Modeling Random Oracles Under Unpredictable Queries

Pooya Farshim[1(✉)] and Arno Mittelbach[2]

[1] ENS, CNRS & INRIA, PSL Research University, Paris, France
pooya.farshim@gmail.com
[2] Darmstadt University of Technology, Darmstadt, Germany
arno.mittelbach@cased.de

Abstract. In recent work, Bellare, Hoang, and Keelveedhi (CRYPTO 2013) introduced a new abstraction called Universal Computational Extractors (UCEs), and showed how they can replace random oracles (ROs) across a wide range of cryptosystems. We formulate a new framework, called Interactive Computational Extractors (ICEs), that extends UCEs by viewing them as models of ROs under unpredictable (aka. high-entropy) queries. We overcome a number of limitations of UCEs in the new framework, and in particular prove the adaptive RKA and semi-adaptive KDM securities of a highly efficient symmetric encryption scheme using ICEs under key offsets.

We show both negative and positive feasibility results for ICEs. On the negative side, we demonstrate ICE attacks on the HMAC and NMAC constructions. On the positive side we show that: (1) ROs are indeed ICE secure, thereby confirming the structural soundness of our definition and enabling a finer layered approach to protocol design in the RO model; and (2) a modified version of Liskov's Zipper Hash is ICE secure with respect to an underlying fixed-input-length RO, for appropriately restricted classes of adversaries. This brings the first result closer to practice by moving away from variable-input-length ROs. Our security proofs employ techniques from indifferentiability in multi-stage settings.

Keywords: Random oracle · Unpredictability · UCE · RKA security · KDM security · Zipper Hash · Indifferentiability · Multi-stage security

1 Introduction

1.1 Background

Since their formal introduction by Bellare and Rogaway [BR93], random oracles (ROs) have found many applications across a wide range of cryptographic protocols. However, due to an uninstantiability result of Canetti et al. [CGH98], which shows that certain (artificial) protocols become insecure as soon as the random oracle is replaced by *any* concrete hash function, reliance on ROs has also become somewhat debatable.

T. Peyrin (Ed.): FSE 2016, LNCS 9783, pp. 453–473, 2016.
DOI: 10.1007/978-3-662-52993-5_23

Two lines of research have been directed at dealing with such uninstantiability results. One is to construct standard-model counterparts of cryptographic primitives designed in the RO model (ROM). This approach comes with the drawback that the resulting cryptosystems often tend to be complex and achieve a lower level of security and/or efficiency. A second, more modular, approach aims to formulate abstractions of the proof-centric properties of random oracles such as extractability, programability, or non-malleability [Can97, CD09, Nie02, CD08, BCFW09]. Assuming that a hash function meets the introduced model, one proceeds to show that it can safely replace the random oracle in a protocol. These formalizations, however, have only been successful to a limited extent, and the question of finding a flexible and general framework that could be applied across a broad range of security goals and protocols remained open until recently.

1.2 UCE Security

Bellare, Hoang, and Keelveedhi (BHK) [BHK13a] revisit the above questions and present a powerful framework called *Universal Computational Extractors* (UCEs) that allow to securely instantiate random oracles in an interesting and diverse set of applications. These include, among other things, security under key-dependent-message (KDM) attacks, security under related-key attacks (RKAs), simultaneous hard-core bits, point function obfuscators, garbling schemes, proofs of storage, deterministic encryption, and message-locked encryption, thereby going far beyond what was previously possible.

Behind UCEs lies a new way to model the indistinguishability of a keyed hash function from a random oracle. Indeed, there are two direct ways to (incorrectly) model the security of a hash function:

(1) Provide the adversary with the hash key and ask it to distinguish an oracle implementing the hash function from one implementing the random oracle. This approach immediately fails as this game can be trivially won with the knowledge of the hash key by computing a hash value and checking the answer against the oracle's answer for the same query.
(2) Adopt the above approach, but now hide the hash key. This leads to PRF security—for which feasibility results are known—but is not useful in the context of hashing as the hash key is typically publicly known.

BHK overcome the above shortcomings by *splitting* the attacker into two parts and *constraining* the communication between the two. The first UCE attacker does *not* get to see the hash key, but has oracle access to either the hash function under a random key or the random oracle according to a random bit. The second attacker, on the other hand, *does* get to see the hash key, but can no longer access the oracle, and it has to guess the bit; see Fig. 1 (left). The two stages of the adversary can communicate only in restricted ways since arbitrary communication would lead to an attack similar to that given above for formulation (1).

More formally, for a keyed hash function H, UCE security is defined via a two-stage game consisting of algorithms S and D, called the *source* and the

distinguisher respectively, as follows. In the first stage, the source is given access to an oracle HASH that depending on a random bit b implements either the random oracle or the concrete hash function H under a random hash key hk. The source terminates by outputting some leakage L, which is then communicated to the second-stage distinguisher D. In addition to leakage L, the distinguisher also gets the hash key hk as input. The distinguisher's task is to guess b, i.e., guess whether the source was talking to the random oracle or the hash function. The UCE advantage of the pair (S, D) is defined as usual to be the probability of correctly guessing the bit b scaled away from one-half. We refer the reader to the original work [BHK13b] for an excellent overview of this approach to modeling hash-function security.

To see that without further restrictions UCE security cannot be achieved, consider a source that leaks one of its oracle queries together with the corresponding oracle answer to the distinguisher. The distinguisher then simply recomputes the hash value on the queried point—the distinguisher knows the hash key—and compares it to the leaked value.

In their original work, BHK [BHK13a] define two restrictions on sources: *computational unpredictability* and *computational reset security*. In the computational unpredictability game, it is required that when the source is run with a *random oracle* its leakage does not computationally reveal any of its queries. This is formalized by requiring that the probability of any efficient predictor P in guessing a query of S when given L is negligible.

The class of computationally unpredictable sources is denoted by $\mathcal{S}^{\text{cup}}$, and the resulting UCE security UCE[$\mathcal{S}^{\text{cup}}$] (aka. UCE1) of a hash function is defined by requiring the advantage of any efficient pair (S, D) with an *unpredictable* $S \in \mathcal{S}^{\text{cup}}$ in the UCE game to be negligible. Reset security imposes a weaker restriction on the source class and leads to the stronger UCE2 notion.

UCE security has been the subject of many recent studies. Brzuska, Farshim, and Mittelbach (BFM) [BFM14] show that, under new cryptographic assumptions, these restrictions are insufficient for a feasible definition. More precisely, assuming the existence of indistinguishability obfuscators [BGI+01, GGH+13], BFM show that the UCE[$\mathcal{S}^{\text{cup}}$] security of *any* hash function can be broken in polynomial time. To overcome this attack, BFM [BFM14] (and subsequently BHK in an updated version of their paper [BHK13b]) propose a *statistical* notion of unpredictability whereby the predictor can even run in unbounded time. Following the attack, BHK also refine the UCE notions based on computational unpredictability and introduce the classes of *bounded parallel* and *split* sources.[1] BFM show that security against bounded parallel source is also infeasible [BFM14], and recently attacks against split sources have also been shown [BST15].

On the positive side, Brzuska and Mittelbach [BM14b, BM15] show how to construct UCEs for the class of *strongly* unpredictable and statistically unpredictable sources for bounded number of queries. Bellare et al. [BHK14] develop

[1] Such computational UCE notions are intrinsically needed for applications such as simultaneous had-core bits and deterministic PKEs.

domain extenders for UCEs, and Bellare and Hoang [BH15] construct deterministic PKEs from UCEs for statistically unpredictable sources and lossy trapdoor functions. BFM [BFM14] have shown that the existence of obfuscation-based attacks against statistically unpredictable sources violates well-known impossibility results. A number of recent works have shown how to use UCEs as RO replacements in other protocols [MH14, BK15, DGG+15].

