otherwise indicated. Running time is worst case. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\ A(x_1, \ldots)$ be the resulting of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots,)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$.

We use the code based game playing framework of [17] augmented with explicit Main procedures as in [67]. (See Fig. 4 for an example.) By $\mathrm{G}^A(\lambda)$ we denote the event that the execution of game G with adversary A and security parameter λ results in output true, the game output being what is returned by Main.

4 UCE1

We define UCE1 security of a family of functions and provide a simplified but equivalent form of unpredictability. In [11] we provide further basic results and also define mUCE1.

Syntax. A family of functions H specifies the following. On input the unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$, key generation algorithm $\mathsf{H.Kg}$ returns a key $hk \in \{0,1\}^{\mathsf{H.Kl}(\lambda)}$, where $\mathsf{H.Kl}\colon \mathbb{N} \to \mathbb{N}$ is the keylength function associated to H. The deterministic, PT evaluation algorithm $\mathsf{H.Ev}$ takes 1^λ, a key $hk \in [\mathsf{H.Kg}(1^\lambda)]$, an input $x \in \{0,1\}^*$ with $|x| \in \mathsf{H.IL}(\lambda)$, and a unary encoding 1^ℓ of an output length $\ell \in \mathsf{H.OL}(\lambda)$ to return an output $\mathsf{H.Ev}(1^\lambda, hk, x, 1^\ell) \in \{0,1\}^\ell$. (The syntax in the Introduction had simplified by dropping the first and last inputs.) Here $\mathsf{H.IL}$ is the input-length function associated to H, so that $\mathsf{H.IL}(\lambda) \subseteq \mathbb{N}$

$\underline{\text{MAIN UCE}^{S,D}_{\mathsf{H}}(\lambda)}$	$\underline{\text{MAIN Pred}^{P}_{S}(\lambda)}$	$\underline{\text{MAIN SPred}^{P'}_{S}(\lambda)}$
$b \leftarrow\!\!\$\ \{0,1\}$; $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$\mathsf{done} \leftarrow \mathsf{false}$; $Q \leftarrow \emptyset$	$Q \leftarrow \emptyset$
$L \leftarrow\!\!\$\ S^{\text{HASH}}(1^\lambda)$	$L \leftarrow\!\!\$\ S^{\text{HASH}}(1^\lambda)$	$L \leftarrow\!\!\$\ S^{\text{HASH}}(1^\lambda)$
$b' \leftarrow\!\!\$\ D(1^\lambda, hk, L)$	$\mathsf{done} \leftarrow \mathsf{true}$	$x \leftarrow\!\!\$\ P'(1^\lambda, L)$
Return $(b' = b)$	$Q' \leftarrow\!\!\$\ P^{\text{HASH}}(1^\lambda, L)$	Return $(x \in Q)$
$\underline{\text{HASH}(x, 1^\ell)}$	Return $(Q \cap Q' \neq \emptyset)$	$\underline{\text{HASH}(x, 1^\ell)}$
If $T[x,\ell] = \perp$ then	$\underline{\text{HASH}(x, 1^\ell)}$	$Q \leftarrow Q \cup \{x\}$
If $b = 1$ then	If $\mathsf{done} = \mathsf{false}$ then	If $T[x,\ell] = \perp$ then
$T[x,\ell] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^\ell)$	$Q \leftarrow Q \cup \{x\}$	$T[x,\ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Else $T[x,\ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	If $T[x,\ell] = \perp$ then	Return $T[x,\ell]$
Return $T[x,\ell]$	$T[x,\ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	
	Return $T[x,\ell]$	

Fig. 4. Games UCE**,** Pred **used to define UCE1 security of family of functions** H**, and game** SPred **defining the simplified but equivalent form of unpredictability.** Here S is the source, D is the distinguisher, P is the predictor and P' is the simple predictor.

