A preliminary version of this paper appears in the proceedings of CRYPTO 2013. This is a revised and updated full version, available as IACR Cryptology ePrint Archive Report 2013/424.

Instantiating Random Oracles via UCEs

Mihir Bellare[1] Viet Tung Hoang[2] Sriram Keelveedhi[3]

September 22, 2013

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 asks that outputs of the function look random given some "leakage," even if the adversary knows the key, as long as the leakage is appropriately restricted.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-0904380, CCF-0915675, CNS-1116800 and CNS-1228890.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: vth005@eng.ucsd.edu. URL: http://csiflabs.cs.ucdavis.edu/~tvhoang/. Supported in part by NSF grants CNS-0904380, CCF-0915675, CNS-1116800 and CNS-1228890. Part of this work was done when Hoang was a Ph.D. student at University of California, Davis and supported in part by NSF grants CNS-0904380 and CNS-1228890.

[3] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: sriramkr@cs.ucsd.edu. URL: http://cseweb.ucsd.edu/~skeelvee/. Supported in part by NSF grants CCF-0915675 and CNS-1116800.

Contents

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) [23], 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 [49] 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.[1] But we show that it works across a fairly large, diverse and interesting spectrum of schemes and goals including deterministic PKE, message-locked encryption, hard-core functions, 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 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 leakage is appropriately restricted. Different restrictions give rise to different specific assumptions in the UCE family. Let us now step back to provide some background and then return to our contributions.

1.1 Background

The random-oracle paradigm of BR93 [23] 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) [49] 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 [98, 75, 13, 50, 60, 90].

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.

1.2 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 [73], 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.

[1] "Work" means allow the instantiated scheme to be proven secure, and "always works" means works for all schemes secure in the ROM. Indeed, we note that neither UCE nor any other achievable, standard-model security attribute of a family of functions can always work. This is implied by known impossibility results [49, 97].

Canetti [47] 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) [47, 52, 48] and non-malleable hash functions [29] have however proven of limited applicability [31]. Another direction has been to try to instantiate the RO in particular schemes like OAEP [24], again with limited success [32, 31] or under strong assumptions on RSA [89].

Our position is philosophically different from that of [47, 52]. 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.

1.3 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. For a class $\mathcal{S}$ of sources, let us say that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}]$, or is $\mathsf{UCE}[\mathcal{S}]$-secure, if, for all $S \in \mathcal{S}$ and all PT D, the advantage of S, D in figuring out b, in the game sketched above, is negligible.

Clearly, $\mathsf{UCE}[\mathcal{S}]$-security is not achievable if $\mathcal{S}$ is the class of *all* PT sources. For example, 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 seek, accordingly, classes $\mathcal{S}$ small enough that the assumption of $\mathsf{UCE}[\mathcal{S}]$-security (that is, that this set is non-empty) is plausible, yet large enough that the same assumption is useful for applications. The classes are obtained by restricting the source. Restrictions we consider include unpredictability, reset-security, splitting and run-time limitations, but many others are possible. Unpredictability of S requires that it be 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 S 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. We let $\mathcal{S}^{\text{cup}}$ and $\mathcal{S}^{\text{sup}}$ be the classes of PT sources unpredictable to PT and unbounded adversaries, respectively, leading to UCE classes $\mathsf{UCE}[\mathcal{S}^{\text{cup}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\text{sup}}]$. Reset-security gives rise, correspondingly, to $\mathsf{UCE}[\mathcal{S}^{\text{crs}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\text{srs}}]$. In the basic definitions, only a single hashing key is involved, and we also define mUCE, a multi-key analogue.

In the preliminary version of our work [18], we stated results for our applications under the assumption of security for the classes $\mathsf{UCE}[\mathcal{S}^{\text{cup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\text{crs}}]$, there called UCE1 and UCE2. However Brzuska, Farshim and Mittelbach (BFM) [45] showed that if indistinguishability obfuscation (iO) is possible [9, 68] then these forms of UCE are not achievable. Our proofs, however, had relied on weaker assumptions (classes) not made explicit in the theorem statements and not broken by the BFM attacks. In the current version of our paper, we update our theorem statements accordingly. The classes for which we now assume security include the statistical $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\text{srs}}]$ as well as $\mathsf{UCE}[\mathcal{S}]$ for certain subclasses $\mathcal{S}$ of $\mathcal{S}^{\text{cup}}$ and $\mathcal{S}^{\text{crs}}$. In Section 4 we formally define these classes, discuss the plausibility of assuming their security, and explain the relation to extractors that inspired the UCE name. We use the term UCE-security loosely to refer to some assumption from the UCE family, with theorem statements clarifying exactly what assumption from the family is used by a particular application.

1.4 Applications

Fig. 1 summarizes the applications we now discuss.

1. Deterministic PKE. The EwH deterministic PKE (D-PKE) ROM scheme of BBO07 [12] 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 UCE-secure then this instantiated D-PKE scheme is PRIV-secure

Goal	Result
D-PKE	Instantiation of the ROM EwH scheme of [12] to obtain the first standard model deterministic PKE scheme providing full IND [15] and PRIV [12] security. Section 5.3.
MLE	Instantiation of the ROM convergent encryption scheme of [63, 19], showing this in-use message-locked encryption scheme meets the IND$-CDA goal of [19]. Section 5.4.
HC	A UCE-secure family is hardcore for any one-way function and allows for extraction of any number of hardcore bits. Section 5.1.
BR93 PKE	Instantiation of a natural ROM PKE scheme from BR93 [23] showing it is IND-CPA-secure. Section 5.2.
PFOB	Instantiation of a ROM point-function obfuscation scheme of [95] to obtain a secure standard-model scheme. Section 5.5.
KDM	Instantiation of the ROM BRS scheme [27] to get an efficient and natural standard-model symmetric scheme for encryption of key-dependent messages. Section 5.6.
RKA	An efficient standard-model symmetric encryption scheme providing best-possible security against related-key attacks. Section 5.7.
CIH	Construction from UCE of correlation-intractable hash functions meeting the strongest notion of [79]. Section 5.10.
STORE	Instantiation of a natural ROM proof of storage scheme from [103]. Section 5.9.
OAEP	IND-CPA-KI security of OAEP [24] assuming partial one-wayness (with unpredictability) or one-wayness (with reset-security) of the underlying trapdoor function. Section 5.8.
GB	Standard-model adaptively secure garbling with short tokens. Section 5.11.

Figure 1: **Applications of UCE:** We summarize results for different goals.

in the standard model. This is the first standard-model PRIV-secure scheme (previous standard-model D-PKE schemes achieve only restricted notions of blocksource-PRIV-security [30, 15, 43, 67]). Our proof makes crucial use of the equivalence between PRIV and an indistinguishability-style notion IND of D-PKE security [15].

2. Message-locked encryption. In convergent encryption (CE) [63, 19], 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 [19] to meet, in the ROM, a formal notion of MLE-security called PRV$-CDA. We instantiate with a UCE-family, putting the key in public parameters, and show that the resulting MLE scheme is PRV$-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. UCE 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 [23] 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 UCE-secure family preserves the IND-CPA security.

5. Point-function obfuscation. A *point function* has non-$\perp$ output on just one point. Lynn, Prabhakaran, and Sahai [95] give a ROM point-function obfuscation scheme. We UCE-instantiate their construction to obtain a standard-model point-function obfuscation scheme.

6. KDM-secure SE. Black, Rogaway and Shrimpton (BRS) [27] 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 UCE-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 [41, 5, 10, 96, 4].