Despite the above advances, and irrespective of the restrictions imposed on sources, the UCE framework is intrinsically limited in a number of aspects: it only allows the source to place HASH queries which are *independent* of the hash key; after leakage is communicated from the source to the distinguisher no further HASH oracle queries can be made, and hence hash queries are inherently *non-adaptive*; UCEs cannot model *unkeyed* hash functions nor hash functions with weak keys where the key does not come from the uniform distribution. Motivated by these shortcomings, and the ultimate goal of basing the security of highly efficient and practical protocols on well-defined and feasible properties of random oracles, we set out to formalize an enhanced framework for the study ROM protocols.

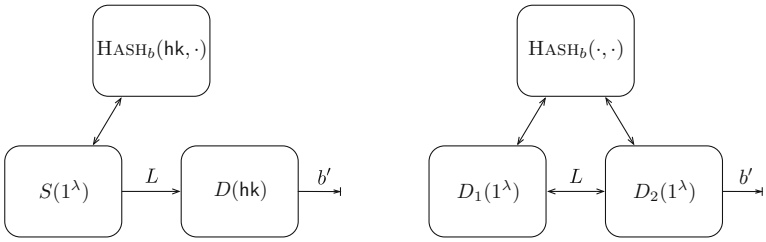

Fig. 1. The interactions in the UCE game (left) and the ICE game (right).

1.3 Interactive Computational Extractors

Given the development of UCEs, defining an extended model which meets the above-mentioned specifications is an intricate task. Indeed, well before the emergence of obfuscation-based attacks, BHK [BHK13b, p. 9] warned that extending UCEs to an interactive setting is "a dangerous path to tread." As an example, assume that we introduce a bi-directional communication channel between the distinguisher and the source so that our adaptivity targets are met. This extension can be shown to fall prey to somewhat non-trivial attacks that utilize general-purpose multi-party computation (MPC) protocols. Suppose the source S holds a random input x whose hash is y, and D holds hk. The two parties then run an MPC protocol to compute the Boolean value $y = \mathsf{H}(hk, x)$. The distinguisher finally returns this value as its guess. This attack would meet any reasonable notion of computational unpredictability since the security of the MPC protocol would ensure that the parties learn no more than what can be

deduced from their individual private inputs.[2] Allowing hash queries to depend on the hash key hk is also challenging since similarly to approach (1) above access to both hk and the hash oracle would trivialize the notion. For similar reasons, formulating a UCE-like model for unkeyed hash functions is also non-trivial. As we shall see, other forms of attacks also arise that should be ruled out for a feasible model.

The ICE framework. Let us call an input (hk, x), consisting of the hash key hk and a domain point x, to a hash function a *full* input. One way to view UCEs is that they adopt the indistinguishability-based approach (1) above, but restrict hash queries so that full inputs remain hidden from the attacker(s). It is clear that such hidden queries are not meaningful in the presence of a single adversary—any adversary knows its own queries—and hence UCEs come with two adversaries. Unpredictability together with denial of oracle access to D ensures that the x components of full inputs remain hidden from D. On the other hand, the hk components of full inputs remain hidden from S as the source is denied access to hk (and no communication from D to S is allowed). As a result, full inputs (hk, x) remain hidden from *both* parties involved in a UCE attack.

This perspective allows us to build on UCEs and extend them as follows. In our new framework, which we call *Interactive Computational Extractors* (ICEs),[3] a general mechanism for the joint generation of full inputs is enabled and adversarial restrictions that formalize what it means for full hash inputs to have high entropy are imposed.

We let two distinguishers (D_1, D_2) to take part in an attack, and allow them to communicate via a bi-directional channel. Both distinguishers get access to a challenge hash oracle, which depending on a challenge bit implements either the real hash function or a (keyed) random oracle. To enable the two parties to make hidden queries, we introduce a shared write-only tape that both D_1 and D_2 can write onto. When a distinguisher queries the hash oracle, the (real or ideal) hash of the full contents of the tape is returned. In contrast to UCEs, D_1 or D_2 can generate a hash key and perhaps modify it throughout the attack. This attack scenario is symmetric for D_1 and D_2 and, without loss of generality, the game terminates by D_2 outputting with a bit. (Our formal definition, however, comes with a slightly more general return statement.) For a class $\mathcal{C}$ of distinguishers, we define ICE[$\mathcal{C}$] security by demanding that the probability of guessing the challenge bit for any $D = (D_1, D_2) \in \mathcal{C}$ is negligibly close to 1/2. See Fig. 1 (right) for a summary of this interaction.

Entropic queries. Similarly to UCEs, the ICE notion cannot be achieved without constraining the way the two distinguishers communicate. The main restriction that we introduce is analogous to *statistical* unpredictability for UCEs:

[2] This can be viewed as an interactive analogue of BFM's attack [BFM14].

[3] In UCEs, "universal" refers to the fact that extraction should work with respect to universal (i.e., all admissible) sources. Analogously, "interactive" in ICEs refers to the fact that extraction should work for sources that can interact.

we demand the statistical unpredictability of *full* inputs to the hash function, including the hash key hk, from each distinguisher's point of view. We choose a statistical, rather than a computational, notion so that our definitions do not become subject to the interactive versions of the attacks highlighted in [BFM14].[4] More precisely, we require that when the hash oracle implements a keyed random oracle, no (possibly unbounded) predictor can guess a full input (hk, x) used to compute a hash value when it is provided with a distinguisher's *view* consisting of its inputs, random coins, and all incoming messages and oracle responses.

Since our framework allows oracle access to both parties, unlike UCEs the two distinguishers can implicitly communicate via *hash patterns* as follows. Suppose D_2 wants to leak a bit d to D_1. Algorithm D_2 starts by writing a random string onto the second half of the input and hands over the attack to D_1. Algorithm D_1 writes a random value to the first half of the input, calls HASH to receive a first hash value h_1, and hands over the attack back to D_2. Now algorithm D_2, according to the value of d, either modifies the contents of the second half of the input tape or leaves them unchanged. D_1 can recover d by obtaining a second hash value h_2 and checking if $(h_1 = h_2)$. The two distinguishers can also communicate via a *bit-fixing* attack: D_2 samples many (unpredictable) random values x conditioned on its hash value beginning with bit d, which D_1 can then recover via a hash query.

In our unpredictability definition the predictor gets to see all hash responses, and hence if there are any repetitions they will be seen by the predictor. Unpredictability will therefore ensure that such repetition patterns will not leak any of the queries. Sometimes, however, we need to explicitly disallow any repeat queries to enable a security proof to go through. In such a scenario, we can ensure that there is no leakage via hash patterns either. Repeat-freeness appears in other related settings such as related-key attacks or correlated-input hashing [BK03, GOR11].

1.4 Applications

BHK [BHK13a] use UCEs to show that the encryption scheme of Black, Rogaway, and Shrimpton (BRS) [BRS03] is secure under related-key attacks (RKAs) and key-dependent-message (KDM) attacks as long as the related keys/key-dependent messages are derived non-adaptively at the onset and without access to the hash key or previous ciphertexts.[5]

As we shall see, ICE encompasses UCE as a special case, and the BRS scheme can also be instantiated under the above models using ICEs. We can however also obtain feasibility results that are outside the reach of UCEs. A practically

[4] This is also motivated by impossibility results for statistically secure two-party protocols.