is the (non-empty) set of allowed input lengths, and similarly $\mathsf{H.OL}$ is the output-length function associated to H, so that $\mathsf{H.OL}(\lambda) \subseteq \mathbb{N}$ is the (non-empty) set of allowed output lengths. The latter allows us to cover fixed output length (FOL) functions, captured by $\mathsf{H.OL}(\lambda)$ being a set of size one, or variable output length (VOL) functions, where $\mathsf{H.OL}(\lambda)$ could be larger and even be $\mathbb{N}$. We say that H has input-length $\ell\colon \mathbb{N} \to \mathbb{N}$ if $\mathsf{H.IL}(\lambda) = \{\ell(\lambda)\}$ for all $\lambda \in \mathbb{N}$, and if such an ℓ exists we denote it by $\mathsf{H.il}$. We say H has output-length $\ell\colon \mathbb{N} \to \mathbb{N}$ if $\mathsf{H.OL}(\lambda) = \{\ell(\lambda)\}$ for all $\lambda \in \mathbb{N}$, and if such an ℓ exists we denote it by $\mathsf{H.ol}$.

UCE1 SECURITY. We define what it means for a family of functions H to be UCE1-secure. Let S be an adversary called the *source* and D an adversary called the *distinguisher*. We associate to them and H the game $\text{UCE}^{S,D}_{\mathsf{H}}(\lambda)$ of Fig. 4. The source has access to an oracle HASH and we require that any query $x, 1^\ell$ made to this oracle satisfy $|x| \in \mathsf{H.IL}(\lambda)$ and $\ell \in \mathsf{H.OL}(\lambda)$. When the challenge bit b is 1 (the "real" case) the oracle responds via $\mathsf{H.Ev}$ under a key hk that is chosen by the game and *not* given to the source. When $b = 0$ (the "random" case) it responds as a RO. The source communicates to its accomplice distinguisher a string $L \in \{0,1\}^*$ we call the *leakage*. The distinguisher *does* get the key hk as input and must now return its guess $b' \in \{0,1\}$ for b. The game returns true iff $b' = b$, and the UCE1 advantage of (S, D) is defined for $\lambda \in \mathbb{N}$ via $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\lambda) = 2\Pr[\text{UCE}^{S,D}_{\mathsf{H}}(\lambda)] - 1$. One's first thought may now be to say that H is UCE1-secure if $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible for all PT S and all PT D. But an obvious attack shows that no H can meet this definition. Indeed, S can pick some x and ℓ, let $h \leftarrow \text{HASH}(x, 1^\ell)$ and return leakage $L = (x, h, 1^\ell)$ to D. The latter, knowing hk, can return 1 if $h = \mathsf{H.Ev}(1^\lambda, hk, x, 1^\ell)$ and 0 otherwise. We obtain a meaningful and useful definition of UCE1-security for H

by restricting attention to sources that are what we call "unpredictable." The formalization considers game $\mathrm{Pred}^P_S(\lambda)$ of Fig. 4 associated to source S and an adversary P called a *predictor*. Given the leakage, the latter outputs a set Q'. It wins if this set contains any HASH-query of the source. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{pred}}_{P,S}(\lambda) = \Pr[\mathrm{Pred}^P_S(\lambda)]$. We say that source S is *unpredictable* if $\mathsf{Adv}^{\mathsf{pred}}_{P,S}(\cdot)$ is negligible for all PT predictors P. We stress that in the prediction game, the HASH oracle of the source is a RO like in the random game, and the predictor gets the same oracle. The family H is not involved in this definition; unpredictability is a property of the source. Finally, we say that H is UCE1-secure if $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible for all unpredictable, PT sources S and all PT distinguishers D. It is convenient to let $\mathsf{UCE1}$ denote the set of all function families H that are UCE1-secure.