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 [6, 22] 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 UCE-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 [79] and proposed several notions of security for them. GOR provided constructions achieving limited CIH security from the q-DHI assumption of [35] and from RKA-secure blockciphers, but achieving full CIH security in the standard model has remained open. We solve this problem, showing that UCE-secure function families are selective (pseudorandomness) CIH secure in the terminology of GOR.
9. Secure storage. Ristenpart, Shacham and Shrimpton [103] 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 [97, 56] may fail to securely replace the RO. In contrast, we show that UCE instantiation succeeds. (Our instantiation lets the challenge in the protocol be a key naming a member of a UCE-secure family of functions.)
10. OAEP. OAEP [24] has been a benchmark for RO instantiation [32, 31, 89]. 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 unpredictability-based UCE, we get IND-CPA-KI security under the partial-domain one-wayness, and hence by [66] under standard one-wayness, of RSA; under reset-security based UCE 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.[2] Kiltz, O'Neill and Smith (KOS) [89] 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 [46]. In comparison our results for RSA are under the standard one-wayness assumption.
11. Adaptively-secure garbling. Verifiable outsourcing [70], as well as one-time programs [76], call for garbling schemes that are adaptively secure [16]. 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 [16, 78]. 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 UCE-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 [70]. Our garbling scheme is obtained by instantiating a ROM garbled circuit construction of [101].

1.5 Constructing UCE-secure families

We provide a ROM construction of a family of functions shown to achieve all the forms of UCE we use in this paper.

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.

[2] More precisely, they are not allowed to depend on hk but are allowed to depend on the RSA part of 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.

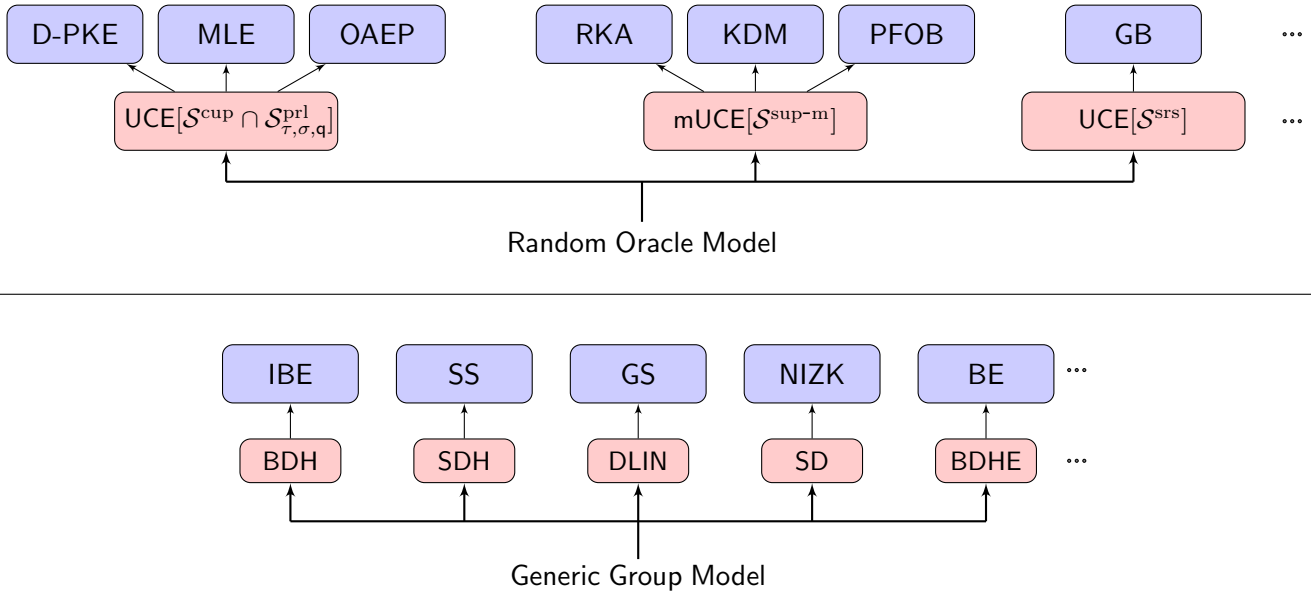

Figure 2: **The layered-cryptography paradigm for the ROM (top) and for pairing-based cryptography (bottom).** Assumptions are validated in the idealized model and then used to attain end goals entirely in the standard model. Definitions of the UCE classes depicted above are in Section 4. SS refers to the short signatures of [34]; BE refers to the broadcast encryption scheme of [39]; NIZK refers to the NIZK arguments of [80]. See text for other abbreviations.

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 that HMAC [14, 11] is a suitable instantiation for some of our forms of UCE.

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. 2. 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 instantiate families assumed UCE-secure via appropriately-keyed cryptographic hash functions whenever these appear to meet the particular UCE notion being used.

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 [38], DLIN [37], SDH [37], BDHE [36] and SD (Subgroup Decision) [40] 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 model. But the generic-group model is subject to issues, critiques and counter-examples analogous to those for the ROM [64, 59]. 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. 2). 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.

Unification. The ability to UCE-instantiate the RO across different schemes and goals shows that these ROM schemes have something in common, meaning they are in some way relying on the same attributes of the RO for security. This was not obvious to us prior to conceiving UCE. UCE thus leads to a better understanding of what properties of the RO schemes rely on, and enables us to unify different usages under a common umbrella.

Assumption degree and achieving UCE. In the UCE definition, the adversary consists of stages (source and distinguisher) that (due to the imposed restrictions on the source such as unpredictability or reset-security) 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 [83, 100] 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 [105] 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.

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 [49, 97]. (For example, UCE does not imply correlation intractability as defined, and shown to be unachievable in the standard model, by [49].)

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 [25]. This is consistent with impossibility results [60]. However, it is possible to instantiate the RO in FDH signatures directly [85].

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 [12] is simple enough, but a rigorous ROM proof is in our view less straightforward than our proof of Theorem 5.3 for the UCE-based instantiation of EwH.

The term "computational extractor" has been used for primitives that extract pseudorandomness from distributions that have computational min-entropy [57, 92, 65]. A UCE-secure family instead extracts pseudorandomness from distributions constrained in other ways, for example being unpredictable. These may or may not have computational min-entropy in the sense of [81, 86] but we viewed UCE as similar in spirit and so preserved the "extractor" name. "Universal" refers to the ability to get randomness from

starting distribution subject to a variety of conditions.

Programmable hash functions [84] are an information-theoretic tool that in some way mimic the "programmability" of ROs and were used by [84] 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 [71, 33] define new security properties of hash functions tailored for their own particular applications.

FUTURE DIRECTIONS AND OPEN QUESTIONS. Achieving UCE under other assumptions is an interesting and important direction for future work. We suggest to begin by targeting restricted versions of UCE, starting with independent sources (ones whose oracle queries consist of uniform, independent strings) and moving on to block sources (each oracle query retains high min-entropy even given previous ones) [54]. In these cases, we may hope to achieve security under first-degree assumptions. An indication (but not a proof) that this may be possible is that D-PKE that is PRIV-secure for these kinds of sources has been achieved under first-degree assumptions [15, 30, 43, 67]. Full UCE security would, of course, require second-degree assumptions.

Another interesting direction is to find further applications of UCE, in particular to instantiate ROs or build schemes without random oracles.

UCE is a framework permitting definitional variants beyond the ones 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. (By "broken" we mean that we have found attacks showing that *no* family can meet the definition.) Beyond this the larger agenda is to further layered cryptography for the ROM by finding other standard-model definitions for hash families that permit these families to instantiate ROs in applications. A target of particular interest is instantiation of the RO in FDH signatures [25, 55, 60, 88].

Determining the relationship between UCE and mUCE is an interesting open question. That is, given a family of functions H that is UCE-secure relative to a particular form of UCE, is it mUCE-secure relative to the multi-key extension of the same form? We conjecture that the answer is "yes" for a version of mUCE in which the number of keys is a constant independent of the adversary, but in general the answer is "no." To demonstrate the latter would require a counter-example, meaning a family H that is UCE-secure but not mUCE-secure.