[5] Recall that in RKA security the adversary can see encryptions of messages under keys $\phi(K)$ for a random K and functions ϕ of its choice. In KDM security the adversary can see encryptions of $\phi(K)$, under a random key K, for ϕ's of its choice.

relevant and desirable level of RKA security is that corresponding to key offsets (the so-called xor-RKA security [LRW02,BK09]). We show that ICEs are sufficient to prove the *full* xor-RKA security of the BRS scheme. Our formal result is more general and applies to the larger class of *split* functions that take the form $\phi(K_1\|K_2) = \phi_1(K_1)\|\phi_2(K_2)$. (Such functions have been used to build RKA-secure PRFs [BC10], and also appear in other related contexts [CG14,LL12].) In addition to achieving stronger security guarantees, ICEs allow instantiating the BRS scheme using unkeyed hash functions, which is arguably closer to the original formulation of BRS.[6]

We also strengthen the attainable KDM security guarantees for BRS by showing that adversaries can choose key-dependent messages adaptively based on the hash key and also semi-adaptively depending on previous ciphertexts. We prove that ICEs are adaptively correlated-input secure [GOR11] and that they relate well to other standard security properties of random oracles, such as pseudorandomness, randomness extraction, and one-way security (see full version). We leave it as open questions to see if full RKA beyond xor offsets or full KDM security can be established using extractor-like notaions.

1.5 Instantiations

BHK show that random oracles fulfill their strongest proposed UCE notion, namely UCE security with respect to computationally unpredictable sources.[7] We prove that random oracles are also ICE secure. The significance of these results are twofold [BHK13a]: (1) there are no generic attacks on ICEs and the model is structurally sound; and (2) a *layered* approach to security analysis can be enabled, whereby one first proves the security of a scheme under an ICE assumption and then applies the RO model feasibility result. The latter is akin to security analyses carried out in the generic group model.

Practical hash functions, however, are not monolithic objects and often follow an iterative procedure to convert a fixed-input-length random oracle (FIL-RO) into a variable-input-length random oracle (VIL-RO). This, in turn, raises the question whether or not the above result can be brought closer to practice by demonstrating positive feasibility results for VIL-ICEs in the FIL-RO model. A seemingly immediate way to establish this result would be to start with a hash function that is known to be *indifferentiable* from a VIL RO (e.g., the HMAC or the NMAC construction), and then apply the RO feasibility result above to conclude. This argument, however, fails as the ICE game is *multi-staged* and indifferentiability does not necessarily guarantee composition in such settings [RSS11].

Motivated by the above observations, we show both positive and negative feasibility results for ICEs. On the negative side, we show that the indifferentiable HMAC and NMAC constructions are provably ICE *in*secure in the FIL-RO

[6] BRS [BRS03] analyze their scheme in the unkeyed RO model, which translates to unkeyed instantiations in practice.

[7] Note that this does not contradict the BFM attack as ROs do not have succinct descriptions.

model. On the positive side, and building on Mittelbach's techniques [Mit14], we prove that a keyed version of Liskov's Zipper Hash [Lis07] *is* ICE secure (as a VIL hash function) under the assumption that the underlying compression function is a FIL-RO. Zipper Hash can be seen as a variant of the classical Merkle–Damgård [Dam90, Mer90] construction where the message blocks are processed twice in the forward and backward directions. Hence our results strengthen the VIL-RO feasibility result above, and also provide formal evidence for the (intuitive) added security guarantees that multi-pass hash functions seem to offer over their single-pass counterparts. For instance, combined with our RKA and KDM results, we may conclude that Zipper Hash can be safely used within the BRS scheme with no adverse affects on its security.

The above analysis can be further strengthened in at least two directions. First, one can weaken the underlying assumption and assume that the compression function underlying Zipper Hash is only a FIL-ICE (rather than a FIL-RO). To this end, BHK [BHK14] give domain extenders for UCEs. Second, and motivated by the standard-model realizations of ICEs and UCEs, we ask if these primitives can be based on plausible hardness assumptions. Brzuska and Mittelbach [BM14a, BM15] have recently shown positive results for UCEs with respect to restricted classes of sources.

2 Notation

We denote the security parameter by $\lambda \in \mathbb{N}$, which is implicitly given to all algorithms (if not explicitly stated so) in the unary representation 1^λ. By $\{0,1\}^\ell$ we denote the set of all bit strings of length ℓ and $\{0,1\}^*$ is the set of all finite-length bit strings. For $x, y \in \{0,1\}^*$ we denote their concatenation by $x \| y$, the length of x by $|x|$, the ith bit of x by $x[i]$, and the substring of x formed using bits i to j by $x[i..j]$. We denote the empty string by ε. For X a finite set, $|X|$ denotes its cardinality, and $x \leftarrow\!\!\$\, X$ denotes the action of sampling x uniformly at random from X. If Q is a list and x a string then $\mathsf{Q} : x$ denotes the list obtained by appending x to Q. Similarly, If Q_1 and Q_2 are lists, then $\mathsf{Q}_1 : \mathsf{Q}_2$ denotes the concatenated list. Unless stated otherwise, algorithms are assumed to be randomized. We call an algorithm efficient or PPT if it runs in time polynomial in the security parameter. By $y \leftarrow \mathcal{A}(x; r)$ we denote that y was output by algorithm $\mathcal{A}$ on input x and randomness r. If $\mathcal{A}$ is randomized and no randomness is specified, then we assume that $\mathcal{A}$ is run with freshly sampled uniform random coins, and write $y \leftarrow\!\!\$\, \mathcal{A}(x)$. We use $\mathsf{Coins}[A]$ to denote the polynomially long string of random coins r used by a PPT machine A. We say a function $\mathrm{negl}(\lambda)$ is negligible if $\mathrm{negl}(\lambda) \in \lambda^{-\omega(1)}$.

Hash functions. In the line with [BHK13a], we consider the following (simplified) formalization of hash functions. A hash function consists of five PPT algorithms $\mathsf{H} := (\mathsf{H.Kg}, \mathsf{H.Ev}, \mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol})$ as follows. The key-generation algorithm $\mathsf{H.Kg}$ gets the security parameter 1^λ as input and outputs a key $hk \in \{0,1\}^{\mathsf{H.kl}(\lambda)}$, where $\mathsf{H.kl}(\lambda)$ is the key-length function. Algorithm $\mathsf{H.il}(\lambda)$ outputs the length of admissible inputs, which could take the special value $*$

denoting the variable-length input space $\{0,1\}^*$. Algorithm $\mathsf{H.ol}(\lambda)$ outputs the length of admissible outputs, which we assumed to be a fixed polynomial function of the security parameter. The deterministic evaluation algorithm $\mathsf{H.Ev}$ takes as input the security parameter 1^λ, a key hk, a point $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$, and generates a hash value $\mathsf{H.Ev}(1^\lambda, hk, x) \in \{0,1\}^{\mathsf{H.ol}(\lambda)}$. To ease notation, we often suppress the security parameter and simply write $\mathsf{H.Ev}(hk, x)$.

3 The ICE Framework

In this section we precisely define the ICE framework. We refer the reader to the introduction for a high-level overview of the model.