SIMPLE UNPREDICTABILITY. Applications of UCE1 will involve proving the unpredictability of sources we construct. This task is simplified by using a simpler formulation of unpredictability, called simple unpredictability, that is equivalent to the original. The formalization considers game $\mathrm{SPred}^{P'}_S(\lambda)$ of Fig. 4 associated to source S and an adversary P' called a *simple predictor*. There are two simplifications: the simple predictor does not have access to the RO HASH, and its output is a single string x rather than a set of strings. It wins if x is a HASH-query of the source. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{spred}}_{P',S}(\lambda) = \Pr[\mathrm{SPred}^{P'}_S(\lambda)]$. We say that source S is *simple unpredictable* if $\mathsf{Adv}^{\mathsf{spred}}_{P',S}(\cdot)$ is negligible for all PT simple predictors P'. The following, whose proof is in [11], says that simple unpredictability is equivalent to unpredictability.

Lemma 1. Let S be a source. Then S is unpredictable if and only if it is simple unpredictable.

FROM FOL TO VOL. In [11] we show how to build a UCE1-secure family with variable output length (VOL) from a UCE1-secure family with fixed output length (FOL) in a simple way using a PRF.

5 Applications of UCE1

We detail one of the 11 applications of Fig. 2. For the rest, see [11].

DETERMINISTIC ENCRYPTION. EwH is a simple and natural D-PKE scheme from [6] that deterministically encrypts m by encrypting m with a randomized IND-CPA scheme with the coins derived by applying a RO to m. In the ROM the scheme is PRIV-secure [6] and equivalently IND-secure [9]. We show that instantiating the RO with a UCE1 hash family results in a scheme meeting the same notion of security in the standard model. Previous standard model schemes have met notions providing security only when one assumes messages are drawn from a blocksource, meaning each message has high min-entropy even given previous ones [20,32]. Instantiated EwH however meets the original and full notions of [6,9] which only make the necessary assumption that each individual

$\underline{\text{MAIN } \text{IND}^A_{\mathsf{PKE}}(\lambda)}$	$\underline{\mathsf{DE.Kg}(1^\lambda)}$
$b \leftarrow\!\!\$\ \{0,1\}$	$(ek, dk) \leftarrow\!\!\$\ \mathsf{RE.Kg}(1^\lambda)\ ;\ hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$(ek, dk) \leftarrow\!\!\$\ \mathsf{DE.Kg}(1^\lambda)$	Return $((ek, hk), dk)$
$(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)$	$\underline{\mathsf{DE.Enc}(1^\lambda, (ek, hk), m)}$
For $i = 1$ to $\lvert\mathbf{m}_b\rvert$ do	$r \leftarrow \mathsf{H.Ev}(1^\lambda, hk, ek \,\|\, m, 1^{\mathsf{RE.rl}(\lambda)})$
$\quad \mathbf{c}[i] \leftarrow\!\!\$\ \mathsf{DE.Enc}(1^\lambda, ek, \mathbf{m}_b[i])$	$c \leftarrow \mathsf{RE.Enc}(1^\lambda, ek, m; r)\ ;$ Return c
$b' \leftarrow\!\!\$\ A_2(1^\lambda, ek, \mathbf{c})$	$\underline{\mathsf{DE.Dec}(1^\lambda, dk, c)}$
Return $(b = b')$	$m \leftarrow \mathsf{RE.Dec}(1^\lambda, dk, c)\ ;$ Return m

Fig. 5. Left: The IND game. **Right:** D-PKE scheme $\mathsf{DE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$.

message has high min-entropy, but allow messages to be arbitrarily correlated. This is the first standard-model scheme meeting the PRIV and IND notions.