We have shown in Section 4.5 how to build a VOL (variable output length) UCE family from a FOL (fixed output length) UCE family. An interesting direction is the analog for input lengths, namely the problem usually called domain extension: build a UCE-secure family taking arbitrary-length inputs from one taking fixed-length inputs. Domain extension has been a popular topic for many primitives in the past.

We have suggested in Section 6.2 that HMAC [14, 93] is a candidate for a practical instantiation of a UCE-secure family. An interesting problem is to either refute this via an attack or validate it in an idealized model, namely prove that HMAC meets our ROM-based UCE definitions of Section 6.1 assuming the compression function underlying the hash function is ideal. Since we have shown in Section 6.1 that a RO is effectively UCE-secure, one might hope to obtain the desired result for HMAC based on the indifferentiability of the latter from a RO [62], but, as per [103, 58], indifferentiability [97, 56] may not suffice since UCE is a second-degree primitive whose security definition is underlain by a multi-stage game. Thus some other approach or a direct analysis may be needed.

Most importantly, UCE needs further cryptanalysis to test the plausibility of the various assumptions being made. The BFM attack [45] indicates that one should be cautious with regard to UCE-related assumptions and in particular we need to better understand the ability of indistinguishability obfuscation [9, 68] to attack UCE.

3 Preliminaries

NOTATION. By $\lambda \in \mathbb{N}$ we denote the security parameter. If $n \in \mathbb{N}$ then 1^n denotes its unary representation. We denote the size of a finite set X by $|X|$, the number of coordinates of a vector $\mathbf{x}$ by $|\mathbf{x}|$, and the length

of a string $x \in \{0,1\}^*$ by $|x|$. We let ε denote the empty string. If x is a string then $x[i]$ is its i-th bit and $x[1,\ell] = x[1]\ldots x[\ell]$. By $x\|y$ we denote the concatenation of strings x, y. If x is a string and $0 \le \ell \le |x|$, then $y\|_\ell z \leftarrow x$ denotes letting y and z be strings such that $|y| = \ell$ and $y\|z = x$. If X is a finite set, we let $x \leftarrow\!\!\$\ X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be 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 $x_1,\ldots$.

For $a, b \in \mathbb{N}$ and $a \le b$, let $[a,b]$ denote the set $\{a, a+1,\ldots,b\}$. We say that $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every polynomial p, there exists $n_p \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for all $n > n_p$. An adversary is an algorithm or a tuple of algorithms.

GAMES. We use the code based game playing framework of [26] augmented with explicit MAIN procedures as in [103]. (See Fig. 3 for an example.) By $\mathrm{G}^A(\lambda) \Rightarrow y$ we denote the event that the execution of game G with adversary A and security parameter λ results in output y, the game output being what is returned by MAIN. We abbreviate $\mathrm{G}^A(\lambda) \Rightarrow \mathsf{true}$ by $\mathrm{G}^A(\lambda)$, the occurrence of this event meaning that A wins the game. The running time of an adversary A in a game G is a function that associates to $\lambda \in \mathbb{N}$ the worst-case number of steps executed in $\mathrm{G}^A(\lambda)$. Oracle queries are charged the cost of writing the query and reading the response, meaning a query x getting response y is charged time $O(|x| + |y|)$ where the big-oh hides an absolute constant. Thus, the running time of game procedures is not included in the running time of the adversary.

4 UCE

We introduce the general UCE definitional framework and then discuss specific UCE classes (assumptions). We provide some simplifications and relations with other notions. We then discuss multi-UCE.

4.1 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 $\mathsf{H.IL}(\lambda) \subseteq \mathbb{N}$ 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 $\mathsf{H.ol}$.

4.2 UCE security

FRAMEWORK. Let H be a family of functions. Let S be an adversary called the *source* and D an adversary called the *distinguisher*. We associate to them and H the game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\lambda)$ of Fig. 3. 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*.

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

Figure 3: **Games** UCE**,** Pred **used to define UCE 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.

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 uce-advantage of (S, D) is defined for $\lambda \in \mathbb{N}$ via

$$\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\lambda) = 2\Pr[\text{UCE}^{S,D}_{\mathsf{H}}(\lambda)] - 1\ . \tag{1}$$

One's first thought may now be to say that H is UCE-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.

To obtain useful and potentially achievable definitions of UCE-security for H, we will restrict the adversaries. There are many ways to do this, so that UCE will be not a single definition but rather a framework in which many definitions are possible.

Let $\mathcal{S}$ be a class of sources and $\mathcal{D}$ a class of distinguishers. Then we let $\mathsf{UCE}[\mathcal{S}, \mathcal{D}]$ be the set of all H such that $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible for all $(S, D) \in \mathcal{S} \times \mathcal{D}$. To say that H is $\mathsf{UCE}[\mathcal{S}, \mathcal{D}]$-secure simply means that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}, \mathcal{D}]$. We let $\mathcal{S}^{\mathsf{poly}}$ be the class of all PT sources and $\mathcal{D}^{\mathsf{poly}}$ the class of all PT distinguishers. We will almost always restrict attention to the latter and it is thus convenient to let $\mathsf{UCE}[\mathcal{S}] = \mathsf{UCE}[\mathcal{S}, \mathcal{D}^{\mathsf{poly}}]$. The following simple fact will often be used:

Proposition 4.1 Suppose $\mathcal{S}_1 \subseteq \mathcal{S}_2$. Then $\mathsf{UCE}[\mathcal{S}_2] \subseteq \mathsf{UCE}[\mathcal{S}_1]$.

Put another way, the smaller the set of allowed sources, the weaker the assumption of UCE-security. Now we consider different choices of $\mathcal{S}$ for which we will make $\mathsf{UCE}[\mathcal{S}]$-security assumptions. We restrict sources in two basic ways, namely by requiring either *unpredictability* or *reset-security*, as we discuss next.

UNPREDICTABLE SOURCES. Let S be a source. Consider game $\text{Pred}^{P}_{S}(\lambda)$ of Fig. 3 associated to 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

$$\mathsf{Adv}^{\mathsf{pred}}_{S,P}(\lambda) = \Pr[\text{Pred}^{P}_{S}(\lambda)]\ .$$

We say that P is a computational predictor if it is PT, and we say it is a statistical predictor if there exist polynomials q, q' such that for all $\lambda \in \mathbb{N}$, predictor P makes at most $q(\lambda)$ oracle queries and outputs a set Q' of size at most $q'(\lambda)$ in game $\text{Pred}^{P}_{S}(\lambda)$. We stress that in this case the predictor need not be PT. We say S is *computationally unpredictable* if $\mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot)$ is negligible for all computational predictors P. We say S is *statistically unpredictable* if $\mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot)$ is negligible for all statistical predictors P. We let $\mathcal{S}^{\text{cup}}$ be the class of all PT, computationally unpredictable sources and $\mathcal{S}^{\text{sup}} \subseteq \mathcal{S}^{\text{cup}}$ the class of all PT, statistically unpredictable sources. This gives our first UCE classes, namely $\mathsf{UCE}[\mathcal{S}^{\text{cup}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\text{sup}}]$.

MAIN $\mathrm{Reset}_S^R(\lambda)$	MAIN $\mathrm{mReset}_S^R(\lambda)$
$\mathsf{Dom} \leftarrow \emptyset$; $L \leftarrow\!\!\$\ S^{\mathrm{HASH}}(1^\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$	$\mathsf{Dom} \leftarrow \emptyset$; $(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$; $L \leftarrow\!\!\$\ S^{\mathrm{HASH}}(1^\lambda, t)$; $b \leftarrow\!\!\$\ \{0,1\}$
If $b = 0$ then // reset the array T	If $b = 0$ then // reset the array T
For all $(x, \ell) \in \mathsf{Dom}$ do $T[x, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	For all $(x, \ell, i) \in \mathsf{Dom}$ do $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
$b' \leftarrow R^{\mathrm{HASH}}(1^\lambda, L)$; Return $(b' = b)$	$b' \leftarrow\!\!\$\ R^{\mathrm{HASH}}(1^\lambda, 1^n, L)$; Return $(b = b')$
HASH$(x, 1^\ell)$	HASH$(x, 1^\ell, i)$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{(x, \ell)\}$	$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{(x, \ell, i)\}$
If $T[x, \ell] = \perp$ then $T[x, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell]$	Return $T[x, \ell, i]$

Figure 4: **Left:** Game defining reset security of a source. **Right:** Game defining reset security of a multi source.