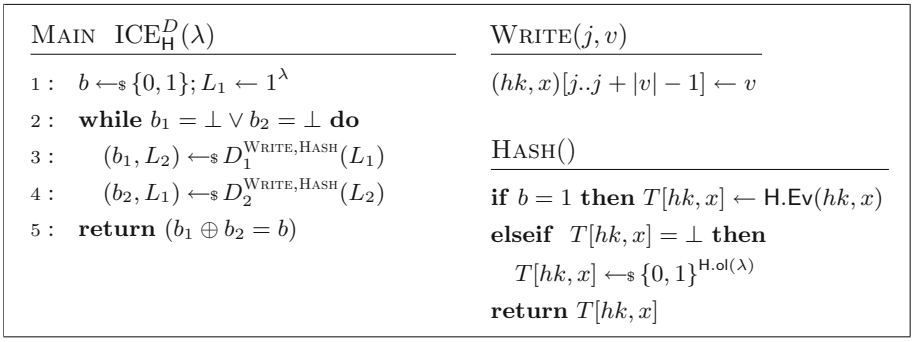

Fig. 2. The ICE game with respect to hash function H and distinguishers $D = (D_1, D_2)$. We have omitted the initialization of various variables for readability.

THE ICE GAME. Let $\mathsf{H} = (\mathsf{H.Kg}, \mathsf{H.Ev}, \mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol})$ be a hash function and let $D = (D_1, D_2)$ be a pair of algorithms. We define the ICE advantage of D against H as

$$\mathbf{Adv}^{\mathrm{ice}}_{\mathsf{H},D}(\lambda) := 2 \cdot \Pr\left[\mathrm{ICE}^D_{\mathsf{H}}(\lambda)\right] - 1,$$

where game $\mathrm{ICE}^D_{\mathsf{H}}(\lambda)$ is shown in Fig. 2. As mentioned in the introduction, we may assume, without loss of generality, that the game termites by D_2 outputting a bit. However, in order to preserve the symmetry of the definition (which will simplify our adversarial restrictions later on) and for added generality, we let the distinguishers jointly guess the challenge bit by computing $b_1 \oplus b_2$, where b_i is D_i's guess. The interaction terminates when both distinguishers return non-$\bot$ values for b_1 and b_2. For a class $\mathcal{C}$ of distinguishers, we define ICE[$\mathcal{C}$] security by requiring the advantage of any adversary $D \in \mathcal{C}$ to be negligible in the ICE game.

We require (D_1, D_2) not to leave any superfluous blank spaces on the joint tape. That is, a WRITE call must ensure that before the HASH oracle is called there do not exist indices $i < j$ such that $x[i] = \varepsilon \neq x[j]$ or $hk[i] = \varepsilon \neq hk[j]$.

We also demand that the full inputs (hk, x) are valid in the sense that prior to a HASH call $hk \in \{0,1\}^{\mathsf{H.kl}(\lambda)}$ and $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$. Although the distinguishers D_1 and D_2 are in general stateful algorithms, we omit the explicit handling of state values from the inputs and outputs of D_i.

RESTRICTIONS. As discussed in the introduction, the ICE model is not feasible unless additional restrictions on the distinguishers are imposed. We formulate our restrictions as joint properties of (D_1, D_2). Before presenting our main restrictions corresponding to high-entropy queries, we give a set of basic classes that will be useful in studying ICEs. As an example, for polynomials w, q, and r we define $\mathcal{C}_i^{w,q,r}$ to be the set of all (D_1, D_2) such that when (D_1, D_2) is run in the ICE game conditioned on $b = 0$ (i.e., with respect to the random oracle), the distinguisher D_i places at most $w(\lambda)$ queries to WRITE, at most $q(\lambda)$ queries to HASH, and terminates after at most $r(\lambda)$ invocations. We formalize a number of other notions below and omit the preamble "The set of all (D_1, D_2) such that when (D_1, D_2) is run in ICE with $b = 0$, we have with overwhelming probability that" from their definitions. Note that the classes below depend on $i \in \{1, 2\}$. For classes $\mathcal{C}_i^{\text{label}}$ we define $\mathcal{C}^{\text{label}} := \mathcal{C}_1^{\text{label}} \cap \mathcal{C}_2^{\text{label}}$. In the following table we present several restrictions that we will be using throughout this paper.

Class	Description
$\mathcal{C}_i^{w,q,r}$	D_i places at most $w(\lambda)$ queries to WRITE, at most $q(\lambda)$ queries to HASH, and terminates after at most $r(\lambda)$ invocations
$\mathcal{C}_i^{\text{poly}}$	D_i makes polynomially many oracle queries
$\mathcal{C}_i^{\text{ppt}}$	D_i runs in polynomial time on each invocation and terminates after a polynomial number of rounds
$\mathcal{C}_i^{0}$	D_i sets $b_i := 0$ in all invocations
$\mathcal{C}_i^{\varepsilon}$	D_i sets $L_{3-i} := \varepsilon$ in all invocations
$\mathcal{C}_i^{\text{0-hk}}$	D_i never writes onto the hk part of the tape
$\mathcal{C}_i^{\text{1-hk}}$	On its first invocation, D_i writes a random hk onto the hk-part of the tape. In subsequent invocations, D_i never writes onto the hk-part of the tape
$\mathcal{C}_i^{\text{dist}}$	D_i makes distinct queries to HASH. That is, for lists Q_1 and Q_2 defined in Fig. 3, the *combined* list $\mathsf{Q}_1 : \mathsf{Q}_2$ is repetition-free. Note that $\mathcal{C}_i^{\text{dist}} = \mathcal{C}_{3-i}^{\text{dist}} = \mathcal{C}^{\text{dist}}$
$\mathcal{C}_i^{\text{sup}}$	The probability that any (possibly unbounded) predictor P can guess a full query of D_i is negligible. We call this the class of statistically unpredictable D_i. See Fig. 3 for the formal definition. Class $\mathcal{C}_i^{\text{cup}}$ is the computational analogue, where P is restricted to be ppt

AN EXAMPLE: UCE WITHIN ICE. We describe how UCEs can be captured within the ICE framework. Since ICE is more expressive a framework, we need

to (drastically) restrict the distinguishers. In modeling UCEs, we identify the UCE distinguisher with D_1 and the UCE source with D_2. All parties typically run in polynomial time and hence we restrict to $\mathcal{C}^{\text{ppt}} := \mathcal{C}_1^{\text{ppt}} \cap \mathcal{C}_2^{\text{ppt}}$. In UCEs, the source queries HASH on an unknown hash key. The distinguisher, on the other hand, gets to see the hash key. Thus, we let D_1 (which represents the distinguisher) write a random hk to the joint input and then hand the attack to D_2 on the first invocation, i.e., $D \in \mathcal{C}_1^{\text{1-hk}}$. We further restrict to $\mathcal{C}_1^{\varepsilon}$, as a UCE distinguisher does not leak. Since the UCE game only has a single round, we also restrict to $\mathcal{C}_1^{1,0,2}$ (one round is used to write the hk). Finally, the source does not take part in decision making and cannot modify the hash key: UCEs are modeled by ICE[$\mathcal{C}^{\text{uce}}$] where

$$\mathcal{C}^{\text{uce}} := \mathcal{C}^{\text{ppt}} \cap \mathcal{C}_1^{\text{1-hk}} \cap \mathcal{C}_1^{\varepsilon} \cap \mathcal{C}_1^{1,0,2} \cap \mathcal{C}_2^{0} \cap \mathcal{C}_2^{\text{0-hk}}.$$

Note that the above models UCEs without any additional restrictions on the source classes. Such requirements can be added on top by appropriately restricting $\mathcal{C}^{\text{uce}}$.