A PKE scheme PKE specifies a triple of PT algorithms. Via $(ek, dk) \leftarrow\!\!\$\ \mathsf{PKE.Kg}(1^\lambda)$ we generate keys. Via $c \leftarrow\!\!\$\ \mathsf{PKE.Enc}(1^\lambda, ek, m)$ we can encrypt a message $m \in \{0,1\}^{\mathsf{PKE.il}(\lambda)}$ where $\mathsf{PKE.il}\colon \mathbb{N} \to \mathbb{N}$ is the message-length function of the scheme. Via $m \leftarrow \mathsf{PKE.Dec}(1^\lambda, dk, c)$ we deterministically decrypt. We say PKE is a D-PKE scheme if the encryption algorithm PKE.Enc is deterministic. The game defining the IND notion of security for D-PKE scheme DE, following [9], is in Fig. 5. An IND adversary $A = (A_1, A_2)$ is a pair of PT algorithms, where A_1 on input 1^λ returns a pair $(\mathbf{m}_0, \mathbf{m}_1)$ of vectors of messages. It is required that there are functions v, ℓ, depending on the adversary, such that $|\mathbf{m}_0| = |\mathbf{m}_1| = v(\lambda)$ and $|\mathbf{m}_b[i]| = \ell(\lambda)$ for all $b \in \{0,1\}$ and $i \in [v(\lambda)]$. It is also required that the strings (messages) $\mathbf{m}_0[1], \ldots, \mathbf{m}_0[|\mathbf{m}_0|]$ are distinct and the strings (messages) $\mathbf{m}_1[1], \ldots, \mathbf{m}_1[|\mathbf{m}_1|]$ are distinct. The guessing probability $\text{Guess}_A(\cdot)$ of A is the function that on input $\lambda \in \mathbb{N}$ returns the maximum, over all b, i, m, of $\Pr[\mathbf{m}_b[i] = m]$, the probability over $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)]$. We say that A has *high min-entropy* if $\text{Guess}_A(\cdot)$ is negligible. We let $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\lambda) = 2\Pr[\text{IND}^A_{\mathsf{DE}}(\lambda)] - 1$ and say that DE is IND-secure if $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\cdot)$ is negligible for all PT A of high min-entropy. Let IND be the set of all IND-secure D-PKE schemes.

Let RE be a PKE scheme. Let $\mathsf{RE.rl}\colon \mathbb{N} \to \mathbb{N}$ denote its randomness-length function, meaning $\mathsf{RE.Enc}(1^\lambda, \cdot, \cdot)$ draws its coins at random from $\{0,1\}^{\mathsf{RE.rl}(\lambda)}$. Let H be a family of functions with $\mathsf{H.IL} = \mathbb{N}$ and $\mathsf{RE.rl}(\lambda) \in \mathsf{H.OL}(\lambda)$ for all $\lambda \in \mathbb{N}$. Our standard-model instantiation of the ROM encrypt-with-hash transform of BBO07 [6] associates to RE and H the (standard-model) D-PKE scheme $\mathsf{DE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$ described in Fig. 5. The message length of DE is that of RE. The following theorem says that the transform yields an IND-secure D-PKE scheme if H is UCE1-secure and RE is IND-CPA-secure. Here IND-CPA denotes the set of all IND-CPA-secure PKE schemes.

Theorem 2. If $\mathsf{H} \in \mathsf{UCE1}$ and $\mathsf{RE} \in \mathsf{IND\text{-}CPA}$ then $\mathsf{EwH}[\mathsf{H}, \mathsf{RE}] \in \mathsf{IND}$.