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 the definition of unpredictability: the latter is a property of the source.

RESET-SECURE SOURCES. Let S be a source. Consider game $\mathrm{Reset}_S^R(\lambda)$ of Fig. 4 associated to S and an adversary R called a reset adversary. The latter wins if it can distinguish between the RO HASH used by the source S and an independent RO. For $\lambda \in \mathbb{N}$ we let

$$\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\lambda) = 2\Pr[\mathrm{Reset}_S^R(\lambda)] - 1 \ .$$

We say R is a computational reset adversary if it is PT, and we say it is a statistical reset adversary if there exists a polynomial q such that for all $\lambda \in \mathbb{N}$, reset adversary R makes at most $q(\lambda)$ oracle queries in game $\mathrm{Reset}_S^R(\lambda)$. We stress that in this case the reset adversary need not be PT. We say S is *computationally reset-secure* if $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot)$ is negligible for all computational reset adversaries R. We say S is *statistically reset-secure* if $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot)$ is negligible for all statistical reset adversaries R. We let $\mathcal{S}^{\mathrm{crs}}$ be the class of all PT, computationally reset-secure sources and $\mathcal{S}^{\mathrm{srs}} \subseteq \mathcal{S}^{\mathrm{crs}}$ the class of all PT, statistically reset-secure sources. This gives our next UCE classes, namely $\mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$.

For some intuition, recall that unpredictability was introduced as a way to rule out unpreventable success. The example we gave is the source S_1 that queries to its HASH oracle a point x to get back y and returns $L = (x, y)$ as the leakage. S_1 is "insecure" in the sense that there is PT distinguisher D such that $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}$ is high. But this does not violate $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$-security or $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$-security since S_1 is unpredictable. Now, let source S_2 query x and get back y but return only $L = x$ as the leakage. $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$ put this source out of consideration because it is predictable. Yet, this source is "secure" in the sense that a PT distinguisher, given L, has no advantage in determining the challenge bit. This indicates that unpredictability is sometimes too strong a requirement. However S_2 is (statistically) reset secure. Thus, $\mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}]$ and $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$ allow us to "use" S_2. This ability to use a larger class of sources strengthens the notion and is useful in some applications. The following shows that that reset-security based UCE is (strictly) stronger than unpredictability-based UCE. The proof is in Appendix A. Part (2) below assumes that $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}] \neq \emptyset$.

Proposition 4.2 (1) $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$ (2) $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}] \not\subseteq \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$.

DISCUSSION. At this point we have suggested the following, specific UCE classes:

- $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}], \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$ — UCE security for sources that are statistically or computationally unpredictable.
- $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}], \mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}]$ — UCE security for sources that are statistically or computationally reset secure.

$\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$-security was called UCE1 in the preliminary version of this work [18]. Subsequently, Brzuska, Farshim and Mittelbach (BFM) [45] showed that if indistinguishability obfuscation (iO) [9, 68] is possible then UCE1 security is not achievable, meaning $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}] = \emptyset$. Given *any* family of functions H, they

Source $S^{\text{HASH}}(1^\lambda)$	Source $S^{\text{HASH}}(1^\lambda)$
$(L_0, \mathbf{x}, 1^{\boldsymbol{\ell}}) \leftarrow\!\!\$\ S_0(1^\lambda)$	$(L_0, \mathbf{L}') \leftarrow\!\!\$\ S_0(1^\lambda)$
For $i = 1, \ldots, \lvert\mathbf{x}\rvert$ do $\mathbf{y}[i] \leftarrow\!\!\$\ \text{HASH}(\mathbf{x}[i], 1^{\boldsymbol{\ell}[i]})$	For $i = 1, \ldots, \lvert\mathbf{L}'\rvert$ do $\mathbf{L}[i] \leftarrow\!\!\$\ S_1^{\text{HASH}}(1^\lambda, \mathbf{L}'[i])$
$L_1 \leftarrow\!\!\$\ S_1(1^\lambda, \mathbf{y})$; $L \leftarrow (L_0, L_1)$	$L \leftarrow (L_0, \mathbf{L})$
Return L	Return L

Figure 5: **Left:** The split source $S = \mathsf{Splt}[S_0, S_1]$ associated to S_0, S_1. **Right:** The parallel source $S = \mathsf{Prl}[S_0, S_1]$ associated to S_0, S_1.

present an iO-based attack, in the form of a source $S \in \mathcal{S}^{\mathsf{poly}}$ and a distinguisher $D \in \mathcal{D}^{\mathsf{poly}}$ such that $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)$ is not negligible, and, assuming security of the iO, source S is computationally unpredictable. By Proposition 4.2, $\mathsf{UCE}[\mathcal{S}^{\text{crs}}]$-security, called UCE2 in [18], is also not achievable.

The BFM attack [45] however does not break any of the applications in [18], indicating that weaker assumptions may suffice for these applications. Indeed, an examination of the proofs of the applications proved in [18] under the $\mathsf{UCE}[\mathcal{S}^{\text{cup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\text{crs}}]$ assumptions (or their multi-key counterparts) shows that in all cases they in fact use weaker assumptions not broken by the BFM attack. The current version of our paper spells out these weaker assumptions and updates the theorem statements for the applications to make reliance on them explicit.

Amongst the new assumptions are the weaker, statistical versions of the prior assumptions, namely $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$ and $\mathsf{UCE}[\mathcal{S}^{\text{srs}}]$. How safe are these statistical assumptions? The source S in the BFM attack [45] is not known to be statistically unpredictable, so their attack will not apply as stated. For the BFM attack to work here, the iO would have to be statistically secure. There is some evidence that statistically-secure iO for general circuits is not possible. Namely, GGHRSW [68] give a construction of witness encryption from iO, and we observe that if the latter is statistically secure, so is the former. But GGSW [69] show that statistically-secure witness encryption implies a collapse of the PH (Polynomial-Time Hierarchy). Statistically-secure iO is possible in idealized models [53, 42], but this does not translate to a standard-model attack on $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$ or $\mathsf{UCE}[\mathcal{S}^{\text{srs}}]$, although it might translate to some kind of evidence against blackbox proofs from more standard assumptions. Of course this does not mean the assumptions are necessarily sound, for there may be other ways to invalidate them. We remark that BFM [45] independently suggested the use of the statistical assumptions.

The assumption under which iO is achieved in [68] is novel, and there is some chance that it is false, which would remove the attack on $\mathsf{UCE}[\mathcal{S}^{\text{cup}}], \mathsf{UCE}[\mathcal{S}^{\text{crs}}]$. However, at this point our sense is that the existence of iO is more likely than the existence of families that are $\mathsf{UCE}[\mathcal{S}^{\text{cup}}]$-secure or $\mathsf{UCE}[\mathcal{S}^{\text{crs}}]$-secure, and that the latter assumptions should not be used. More broadly, the BFM attack [45] indicates that one must be careful in making any UCE-related assumptions.

The ability of BFM [45] to attack $\mathsf{UCE}[\mathcal{S}^{\text{cup}}]$ validates our thesis that UCE-style assumptions are "falsifiable," not in any formal sense of the word in quotes, but in the sense that cryptanalysis can be used to invalidate them. This was indeed one of the goals of our approach.

As indicated above, many of our applications can be based on the statistical rather than computational versions of our original assumptions. Some of the others, however, used computational unpredictability or reset-security in a crucial way. We will resurrect these via assumptions that restrict the source in other ways, as we now discuss.