UNPREDICTABILITY. We now formally define what we mean by a D that has unpredictable (aka. high-entropy) queries. We focus on a statistical notion of unpredictability [BFM14,BST15].[8] We say $D = (D_1, D_2)$ is statistically unpredictable for the distinguisher i, and write $D \in \mathcal{C}_i^{\text{sup}}$, if the advantage of any unbounded predictor P defined by

$$\mathbf{Adv}_{i,D,P}^{\text{pred}}(\lambda) := \Pr\left[\text{Pred}_{i,D}^{P}(\lambda)\right],$$

is negligible, where game $\text{Pred}_{i,D}^{P}(\lambda)$ is shown in Fig. 3.

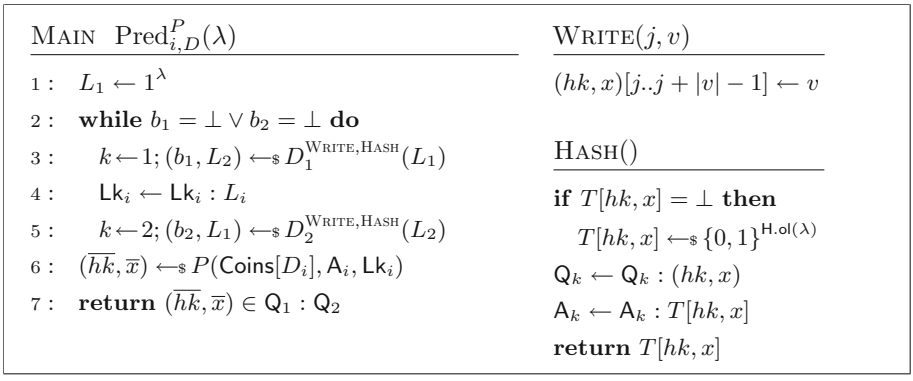

Fig. 3. The unpredictability game.

[8] We emphasize that computational notions are still valuable as combined with our feasibility results, they would enable easier and more modular security proofs in the RO model.

Note that the predicator only gets to see the hash responses for distinguisher D_i—these are within D_i's view—and has to guess a query made by *either* distinguisher in the concatenated list $\mathsf{Q}_1 : \mathsf{Q}_2$. It is easy to check that UCE security with respect to statistically unpredictable sources is equivalent to ICE$[\mathcal{C}^{\text{uce}} \cap \mathcal{C}^{\text{sup}}]$ security.

Remark. Since predictor P receives the full view of a distinguisher D_i, it can perfectly simulate a run of D_i in the ICE game with respect to a *random* implementation of the hash oracle, without any need to see the view of the partner distinguisher D_{3-i}. We will rely on this observation in our proofs.

4 Example Applications

In this section we demonstrate two example use cases of ICEs. Further applications are given in the full version and summarized in Table 1 below. These applications serve to demonstrate that many properties of random oracles that are useful in analyses of ROM cryptosystems can be modeled in a unified way within the ICE framework.

Table 1. Distinguisher classes used (above) and shown feasibility for (below). Here $\mathcal{C}^* := \mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{dist}} \cap \mathcal{C}_1^{\text{1-hk}} \cap \mathcal{C}_2^{\text{0-hk}} \cap \mathcal{C}_2^{\varepsilon}$.

Goal/Model	Class used/Achieved
Split RKA	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_2^0$
Split KDM	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_1^0$
Split/claw-free CIH	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_2^0$
Extractor	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_1^0 \cap \mathcal{C}^{\varepsilon} \cap \mathcal{C}^{1,1,2}$
Weak PRF	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_1^0 \cap \mathcal{C}^{\varepsilon} \cap \mathcal{C}^{\text{poly,poly},1}$
poly-regular OWF	$\mathcal{C}^* \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}_1^0 \cap \mathcal{C}^{1,1,1}$
VIL-ROM	$\mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{cup}}$ and $\mathcal{C}^{\text{poly}} \cap \mathcal{C}^{\text{sup}}$; both contain $\mathcal{C}^* \cap \mathcal{C}^{\text{sup}}$
FIL-ROM	$\mathcal{C}^* \cap \mathcal{C}^{\text{cup}}$, which contains $\mathcal{C}^* \cap \mathcal{C}^{\text{sup}}$

4.1 Split RKA Security

We show that the symmetric encryption scheme proposed by Black, Rogaway, and Shrimpton (BRS) [BRS03] is secure against related-key attacks (RKAs) when instantiated with an ICE-secure hash function. The encryption algorithm of the BRS scheme is implemented via $\mathsf{Enc}^{\mathsf{H}}(K, M; R) := (R, M \oplus \mathsf{H}(K \| R))$, for a hash function H, randomness R and key K. Recall that in an RKA, an adversary can obtain encryptions of messages of its choice under correlated keys (e.g., under K and $K \oplus 1$).

Split related-key derivation (RKD) functions ϕ decompose into two sub-RKD functions ϕ_1 and ϕ_2 that are applied in parallel to two (fixed) sub-strings of the key: $\phi(K_1\|K_2) = \phi_1(K_1)\|\phi_2(K_2)$.[9] Split functions capture many RKA cases of interest including the case of xoring constants into keys. Without the minimal assumption that ϕ's have unpredictable outputs (i.e., the guessing probability of the outputs of $\phi(K)$ over randomly chosen K is negligible) RKA security is not achievable [BK03]. In our proof, we will require a slightly stronger condition that the sub-RKD functions $\phi_1(K_1)$ and $\phi_2(K_2)$ are *individually* unpredictable. Note that offsetting keys via xor enjoys this property as xor induces a permutation over the two halves of the key.

BHK [BHK13a], by interpreting encryption randomness as hash keys, show that BRS is selectively RKA secure using a multi-key extension of UCE[$\mathcal{S}^{\text{cup}}$]. In contrast, the adversary in our model retains its capability to adaptively query RKD functions of its choice depending both on the hash key and the ciphertexts that it has previously seen. For this result, although ICE[$\mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{sup}}$] is sufficient, the assumption can be fine-tuned to ICE[$\mathcal{C}$] where

$$\mathcal{C} := \mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}^{\text{dist}} \cap \mathcal{C}_1^{\text{1-hk}} \cap \mathcal{C}_2^{\text{0-hk}} \cap \mathcal{C}_2^{0} \cap \mathcal{C}_2^{\varepsilon}.$$

We defer the formal proof to the full version and give a detailed outline here.

The ICE adversary. Given an RKA adversary A, we construct an ICE adversary (D_1, D_2), where D_1 handles the left components of A's RKA queries and D_2 handles the right components as follows.

$D_1(L_1)$: On initial invocation, generate a hash key hk, a random K_1, and a random bit b. Store these values and write hk onto the hk-part of the tape. Run $A(hk)$ to get an RKA query $((\phi_1, \phi_2), M_0, M_1)$. Output $(b_1, L_2) := (\perp, \phi_2)$. Proceed as follows in subsequent invocations. Generate and store a random R and write $\phi_1(K_1)$ onto the 1st segment (out of three segments) of the x-part of the tape and R onto its 3rd segment. Query Hash to get H. Recover R and resume A on $(R, H \oplus M_b)$ to get a new RKA query $((\phi_1, \phi_2), M_0, M_1)$, or a bit b'. If A outputs a bit b', return $(b_1, L_2) := (b = b', \varepsilon)$ and terminate. Else output $(b_1, L_2) := (\perp, \phi_2)$.