The proof of Theorem 2 is in [11]. Here we give a sketch. Let $\mathsf{PKE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$. Given a high min-entropy adversary $A = (A_1, A_2)$ for game $\mathrm{IND}^A_{\mathsf{PKE}}(\lambda)$, we build a source S and distinguisher D as follows. The source $S^{\text{HASH}}(1^\lambda)$ picks $(ek, dk) \leftarrow\!\!\$\ \mathsf{RE.Kg}(1^\lambda)$ and $d \leftarrow\!\!\$\ \{0,1\}$. It runs $A_1(1^\lambda)$ to get $(\mathbf{m}_0, \mathbf{m}_1)$ and lets $n \leftarrow |\mathbf{m}_d|$. For $i = 1, \ldots, n$ it obtains coins $\mathbf{r}[i]$ by calling its HASH oracle with $ek \| \mathbf{m}_d[i], 1^{\mathsf{RE.rl}(\lambda)}$. It then creates ciphertexts $\mathbf{c}[i] \leftarrow \mathcal{E}(1^\lambda, ek, \mathbf{m}_d[i]; \mathbf{r}[i])$ for $i = 1, \ldots, n$. It would like now to run A_2 on $\mathbf{c}$ but cannot since A_2 needs the public key, which includes hk. Accordingly, S returns as leakage $L \leftarrow (ek, d, \mathbf{c})$. Distinguisher $D(1^\lambda, hk, L)$ can create public key (ek, hk). It now lets $d' \leftarrow\!\!\$\ A_2(1^\lambda, (ek, hk), \mathbf{c})$. If $d = d'$ it sets $b' \leftarrow 1$, else $b' \leftarrow 0$. It returns b'. When the challenge bit in game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\lambda)$ is $b = 1$, adversaries S, D are simulating game $\mathrm{IND}^A_{\mathsf{PKE}}(\lambda)$, so that $2\Pr[d' = d \,|\, b = 1] - 1 = \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\lambda)$. If $b = 0$ then A_2 is seeing ciphertexts under the randomized RE scheme, and the assumed IND-CPA security of RE can be used to show that $2\Pr[d' = d \,|\, b = 0] - 1$ is negligible. This will allow us to upper bound $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\cdot)$ by $2\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ plus a negligible amount. To conclude it suffices to show that $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible. This follows if we show that S is unpredictable. By Lemma 1 it suffices to show that S is simple-unpredictable. Since oracle queries of S include messages created by A_1, (simple) unpredictability may seem at first to follow from the high min-entropy assumption on A. However we will additionally exploit (once again) the assumed IND-CPA security of the randomized RE scheme. This is because the leakage contains the ciphertexts. Overall, we exploit the IND-CPA security of RE in two places, building two corresponding adversaries.

Acknowledgments. We thank the Crypto 2013 PC for their many valuable comments and suggestions. We thank Dan Boneh and Adam O'Neill for their comments.

References

1. Applebaum, B.: Key-dependent message security: Generic amplification and completeness. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 527–546. Springer, Heidelberg (2011)
2. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
3. Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: Yao, A.C.-C., Yao, A.C.-C. (eds.) ICS 2011. Tsinghua University Press (2011)
4. Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 423–444. Springer, Heidelberg (2010)
5. Bellare, M.: New proofs for NMAC and HMAC: Security without collision-resistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (2006)

6. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
7. Bellare, M., Boldyreva, A., Palacio, A.: An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 171–188. Springer, Heidelberg (2004)
8. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
9. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
10. Bellare, M., Hoang, V.T., Rogaway, P.: Adaptively secure garbling with applications to one-time programs and secure outsourcing. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 134–153. Springer, Heidelberg (2012)
11. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. Cryptology ePrint Archive (2013)
12. Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 296–312. Springer, Heidelberg (2013)
13. Bellare, M., Paterson, K., Thomson, S.: RKA Security beyond the Linear Barrier: IBE, Encryption and Signatures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 331–348. Springer, Heidelberg (2012)
14. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)
15. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
16. Bellare, M., Rogaway, P.: The exact security of digital signatures: How to sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996)
17. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
18. Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)
19. Boldyreva, A., Cash, D., Fischlin, M., Warinschi, B.: Foundations of non-malleable hash and one-way functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 524–541. Springer, Heidelberg (2009)
20. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
21. Boldyreva, A., Fischlin, M.: Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 412–429. Springer, Heidelberg (2005)
22. Boldyreva, A., Fischlin, M.: On the security of OAEP. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 210–225. Springer, Heidelberg (2006)
23. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)

24. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21(2), 149–177 (2008)
25. Boneh, D., Boyen, X.: Efficient selective identity-based encryption without random oracles. Journal of Cryptology 24(4), 659–693 (2011)
26. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
27. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
28. Boneh, D., Franklin, M.K.: Identity based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
29. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
30. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
31. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision diffie-hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
32. Brakerski, Z., Segev, G.: Better security for deterministic public-key encryption: The auxiliary-input setting. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 543–560. Springer, Heidelberg (2011)
33. Cachin, C., Micali, S., Stadler, M.A.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
34. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)
35. Canetti, R., Dakdouk, R.R.: Extractable perfectly one-way functions. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 449–460. Springer, Heidelberg (2008)
36. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC, pp. 209–218. ACM Press (May 1998)
37. Canetti, R., Goldreich, O., Halevi, S.: On the random-oracle methodology as applied to length-restricted signature schemes. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 40–57. Springer, Heidelberg (2004)
38. Canetti, R., Tauman Kalai, Y., Varia, M., Wichs, D.: On symmetric encryption and point obfuscation. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 52–71. Springer, Heidelberg (2010)
39. Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions (preliminary version). In: 30th ACM STOC, pp. 131–140. ACM Press (May 1998)
40. Dachman-Soled, D., Gennaro, R., Krawczyk, H., Malkin, T.: Computational extractors and pseudorandomness. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 383–403. Springer, Heidelberg (2012)
41. Dent, A.W.: Adapting the weaknesses of the random oracle model to the generic group model. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 100–109. Springer, Heidelberg (2002)

42. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449–466. Springer, Heidelberg (2005)
43. Douceur, J., Adya, A., Bolosky, W., Simon, P., Theimer, M.: Reclaiming space from duplicate files in a serverless distributed file system. In: Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002, pp. 617–624. IEEE (2002)
44. Fischlin, M.: A note on security proofs in the generic model. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 458–469. Springer, Heidelberg (2000)
45. Fouque, P.-A., Pointcheval, D., Zimmer, S.: HMAC is a randomness extractor and applications to TLS. In: Abe, M., Gligor, V. (eds.) ASIACCS 2008, pp. 21–32. ACM Press (March 2008)
46. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. Journal of Cryptology 17(2), 81–104 (2004)
47. Fuller, B., O'Neill, A., Reyzin, L.: A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 582–599. Springer, Heidelberg (2012)
48. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 465–482. Springer, Heidelberg (2010)
49. Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
50. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33, 792–807 (1986)
51. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th FOCS, pp. 102–115. IEEE Computer Society Press (October 2003)
52. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)
53. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 308–326. Springer, Heidelberg (2010)
54. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
55. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006)
56. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
57. Hofheinz, D.: Possibility and impossibility results for selective decommitments. Journal of Cryptology 24(3), 470–516 (2011)
58. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
59. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (2010)
60. Kiltz, E., Pietrzak, K.: On the security of padding-based encryption schemes – or – why we cannot prove OAEP secure in the standard model. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 389–406. Springer, Heidelberg (2009)

61. Krawczyk, H.: Cryptographic extraction and key derivation: The HKDF scheme. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 631–648. Springer, Heidelberg (2010)
62. Malkin, T., Teranishi, I., Yung, M.: Efficient circuit-size independent public key encryption with KDM security. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 507–526. Springer, Heidelberg (2011)
63. Maurer, U.M., Renner, R.S., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
64. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)
65. Pass, R.: Limits of provable security from standard assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC, pp. 109–118. ACM Press (June 2011)
66. Pinkas, B., Schneider, T., Smart, N.P., Williams, S.C.: Secure two-party computation is practical. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 250–267. Springer, Heidelberg (2009)
67. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of the indifferentiability framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011)
68. Wichs, D.: Barriers in cryptography with weak, correlated and leaky sources. In: ITCS, 2013. Cryptology ePrint Archive, Report 2012/459 (2013)