Split sources. Let S_0, S_1 be algorithms, neither of which have access to any oracles. The *split source* $S = \mathsf{Splt}[S_0, S_1]$ associated to S_0, S_1 is defined on the top left of Fig. 5. Here S_0 returns two vectors of the same length, the second consisting of unary-encoded integers, meaning $1^{\boldsymbol{\ell}}$ denotes the vector whose i-th entry $1^{\boldsymbol{\ell}[i]}$ is the unary-encoding of an integer $\boldsymbol{\ell}[i]$. The first adversary creates the oracle queries for the source S, the latter making these queries and passing the replies to the second adversary to get the leakage. In this way, neither S_0 nor S_1 have an input-output pair from the oracle, limiting their ability to create leakage useful to the distinguisher. A source S is said to belong to the class $\mathcal{S}^{\text{splt}}$ if there exist PT S_0, S_1 such that $S = \mathsf{Splt}[S_0, S_1]$, meaning is defined as above. The associated UCE classes are $\mathsf{UCE}[\mathcal{S}^{\text{cup}} \cap \mathcal{S}^{\text{splt}}]$ and $\mathsf{UCE}[\mathcal{S}^{\text{crs}} \cap \mathcal{S}^{\text{splt}}]$, meaning UCE-security for computationally unpredictable, split sources or computationally

reset-secure, split sources.

BOUNDED SOURCES. Let S_0, S_1 be algorithms, where S_1 has access to an oracle but S_0 does not. The *parallel source* $S = \mathsf{Prl}[S_0, S_1]$ associated to S_0, S_1 is defined on the top right of Fig. 5. Here $\mathbf{L}', \mathbf{L}$ are vectors. We clarify that the $|\mathbf{L}'|$ invocations of S_1 are independent of each other, sharing no coins or state. A source S is said to belong to the class $\mathcal{S}^{\rm prl}_{\tau,\sigma,q}$, where τ, σ, q are polynomials, if there exist PT S_0, S_1 such that (1) $S = \mathsf{Prl}[S_0, S_1]$ (2) the running time (circuit size) of S_1 is $\tau(\cdot)$ and it makes at most $q(\cdot)$ oracle queries, and (3) the length of the leakage string L_0 produced by S_0 is at most $\sigma(\cdot)$. The associated UCE classes are $\mathsf{UCE}[\mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}] \subseteq \mathsf{UCE}[\mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}]$, meaning UCE-security for computationally unpredictable, parallel sources or computationally reset-secure, parallel sources that obey the complexity constraints given by the parameters τ, σ, q.

In applications, τ will be related to a base scheme, for example the time to compute one encryption under some base encryption scheme, so that it does not involve the hash family H. Thus τ will be a fixed polynomial. The same will be true of σ. The parameter q will typically be a small constant. This makes it possible to pick H so that a circuit describing it, and thus an obfuscation of this circuit, has size larger than $\tau(\cdot)$ and $\sigma(\cdot)$, preventing S_0 from communicating such a circuit, and preventing S_1 from creating (and thus communicating) such a circuit, in the leakage. We note that while S_1 is restricted to time $\tau(\cdot)$, the source S itself has running time $\mathcal{O}(\tau_0 + p\tau)$ where $\tau_0(\cdot)$ is the running time of S_0 and $p(\cdot)$ is the length of $\mathbf{L}'$, and thus the running time of S can be an arbitrary polynomial.

Looking back, the difficulty of instantiating a RO via a family H has always been that H has a succinct description. This is what the CGH attacks [49] exploit, and the BFM attack [45] does as well. The idea above is to counter these attacks by limiting the time or space complexity of the adversary so that it may not be able to create or communicate a description of the hash function. The complication is that if the total time of our source is limited, it cannot run another adversary as required in a reduction. The parallel model allows us to limit the complexity of relevant parts of the computation of the source while still letting its overall running time be an arbitrary polynomial.

INSTANTIATIONS. In an implementation, the instantiation of an assumed UCE-secure function would depend on what type of UCE-security is assumed. For $\mathsf{UCE}[\mathcal{S}^{\rm sup}], \mathsf{UCE}[\mathcal{S}^{\rm srs}], \mathsf{UCE}[\mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm splt}], \mathsf{UCE}[\mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm splt}]$, we would suggest an instantiation based on HMAC [14, 11]. For $\mathsf{UCE}[\mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}], \mathsf{UCE}[\mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}]$, this may not suffice because $\tau(\cdot)$ may be much larger than the time to compute HMAC. A slower instantiation may be safer. This means that applications relying on these assumptions may see a loss in efficiency.

RELATION TO EXTRACTORS. The UCE framework captures many well-known primitives as special cases. One of these is (strong, randomness) extractors. We will use an asymptotic version of the definition of [99]. Namely, a family H with input length $\mathsf{H.il}$ and output length $\mathsf{H.ol}$ is an extractor if, whenever $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$ is drawn from a distribution $X(\lambda)$ such that $2^{\mathsf{H.ol}(\cdot) - \mathrm{H}_\infty(X(\cdot))}$ is negligible, then the distributions $(hk, \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})$ and $(hk, U(\lambda))$ are statistically close, where $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ and $U(\lambda)$ is the uniform distribution on $\{0,1\}^{\mathsf{H.ol}(\lambda)}$. We will define classes $\mathcal{S}^{\rm ext}, \mathcal{D}^{\rm ext}$ such that a family H is an extractor if and only if $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm ext}, \mathcal{D}^{\rm ext}]$. We let $\mathcal{D}^{\rm ext}$ be the class of all distinguishers, meaning a distinguisher is computationally unbounded. A source S is in the class $\mathcal{S}^{\rm ext}$ if (1) It makes exactly one oracle query, and returns the response as the leakage (2) For every simple statistical predictor P' (cf. Section 4.3) the function $2^{\mathsf{H.ol}(\cdot)} \cdot \mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot)$ is negligible. (The oracle query plays the role of the input x and the second condition captures the min-entropy requirement.) Then $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm ext}, \mathcal{D}^{\rm ext}]$ if and only if H is an extractor as per the asymptotic definition we sketched above. By allowing the leakage to also include a function of the oracle query x, we can capture average-case extractors as defined in [61]. Similarly one can define classes $\mathcal{S}, \mathcal{D}$ such that $\mathsf{UCE}[\mathcal{S}, \mathcal{D}]$ captures computational extractors as per [57, 92, 65]. We can also capture hardcore predicates. These relations influenced our choice of the term "universal computational extractor" for our framework and show that the UCE framework can function as an umbrella abstraction for many notions in the literature.

4.3 Simple unpredictability

Applications of unpredictability-based UCE assumptions 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 $\mathrm{SPred}_S^{P'}(\lambda)$ of Fig. 3 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

$$\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\lambda) = \Pr[\mathrm{SPred}_S^{P'}(\lambda)] \ .$$

We say that P' is computational if it is PT, and we say it is statistical if it may or may not be PT. We say that source S is *simple computationally unpredictable* if $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot)$ is negligible for all computational simple predictors P'. We say that source S is *simple statistically unpredictable* if $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot)$ is negligible for all statistical simple predictors P'. The following says that simple unpredictability is equivalent to unpredictability in both cases.

Lemma 4.3 Let S be a source. Then S is computationally unpredictable if and only if it is simple computationally unpredictable, and S is statistically unpredictable if and only if it is simple statistically unpredictable.