$D_2(L_2)$: When initially invoked, generate a random K_2 and store it. In all invocations (including the first), recover ϕ_2 from L_2. If $\phi_2 = \varepsilon$, return $(b_2, L_1) := (0, \varepsilon)$ and terminate. Else write $\phi_2(K_2)$ onto the 2nd segment of the x-part of the tape. Output $(b_2, L_1) := (0, \varepsilon)$.

Unpredictability. We show that $D \in \mathcal{C}$ for class $\mathcal{C}$ as defined above. To this end, we only prove membership in $\mathcal{C}^{\text{sup}} \cap \mathcal{C}^{\text{dist}}$ as other cases follow via syntactic checks. This follows from the following two observations: (1) The Hash queries are distinct with overwhelming probability since before each query a fresh random value R is written onto the joint tape. (2) The functions ϕ_1 and ϕ_2 are

[9] For simplicity we assume that these are just the left and right halves of the key. Our proof will however also apply to any two substrings of super-logarithmic lengths.

run on independently chosen substrings of the key. Since they are assumed to be individually statistically unpredictable, D_1 observing independently generated random strings corresponding to hash values never gets to know the contents of the tape written by D_2, and vice versa, D_2 never gets to know what is written on to the tape by D_1.

4.2 KDM Security

When the random oracle in the BRS scheme is instantiated with an ICE-secure hash function, we are able to show that the BRS scheme resists a partially adaptive form of KDM security for split key-dependent-message derivation (KDMD) functions ϕ. As for RKD functions, such KDMD functions consist of sub-KDMD functions ϕ_1 and ϕ_2 of the form $\phi(K_1\|K_2) := \phi_1(K_1)\|\phi_2(K_2)$. The adaptivity level that we can tolerate is as follows. In an initial phase of the attack, the adversary can fully adaptively query split KDMD functions that do not depend on K_2. That is, for these functions $\phi_2(K_2)$ is constant and independent of K_2 and its value can be predicted. In a second phase of the attack, the adversary can query split KDMD functions of its choice as long as either $\phi_1(K_1)$ is constant or $\phi_1(K_1)$ was used in the first phase. (We emphasize that these functions are not required to be unpredictable.) This model is strong enough to imply IND-CPA security (without any restrictions), a case that could not be treated using UCEs.

The ICE adversary. Let A be a KDM adversary against the BRS scheme in the model above. Our ICE$[\mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}^{\text{dist}}]$ adversary corresponding to A is as follows, where for simplicity we have assumed the lengths of keys, randomness and messages are all ℓ. (The ICE class can be further restricted as is shown in Table 1.) In this reduction, D_1 faithfully runs the first stage of the attack, while D_2 runs its second stage. To answer KDM queries, D_2 relies on the "homomorphic" property that $H \oplus (x_1\|x_2) = H \oplus (x_1\|0^{|x_2|}) \oplus (0^{|x_1|}\|x_2)$.

$D_1(L_1)$: When initially invoked, generate a random hk, K_1 and b and store them. Write hk to the hk-part and K_1 to the 1st (out of three) segments of the x-part of the tape. (The segments are of lengths $\ell/2$, $\ell/2$ and ℓ corresponding to K_1, K_2 and R respectively.) Output $(\perp, \varepsilon)$. On the second invocation, run $A(hk)$ and answer its KDM queries $((\phi_1^0, \phi_2^0), (\phi_1^1, \phi_2^1))$ as follows. Write a fresh random value R onto the 3rd segment of the x-part of the tape. Call Hash to get H, and resume A on $(R, H \oplus (\phi_1(K_1)\|M_2^*))$, where $M_2^* := \phi_2(0^{\ell/2})$ is the right K_2-independent part of the message. Continue this process until A decides to proceed to its second stage. Let st_A denote A's state. Generate sufficiently many copies $(R_1, C_1'), \ldots, (R_q, C_q')$ of each of the KDM queries made in the first phase. Let List_1 denote the corresponding list of queried ϕ_1^b. Return $(0, (b, st_A, (R_1, C_1'), \ldots, (R_q, C_q'), \mathsf{List}_1))$ and terminate.

$D_2(L_2)$: When initially invoked, generate a random K_2, store it, and write it to the 2nd segment of the x-part of the tape. Hand the attack back to D_1, by outputting $(\perp, \varepsilon)$. On the second invocation, parse L_2 appropriately as above. Resume A on st_A and answer its KDM queries $((\phi_1^0, \phi_2^0), (\phi_1^1, \phi_2^1))$ as follows.

If $\phi_1^b \in \mathsf{List}_1$ pick a fresh ciphertext (R, C') corresponding to ϕ_1^b and *complete* the ciphertext preparation by setting $C \leftarrow C \oplus (0^{\ell/2} \| \phi_2^b(K_2))$. Otherwise generate a random R, write it onto the 3rd segment of the x-part of the tape, query HASH to get H, and set $C \leftarrow H \oplus (\phi_1^b(0^{\ell/2}) \| \phi_2^b(K_2))$. Resume A on $(R, C; st_A)$ and continue in this manner until A outputs a bit b'. Return $(b = b', \varepsilon)$ and terminate.

UNPREDICTABILITY. D's queries are distinct with overwhelming probability as fresh randomness R is written on the tape before each query. Throughout the attack, and when the hash oracle implements a random function, K_2 remains hidden from D_1 as D_1 only sees distinct random values as hash responses. Key K_1 also remains hidden from D_2 as the (incomplete) ciphertext components received from D_1 are random strings. Hence $D \in \mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}^{\text{dist}}$.

5 Feasibility

In this section we start by showing that random oracles are ICE secure with respect to interesting distinguisher classes (in particular, with respect to the restrictions needed for the presented applications). We then consider the ICE security of practical hash constructions built from fix-input-length (FIL) ROs. In particular, we look at a keyed variant of Liskov's Zipper Hash [Lis07] and show that it achieves ICE security in the FIL-RO model. Interestingly, we show that both HMAC and NMAC constructions [BCK96], which were recently shown to be UCE secure in FIL-ROM [Mit14], fail to be ICE secure. This result yields a natural counterexample to the composability of HMAC in multi-stage settings, similarly to that given by Ristenpart, Shacham, and Shrimpton in [RSS11]. Furthermore, it provides a separation between ICE and UCE. Our results also demonstrate that Zipper Hash can provide a higher level of security compared to HMAC when used in multi-stage settings.

5.1 ICEs from Random Oracles

BHK [BHK13b] show that UCE-secure hash functions can be provably constructed in the RO model. The philosophical justifications of this result are that there are *no structural weaknesses* in the definitional framework, and more importantly, a *layered* approach to protocol design in the RO model can be enabled [BHK13b]. We show that ICEs also enjoy RO feasibility.

Let $\mathsf{H.kl}(\cdot)$ and $\mathsf{H.ol}(\cdot)$ be two arbitrary functions as in the syntax of a hash function. Let $\mathcal{R}$ be a family of variable-input-length (VIL) ROs (i.e., with domain $\{0,1\}^*$) and range $\{0,1\}^{\mathsf{H.ol}(\lambda)}$. We construct the required hash function $\mathsf{H}^{\mathcal{R}}$ by defining the key-generation algorithm $\mathsf{H.Kg}(1^\lambda)$ to return a random $hk \leftarrow_{\$} \{0,1\}^{\mathsf{H.kl}(\lambda)}$ and the evaluation algorithm $\mathsf{H.Ev}^{\mathcal{R}}(hk, x)$ to return $\mathcal{R}(hk \| x)$. Our first feasibility result is as follows.

Theorem 1 (ICE feasibility in ROM). *The VIL hash function* $\mathsf{H}^{\mathcal{R}}$ *constructed above is* ICE[$\mathcal{C}$] *secure in the VIL-RO model for* $\mathcal{R}$ *for the following (incomparable) classes of adversaries:*

$$\mathcal{C} := \mathcal{C}^{\mathrm{poly}} \cap \mathcal{C}^{\mathrm{sup}} \quad \textit{and} \quad \mathcal{C} := \mathcal{C}^{\mathrm{ppt}} \cap \mathcal{C}^{\mathrm{cup}}.$$

The proof of this theorem is similar to the proof of [BHK13b, Theorem 6.1] for UCEs, and we give the details in the full version. Intuitively, we rely on unpredictability of queries to simulate the random oracles used in the construction and implicit in the ICE game independently. Interestingly, distinctness of queries will not be needed in this proof and we do not restrict the classes to $\mathcal{C}^{\mathrm{dist}}$. We note that the above classes include all those needed for the applications, as listed in Table 1. We also note that this theorem generalizes the feasibility of UCEs for unpredictable sources in ROM [BHK13b] as it can be easily verified that

$$\mathcal{C}^{\mathrm{uce}} \cap \mathcal{C}_2^{\mathrm{cup}} \subseteq \mathcal{C}^{\mathrm{ppt}} \cap \mathcal{C}^{\mathrm{cup}} \quad \text{and} \quad \mathcal{C}^{\mathrm{uce}} \cap \mathcal{C}_2^{\mathrm{sup}} \subseteq \mathcal{C}^{\mathrm{ppt}} \cap \mathcal{C}^{\mathrm{sup}}.$$

5.2 VIL-ICEs from Ideal Compression

Practical variable-input-length (VIL) hash functions are not monolithic objects. They often follow iterative modes of chaining that convert a fix-input-length (FIL) compression function to one that accepts variable-length inputs. This design principle has been successfully validated via the indifferentiability framework of Maurer et al. [MRH04, CDMP05], whereby an indifferentiable hash-function construction is shown to securely compose when used in place of a random oracle. As pointed out in [RSS11], the indifferentiability framework only guarantees composition in single-stage environments. The ICE and UCE games, however, are inherently multi-staged and lie outside the reach of (plain) indifferentiability. Mittelbach [Mit14] develops new techniques to extend the reach of (plain) indifferentiability to certain classes of multi-stage games. In particular, he shows that the HMAC and NMAC constructions are UCE secure. Interestingly, we show that these results do not carry over to the ICE model: HMAC and NMAC provably fail to be ICE secure. On the other hand, we build on Mittelbach's techniques to prove that a variant of Zipper Hash [Lis07] is provably ICE secure.

Attacks on HMAC and NMAC. The HMAC and NMAC constructions are shown in Fig. 4. If we denote the iterated compression function used in HMAC by h, then it is easily seen that key hk is only used on the "outer" h-evaluations. Consider an ICE distinguisher D_1 which holds hk, computes the values

$$y_1 := \mathsf{h}(hk \oplus \mathsf{ipad}, \mathsf{IV}) \qquad \text{and} \qquad y_2 := \mathsf{h}(hk \oplus \mathsf{opad}, \mathsf{IV})$$

and sends them to distinguisher D_2. Given (y_1, y_2), distinguisher D_2 can compute the HMAC values for any $x \in \{0,1\}^*$ under hk. Thus, in order to win the ICE game, D_2 simply chooses a random x and writes it on the input tape, and calls Hash to receive a value y. It then locally recomputes $\mathsf{H}.\mathsf{Ev}^{\mathsf{h}}(hk, x)$ using the

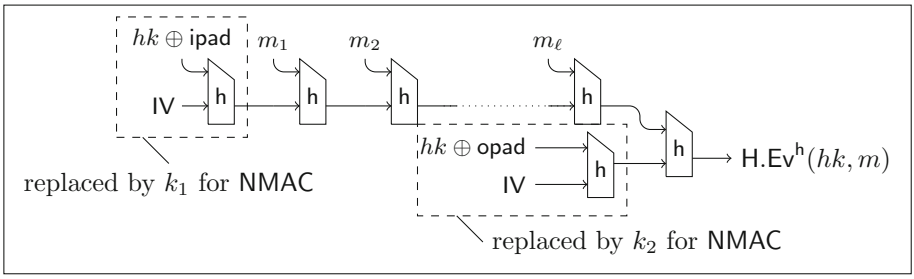

Fig. 4. The HMAC construction. If the dashed boxes are exchanged for independent keys k_1 and k_2, we obtain NMAC. Here we are ignoring padding.

compression function h and values (y_1, y_2). If the results match, it outputs 1, and else it outputs 0. It is easily seen that this adversary wins ICE with overwhelming probability. Furthermore, given (y_1, y_2), the hash key hk remains statistically hidden from D_2 (as the number of h queries is bounded by a polynomial). Value x, being random, also remains statistically hidden from D_1. Formally, this attack breaks ICE[$\mathcal{C}$] for

$$\mathcal{C} := \mathcal{C}^{\text{ppt}} \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}^{1,1,1} \cap \mathcal{C}_1^{\text{1-hk}} \cap \mathcal{C}_1^{0} \cap \mathcal{C}_2^{\text{0-hk}} \cap \mathcal{C}_2^{\varepsilon}.$$

Zipper Hash. The above attack raises the question if any iterative hash function can be ICE[$\mathcal{C}$] secure for a meaningful class of distinguishers $\mathcal{C}$. We show that a hybrid construction of a keyed version of Liskov's Zipper Hash construction [Lis07] and chopped Merkle–Damgård (chop-MD) of Coron et al. [CDMP05] is ICE secure. Zipper Hash can be regarded as a basic Merkle–Damgård scheme where the message is processed twice, the second time in reversed block order. chop-MD refers to the construction where a hash value consists only of the first half of the output bits of the final compression function. Our hybrid construction results from adding the chop step to Zipper Hash. Furthermore, we consider a keyed variant of Zipper Hash by prepending the hash key to the message. We assume that key length matches block length, which means that the *first and last* evaluations of the compression function operate on the hash key. We denote this keyed variant of Zipper Hash by chop-KZIP. Figure 5 shows a schematic diagram of the construction.

Theorem 2 (Zipper Hash's ICE security). *The VIL hash function* chop-KZIP$^{\mathsf{h}}$ *constructed above is* ICE[$\mathcal{C}$] *secure in the FIL-RO model for* $\mathsf{h} : \{0,1\}^{\mu} \times \{0,1\}^{n} \longrightarrow \{0,1\}^{n}$ *for the class*

$$\mathcal{C} := \mathcal{C}^{\text{poly}} \cap \mathcal{C}^{\text{sup}} \cap \mathcal{C}^{\text{dist}} \cap \mathcal{C}_1^{\text{1-hk}} \cap \mathcal{C}_1^{0} \cap \mathcal{C}_2^{\text{0-hk}} \cap \mathcal{C}_2^{\varepsilon}.$$

An analogous result holds for polynomial-time distinguishers that are only computationally unpredictable.

In the full version, we give the proof, where we also present a self-contained introduction to the unsplittability technique [Mit14].