Proof of Lemma 4.3 : Suppose P' is a simple predictor. Let $P^{\mathrm{HASH}}(1^\lambda, L)$ run $x \leftarrow\!\!\$\ P'(1^\lambda, L)$ and return $\{x\}$. Then $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le \mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot)$. This shows that if S is unpredictable then it is also simple unpredictable, whether this be computational or statistical. Turning to the converse, let P be a predictor. We may assume wlog that S and P never repeat HASH queries. We may also assume wlog that the output Q' of P contains every x for which there exists ℓ such that HASH-query $(x, 1^\ell)$ was made by P. (This is wlog because we can modify P to include all such x in Q'.) Let q be a polynomial that bounds the number of elements in the output Q' of P. Game $\mathrm{G}_1^{S,P}(\lambda)$ below includes the boxed code while game $\mathrm{G}_2^{S,P}(\lambda)$ does not:

$\underline{P'(1^\lambda, L)}$	$\underline{\textsc{Main}\ \boxed{\mathrm{G}_1^{S,P}(\lambda)},\ \mathrm{G}_2^{S,P}(\lambda)}$
$Q' \leftarrow\!\!\$\ P^{\mathrm{HASHSIM}}(1^\lambda, L)$; $x \leftarrow\!\!\$\ Q'$; Return x	$Q \leftarrow \emptyset$; $L \leftarrow\!\!\$\ S^{\mathrm{HASH}_1}(1^\lambda)$; $Q' \leftarrow\!\!\$\ P^{\mathrm{HASH}_2}(1^\lambda, L)$
$\underline{\textsc{HashSim}(x, 1^\ell)}$	Return $(Q \cap Q' \neq \emptyset)$
$y \leftarrow\!\!\$\ \{0,1\}^\ell$; Return y	$\underline{\textsc{Hash}_1(x, 1^\ell)}$
	$Q \leftarrow Q \cup \{x\}$; $T[x,\ell] \leftarrow\!\!\$\ \{0,1\}^\ell$;
	Return $T[x,\ell]$
	$\underline{\textsc{Hash}_2(x, 1^\ell)}$
	If $x \in Q$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\text{Return } T[x,\ell]}$
	$y \leftarrow\!\!\$\ \{0,1\}^\ell$; Return y

Game $\mathrm{G}_1^{S,P}(\lambda)$ is identical to $\mathrm{Pred}_S^P(\lambda)$, except that it separates the HASH procedures used by S and P, while maintaining consistency. Setting bad has no effect on the outcome of the game. Games $\mathrm{G}_1^{S,P}(\lambda)$ and $\mathrm{G}_2^{S,P}(\lambda)$ are identical-until-bad. From the fundamental lemma of game-playing [26],

$$\mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot) = \Pr[\mathrm{G}_1^{S,P}(\cdot)] \le \Pr[\mathrm{G}_2^{S,P}(\cdot)] + \Pr[\mathrm{G}_2^{S,P}(\cdot) \text{ sets } \mathsf{bad}].$$

From the assumption that Q' contains every x such that P queried some $(x, 1^\ell)$ to HASH, if game G_2 sets bad then P will surely win. Hence $\Pr[\mathrm{G}_2^{S,P}(\cdot) \text{ sets } \mathsf{bad}] \le \Pr[\mathrm{G}_2^{S,P}(\cdot)]$. Now, consider the simple predictor P' as above. Then

$$\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) = \frac{1}{q}\Pr[\mathrm{G}_2^{S,P}(\cdot)] \ge \frac{1}{2q}\mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot) \ .$$

Note that if P is computational so is P' and if P is statistical then so is P'. This concludes the proof. ▌

$\underline{\text{MAIN } \text{PRF}^A_{\mathsf{H}}(\lambda)}$	$\underline{\text{MAIN } \text{CR}^A_{\mathsf{H}}(\lambda)}$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}\ ;\ b' \leftarrow\!\!\$\ A^{\text{HASH}}(1^\lambda)$	$(x_0, x_1) \leftarrow\!\!\$\ A(1^\lambda, hk)$
Return $(b' = b)$	If $(x_0 = x_1)$ then return false
$\underline{\text{HASH}(x, 1^\ell)}$	Return $(\mathsf{H.Ev}(1^\lambda, hk, x_0, 1^{\mathsf{H.ol}(\lambda)}) = \mathsf{H.Ev}(1^\lambda, hk, x_1, 1^{\mathsf{H.ol}(\lambda)}))$
If $T[x,\ell] = \perp$ then	
If $b = 1$ then	
$T[x,\ell] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^\ell)$	
Else $T[x,\ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	
Return $T[x,\ell]$	

Figure 6: **Games defining PRF and CR security of family of functions H.**

4.4 Relations

We look at how UCE relates to standard notions of security for families of hash functions, namely PRF and CR. We focus on $\mathsf{UCE}[\mathcal{S}^{\text{sup}}], \mathsf{UCE}[\mathcal{S}^{\text{srs}}]$, leaving relations involving other classes we have introduced as open questions.

DEFINITIONS. We begin by recalling the definitions. Let H be a hash family with output length $\mathsf{H.ol}$. We say that H is PRF-secure if $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{H},A}(\cdot)$ is negligible for all PT A, where $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{H},A}(\lambda) = 2\Pr[\text{PRF}^A_{\mathsf{H}}(\lambda)] - 1$ and game $\text{PRF}^A_{\mathsf{H}}(\lambda)$ is shown in Fig. 6. Here we require that a query (x, ℓ) to HASH satisfy $|x| \in \mathsf{H.IL}(\lambda)$ and $\ell = \mathsf{H.ol}(\lambda)$. We say that H is collision-resistant (CR) if $\mathsf{Adv}^{\mathsf{cr}}_{\mathsf{H},A}(\cdot)$ is negligible for all PT A, where $\mathsf{Adv}^{\mathsf{cr}}_{\mathsf{H},A}(\lambda) = \Pr[\text{CR}^A_{\mathsf{H}}(\lambda)]$ and game $\text{CR}^A_{\mathsf{H}}(\lambda)$ is shown in Fig. 6. Here we require that $|x_0|, |x_1| \in \mathsf{H.IL}(\lambda)$. Let PRF be the set of all families H that are PRFs and CR the set of all H that are collision-resistant.

RESULTS. The following says that $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$-security neither implies, nor is implied by, PRF-security, and similarly for collision resistance. In any non-containment $\mathsf{B} \not\subseteq \mathsf{A}$, we assume $\mathsf{B} \neq \emptyset$.

Proposition 4.4 (1) $\mathsf{UCE}[\mathcal{S}^{\text{sup}}] \not\subseteq \mathsf{PRF}$ (2) $\mathsf{PRF} \not\subseteq \mathsf{UCE}[\mathcal{S}^{\text{sup}}]$ (3) $\mathsf{UCE}[\mathcal{S}^{\text{sup}}] \not\subseteq \mathsf{CR}$ (4) $\mathsf{CR} \not\subseteq \mathsf{UCE}[\mathcal{S}^{\text{sup}}]$.

Proof of Proposition 4.4: For simplicity, we only consider hash families of fixed input and output length. Intuitively, the reason (1) is true is that a $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$-secure family could map a particular input, say $0^{\mathsf{H.il}(\lambda)}$, to $0^{\mathsf{H.ol}(\lambda)}$, under all keys. This clearly violates PRF-security but would not contradict $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$-security because the "bad" input is predictable. The counter-example for (3) is a family H where $\mathsf{H}(1^\lambda, hk, \cdot, 1^{\mathsf{H.ol}(\lambda)})$ maps $0^{\mathsf{H.il}(\lambda)}$ and $1^{\mathsf{H.il}(\lambda)}$ to the same output. Collision-resistance obviously fails, but since the "bad" inputs are predictable, $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$-security can be retained. Formally, for parts (1) and (3), let H be a $\mathsf{UCE}[\mathcal{S}^{\text{sup}}]$-secure hash family. Define $\overline{\mathsf{H}}$ as follows. Let $\overline{\mathsf{H}}.\mathsf{il} = \mathsf{H.il}$; let $\overline{\mathsf{H}}.\mathsf{ol} = \mathsf{H.ol}$; let $\overline{\mathsf{H}}.\mathsf{Kg} = \mathsf{H.Kg}$; and let $\overline{\mathsf{H}}.\mathsf{Ev}$ be as shown on the left below:

$\underline{\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})}$	$\underline{S^{\text{HASH}}(1^\lambda)}$	$\underline{\text{MAIN } \text{G}_1^{\overline{S},D}(\lambda)\ ,\ \boxed{\text{G}_2^{\overline{S},D}(\lambda)}}$
$y \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})$	$L \leftarrow\!\!\$\ \overline{S}^{\text{HASHSIM}}(1^\lambda)\ ;$ Return L	$L \leftarrow\!\!\$\ \overline{S}^{\text{HASHSIM}}(1^\lambda)\ ;\ b' \leftarrow\!\!\$\ D(1^\lambda, L)$
If $x \in \{0^{\mathsf{H.il}(\lambda)}, 1^{\mathsf{H.il}(\lambda)}\}$ then	$\underline{\text{HASHSIM}(x, 1^{\mathsf{H.ol}(\lambda)})}$	Return $(b' = 0)$
$y \leftarrow 0^{\mathsf{H.ol}(\lambda)}$	If $x \in \{0^{\mathsf{H.il}(\lambda)}, 1^{\mathsf{H.il}(\lambda)}\}$ then	$\underline{\text{HASHSIM}(x, 1^{\mathsf{H.ol}(\lambda)})}$
Return y	Return $0^{\mathsf{H.ol}(\lambda)}$	If $x \in \{0^{\mathsf{H.il}(\lambda)}, 1^{\mathsf{H.il}(\lambda)}\}$ then
	Else return $\text{HASH}(x, 1^{\mathsf{H.ol}(\lambda)})$	$\mathsf{bad} \leftarrow \mathsf{true}\ ;\ \boxed{T[x] \leftarrow 0^{\mathsf{H.ol}(\lambda)}}$
		If $T[x] \neq \perp$ then $T[x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
		Return $T[x]$

We claim that $\overline{\mathsf{H}} \notin \mathsf{PRF}$ and $\overline{\mathsf{H}} \notin \mathsf{CR}$ but $\overline{\mathsf{H}} \in \mathsf{UCE}[\mathcal{S}^{\text{sup}}]$, which establishes (1) and (3). The reason $\overline{\mathsf{H}} \notin \mathsf{PRF}$ is that an adversary can obtain an advantage of $1/2$ by querying $0^{\mathsf{H.il}(\lambda)}$ to FN and returning 1 if and only

if the first bit of the result is 0. The reason $\overline{\mathsf{H}} \notin \mathsf{CR}$ is that an adversary can output $(0^{\mathsf{H.il}(\lambda)}, 1^{\mathsf{H.il}(\lambda)})$ to win with advantage 1. To see that $\overline{\mathsf{H}} \in \mathsf{UCE}[\mathcal{S}^{\rm sup}]$, let $\overline{S}$ be a statistically unpredictable PT source, and D be a PT distinguisher. Consider the source S constructed above. Since $\overline{S}$ is statistically unpredictable, so is S. Let $P^{\rm HASH}(1^\lambda, L)$ be a predictor that always outputs $\{0^{\mathsf{H.il}(\lambda)}, 1^{\mathsf{H.il}(\lambda)}\}$ regardless of the leakage L. Consider games $\mathrm{G}_1^{\overline{S},D}(\lambda)$ and $\mathrm{G}_2^{\overline{S},D}(\lambda)$ above. Let b and c be the challenge bits of game $\mathrm{UCE}_{\overline{\mathsf{H}}}^{\overline{S},D}(\lambda)$ and game $\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda)$ respectively. The two games are identical-until-bad so by the Fundamental Lemma of Game Playing [26] we have:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{uce}}_{\overline{\mathsf{H}},\overline{S},D}(\cdot) &= \Pr[\mathrm{UCE}_{\overline{\mathsf{H}}}^{\overline{S},D}(\cdot) \mid b=1] + \Pr[\mathrm{UCE}_{\overline{\mathsf{H}}}^{\overline{S},D}(\cdot) \mid b=0] - 1 \\
&= \Pr[\mathrm{UCE}_{\overline{\mathsf{H}}}^{\overline{S},D}(\cdot) \mid b=1] + \Pr[\mathrm{G}_1^{\overline{S},D}(\cdot)] - 1 \\
&\leq \Pr[\mathrm{UCE}_{\overline{\mathsf{H}}}^{\overline{S},D}(\cdot) \mid b=1] + \Pr[\mathrm{G}_2^{\overline{S},D}(\cdot)] - 1 + \Pr[\mathrm{G}_1^{\overline{S},D}(\cdot) \text{ sets } \mathsf{bad}] \\
&= \Pr[\mathrm{UCE}_{\mathsf{H}}^{S,D}(\cdot) \mid c=1] + \Pr[\mathrm{UCE}_{\mathsf{H}}^{S,D}(\cdot) \mid c=0] - 1 + \mathsf{Adv}^{\mathsf{pred}}_{\overline{S},P}(\cdot) \\
&= \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + \mathsf{Adv}^{\mathsf{pred}}_{\overline{S},P}(\cdot) \ .
\end{aligned}$$

The claim then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm sup}]$ and that $\overline{S}$ is statistically unpredictable.

For part (2), the counter-example is a PRP, which is also a PRF but will be efficiently invertible given the key. Formally, the assumption $\mathsf{PRF} \neq \emptyset$ implies that there is an $\mathsf{H} \in \mathsf{PRF}$ that is a PRP. (This follows from [94].) This means $\mathsf{H.il} = \mathsf{H.ol}$ and there is a PT deterministic algorithm $\mathsf{H.Inv}$ which is the inverse of $\mathsf{H.Ev}$, meaning $\mathsf{H.Inv}(1^\lambda, hk, \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)}), 1^{\mathsf{H.ol}(\lambda)}) = x$ for all $\lambda \in \mathbb{N}$, all $hk \in [\mathsf{H.Kg}(1^\lambda)]$ and all $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$. To show that $\mathsf{H} \notin \mathsf{UCE}[\mathcal{S}^{\rm sup}]$, consider source S and distinguisher D below, where $x[1]$ denotes the first bit of x:

$\underline{S^{\rm HASH}(1^\lambda)}$	$\underline{D(1^\lambda, hk, L)}$
$x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.il}(\lambda)}$; $y \leftarrow \mathrm{HASH}(x, 1^{\mathsf{H.ol}(\lambda)})$	$(d, y) \leftarrow L$; $x \leftarrow \mathsf{H.Inv}(1^\lambda, hk, y, 1^{\mathsf{H.ol}(\lambda)})$
$L \leftarrow (x[1], y)$; Return L	If $x[1] = d$ then return 1 else return 0

The source S is statistically unpredictable, but $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) = 1/2$.

For part (4), the counter-example is a collision-resistant family all of whose function-outputs have some common structure, for example beginning with a 1-bit, which is enough to violate $\mathsf{UCE}[\mathcal{S}^{\rm sup}]$-security. Formally, let $\mathsf{H} \in \mathsf{CR}$ and define $\overline{\mathsf{H}}$ as follows: let $\overline{\mathsf{H}}.\mathsf{il} = \mathsf{H.il}$; let $\overline{\mathsf{H}}.\mathsf{ol} = \mathsf{H.ol} + 1$; let $\overline{\mathsf{H}}.\mathsf{Kg} = \mathsf{H.Kg}$; and let $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, x, 1^{\overline{\mathsf{H}}.\mathsf{ol}(\lambda)}) = 1 \,\|\, \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})$ for all $\lambda \in \mathbb{N}$, all $hk \in [\mathsf{H.Kg}(\lambda)]$ and all $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$. Then $\overline{\mathsf{H}} \in \mathsf{CR}$. On the other hand, we claim that $\overline{\mathsf{H}} \notin \mathsf{UCE}[\mathcal{S}^{\rm sup}]$. Let source S pick $x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.il}(\lambda)}$ and return the first bit of $\mathrm{HASH}(x, 1^{\overline{\mathsf{H}}.\mathsf{ol}(\lambda)})$ as the leakage. Then S is statistically unpredictable. Let D be a distinguisher that returns the leakage L. Then $\mathsf{Adv}^{\mathsf{uce}}_{\overline{\mathsf{H}},S,D}(\lambda) = 1/2$. ▌

We remark that a trivial counter-example for (3) is a family $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm sup}]$ with $\lambda \in \mathsf{H.IL}(\lambda)$ and $\mathsf{H.ol}(\lambda) = 1$ for all $\lambda \in \mathbb{N}$, such a family trivially not being in CR. The above counter-example $\overline{\mathsf{H}}$ is more meaningful because $2^{-\overline{\mathsf{H}}.\mathsf{ol}}$ could be negligible.