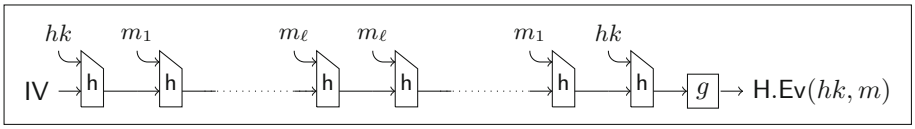

Fig. 5. The Zipper Hash construction merged with chop-MD [CDMP05] and keyed with hk. The final node g corresponds to the projection to the first half of the output of h.

Note that class $\mathcal{C}$ above contains that class used to attack HMAC and hence chop-KZIP$^{\mathsf{h}}$ provably achieves a higher level of security in multi-stage games. We note that the reach of the above feasibility result includes all applications scenarios listed in Table 1. In particular, chop-KZIP$^{\mathsf{h}}$ can security replace the random oracle in these applications. For this also note that we can easily drop $\mathcal{C}_1^0$ by requesting that in the last round D_1 outputs a guess for b which D_2 echoes. With the other restrictions present this change is without loss of generality.

This result cannot be strengthened for the (large) adversarial classes that were used in Theorem 1. To see this, consider two distinguishers that engage in a *distributed* computation of chop-KZIP$^{\mathsf{h}}$ hash values as follows. Distinguisher D_1 knows hk and m_1 and D_2 knows m_2, where message $m := m_1 \| m_2$ is being hashed. Distinguisher D_1 computes an intermediate hash digest using (hk, m_1) and forwards it to D_2. Distinguisher D_2 now computes another iteration of the hash using m_2 and forwards the result to D_1. Distinguisher D_1 can now complete the hash computation using its knowledge of (hk, m_1) and the intermediate hash digest that it receives.

A straightforward generalization of this attack also rules out multi-pass variants of chop-KZIP$^{\mathsf{h}}$ (where messages are processed multiple times in the forward and backward directions), including those whose number of passes is not fixed a priori and can depend on the number of message blocks. This is due to the fact that the number of rounds in an ICE attack is not fixed. This, in turn, raises the question if ICE$[\mathcal{C}^{\text{poly}} \cap \mathcal{C}^{\text{sup}}]$ is feasible in the FIL-RO model. We conclude the paper with a candidate construction that we conjecture to reach this level of security.

Mix Hash. Let $\mathsf{h} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a compression function. Let $m := m_1 \| \cdots \| m_\ell \in (\{0,1\}^n)^\ell$ be a message with ℓ blocks of length n each. Let $\mathsf{Mix}^{\mathsf{h}}(m)$ denote the transformation that maps m to $M := \|_i \|_j M_{i,j}$ where $M_{i,j} := \mathsf{h}(m_i, m_j)$ for $1 \le i < j \le \ell$. (Therefore M has $\ell(\ell-1)/2$ blocks.) Now let $hk \in \{0,1\}^n$ be a hash key and define

$$\mathsf{MixHash}^{\mathsf{h}}(hk, m) := \mathsf{HMAC}^{\mathsf{h}}(0^n, \mathsf{Mix}^{\mathsf{h}}(hk \| m)).$$

Note that MixHash$^{\mathsf{h}}$ places $\Theta(\ell^2)$ calls to its compression function h.[10] The design rationale behind MixHash$^{\mathsf{h}}$ is as follows. All intermediate digests values $M_{i,j}$ are

[10] Indeed, MixHash is a (highly) offline function: for $\alpha \in [0,1]$, it requires space roughly $n\ell\sqrt{1-\alpha}$ bits after a fraction α of the $n\ell(\ell+1)/2$ bits are processed.

needed in order to successfully compute a hash value. These values, however, consist of all pairs (m_i, m_j) compressed through h. Since h is a monolithic object, $M_{i,j}$ cannot be computed in a distributed way, a strategy that was used in all previous attacks. In other words, one of the distinguishers has to know (hk, m) in full and hence will violate unpredictability. To see this, suppose D_1 does not know m_j in full and D_2 does not know m_i in full for some $i < j$. Then there is no way for these parties to learn $M_{i,j} := \mathsf{h}(m_i, m_j)$ without one of them explicitly quarrying h on (m_i, m_j). This however means that both m_i and m_j are known to the quarrying party, which leads to a contradiction. We leave a formal analysis of $\mathsf{MixHash}^{\mathsf{h}}$ in the FIL-RO model for h as future work.

Acknowledgments. The authors would like to thank Christina Brzuska for taking part in the early stages of this work. Pooya Farshim was supported in part by grant ANR-14-CE28-0003 (Project EnBid).

References

[BC10] Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)

[BCFW09] 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)

[BCK96] 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)

[BFM14] Brzuska, C., Farshim, P., Mittelbach, A.: Indistinguishability obfuscation and UCEs: the case of computationally unpredictable sources. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 188–205. Springer, Heidelberg (2014)

[BGI+01] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)

[BH15] Bellare, M., Hoang, V.T.: Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (2015)

[BHK13a] Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (2013)

[BHK13b] Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424 (2013). http://eprint.iacr.org/2013/424

[BHK14] Bellare, M., Hoang, V.T., Keelveedhi, S.: Cryptography from compression functions: the UCE bridge to the ROM. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 169–187. Springer, Heidelberg (2014)

[BK03] Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)

[BK09] Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)

[BK15] Bellare, M., Keelveedhi, S.: Interactive message-locked encryption and secure deduplication. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 516–538. Springer, Heidelberg (2015)

[BM14a] Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 142–161. Springer, Heidelberg (2014)

[BM14b] Brzuska, C., Mittelbach, A.: Using indistinguishability obfuscation via UCEs. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 122–141. Springer, Heidelberg (2014)

[BM15] Brzuska, C., Mittelbach, A.: Universal computational extractors and the superfluous padding assumption for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2015/581 (2015). http://eprint.iacr.org/

[BR93] 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

[BRS03] 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)

[BST15] Bellare, M., Stepanovs, I., Tessaro, S.: Contention in cryptoland: obfuscation, leakage and UCE. Cryptology ePrint Archive, Report 2015/487 (2015). http://eprint.iacr.org/2015/487

[Can97] 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)

[CD08] 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)

[CD09] Canetti, R., Dakdouk, R.R.: Towards a theory of extractable functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009)

[CDMP05] Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: how to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)

[CG14] Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 440–464. Springer, Heidelberg (2014)

[CGH98] Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC, pp. 209–218. ACM Press, May 1998

[Dam90] Damgård, I.B.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)

[DGG+15] Dodis, Y., Ganesh, C., Golovnev, A., Juels, A., Ristenpart, T.: A formal treatment of backdoored pseudorandom generators. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 101–126. Springer, Heidelberg (2015)

[GGH+13] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press, October 2013

[GOR11] 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)

[Lis07] Liskov, M.: Constructing an ideal hash function from weak ideal compression functions. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 358–375. Springer, Heidelberg (2007)

[LL12] Liu, F.-H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 517–532. Springer, Heidelberg (2012)

[LRW02] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Heidelberg (2002)

[Mer90] Merkle, R.C.: One way hash functions and DES. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)

[MH14] Matsuda, T., Hanaoka, G.: Chosen ciphertext security via UCE. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 56–76. Springer, Heidelberg (2014)

[Mit14] Mittelbach, A.: Salvaging indifferentiability in a multi-stage setting. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 603–621. Springer, Heidelberg (2014)

[MRH04] 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)

[Nie02] 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)

[RSS11] 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)