The following is a reset-security analog of Proposition 4.4, showing that $\mathsf{UCE}[\mathcal{S}^{\rm srs}]$ security neither implies, nor is implied by collision resistance. PRF security doesn't imply $\mathsf{UCE}[\mathcal{S}^{\rm srs}]$ security, but whether $\mathsf{UCE}[\mathcal{S}^{\rm srs}]$ security implies PRF security remains open. As usual, any non-containment $\mathsf{B} \not\subseteq \mathsf{A}$ assumes $\mathsf{B} \neq \emptyset$.

Proposition 4.5 (1) $\mathsf{PRF} \not\subseteq \mathsf{UCE}[\mathcal{S}^{\rm srs}]$, (2) $\mathsf{UCE}[\mathcal{S}^{\rm srs}] \not\subseteq \mathsf{CR}$, and (3) $\mathsf{CR} \not\subseteq \mathsf{UCE}[\mathcal{S}^{\rm srs}]$.

Proof: Parts (1) and (3) are direct corollaries of Propositions 4.2 and 4.4. For part (2), a trivial counter-example is a family $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$ with $\lambda \in \mathsf{H.IL}(\lambda)$ and $\mathsf{H.ol}(\lambda) = 1$ for all $\lambda \in \mathbb{N}$, such a family trivially not

being in CR. We'd like however to give a more meaningful counter-example, namely an $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}] \setminus \mathsf{CR}$ with $2^{-\mathsf{H.ol}}$ negligible. Towards this, let $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$ be a family with output length $\mathsf{H.ol}$ such that $2^{-\mathsf{H.ol}}$ is negligible and $\mathsf{H.IL}(\cdot) = \mathbb{N}$. Define a hash family $\overline{\mathsf{H}}$ as follows: (i) $\overline{\mathsf{H}}.\mathsf{Kg} = \mathsf{H.Kg}$ and (ii) $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, x, 1^\ell) = \mathsf{H.Ev}(1^\lambda, hk, x, 1^\ell)$ if $x \neq hk$, and $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, hk, 1^\ell) = \overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, \overline{hk}, 1^\ell)$, where $\overline{hk}$ is the complement of hk. Note $\overline{\mathsf{H}}.\mathsf{ol} = \mathsf{H.ol}$ and $\overline{\mathsf{H}}.\mathsf{IL} = \mathsf{H.IL}$. Then $\overline{\mathsf{H}} \notin \mathsf{CR}$, as $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, hk, 1^\ell) = \overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, \overline{hk}, 1^\ell)$. To prove that $\overline{\mathsf{H}} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$, consider arbitrary PT statistically reset-secure source $\overline{S}$ and PT distinguisher $\overline{D}$. Assume that $\overline{S}$ never repeats an oracle query. Let q be a polynomial such that the number of oracle queries of $\overline{S}$ in game $\mathrm{UCE}^{\overline{S},\overline{D}}_{\overline{\mathsf{H}}}(\lambda)$ is at most $q(\lambda)$ for all $\lambda \in \mathbb{N}$. We'll construct statistically reset-secure source S and distinguisher D such that

$$\mathsf{Adv}^{\mathsf{uce}}_{\overline{\mathsf{H}},\overline{S},\overline{D}}(\cdot) \leq \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + \frac{q}{2^{\mathsf{H.ol}}} \ . \tag{2}$$

The claim then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$ and $2^{-\mathsf{H.ol}}$ is negligible. The constructions of S and D are shown below:

$\underline{S^{\text{HASH}}(1^\lambda)}$
$\mathsf{done} \leftarrow \mathsf{false}$; $\overline{L} \leftarrow\!\!\$\ \overline{S}^{\text{HASHSIM}}(1^\lambda)$
If done then return 1
Return $0 \,\|\, \overline{L}$

$\underline{\text{HASHSIM}(x, 1^\ell)}$
$y \leftarrow \text{HASH}(x, 1^\ell)$
If $(|x| = \mathsf{H.kl}(\lambda)$ and $y = \overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, x, x, 1^{\mathsf{H.ol}(\lambda)}))$ then $\mathsf{done} \leftarrow \mathsf{true}$
Return y

$\underline{D^{\text{HASH}}(1^\lambda, hk, L)}$
If $L[1] = 0$ then
  $\overline{L} \leftarrow L[2, |L|]$
  $b' \leftarrow\!\!\$\ \overline{D}(1^\lambda, hk, \overline{L})$
  Return b'
Return 1

Let $\overline{b}$ and b be the challenge bits of game $\mathrm{UCE}^{\overline{S},\overline{D}}_{\overline{\mathsf{H}}}(\lambda)$ and game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\lambda)$ respectively. Then

$$\begin{aligned}
\Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 1\,] &\geq \Pr[\,\mathrm{UCE}^{\overline{S},\overline{D}}_{\overline{\mathsf{H}}}(\cdot) \,|\, \overline{b} = 1\,] \\
\Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 0\,] &\geq \Pr[\,\mathrm{UCE}^{\overline{S},\overline{D}}_{\overline{\mathsf{H}}}(\cdot) \,|\, \overline{b} = 0\,] - \frac{q}{2^{\mathsf{H.ol}}} \ .
\end{aligned}$$

Summing yields Equation (2). What's left is to prove that S is statistically reset-secure. Let R be a reset-adversary. Consider the following reset-adversary $\overline{R}$:

$$\underline{\overline{R}^{\text{HASH}}(1^\lambda, \overline{L})}$$
$$b' \leftarrow\!\!\$\ R^{\text{HASH}}(1^\lambda, 0 \,\|\, \overline{L})\ ;\ \text{Return } b'$$

Let $\overline{c}$ and c be the challenge bits of game $\mathrm{Reset}^{\overline{R}}_{\overline{S}}(\lambda)$ and $\mathrm{Reset}^{R}_{S}(\lambda)$ respectively. Then

$$\begin{aligned}
\Pr[\,\mathrm{Reset}^{R}_{S}(\cdot) \,|\, c = 1\,] &\leq \Pr[\,\mathrm{Reset}^{\overline{R}}_{\overline{S}}(\cdot) \,|\, \overline{c} = 1\,] + \frac{q}{2^{\mathsf{H.ol}}} \\
\Pr[\,\mathrm{Reset}^{R}_{S}(\cdot) \,|\, c = 0\,] &\leq \Pr[\,\mathrm{Reset}^{\overline{R}}_{\overline{S}}(\cdot) \,|\, \overline{c} = 0\,] + \frac{q}{2^{\mathsf{H.ol}}} \ .
\end{aligned}$$

Hence $\mathsf{Adv}^{\mathsf{reset}}_{R,S}(\cdot) \leq \mathsf{Adv}^{\mathsf{reset}}_{\overline{R},\overline{S}}(\cdot) + 2q/2^{\mathsf{H.ol}}$. ∎

4.5 From FOL to VOL

We show that we can build a UCE-secure family with variable output length (VOL) from a UCE-secure family with fixed output length (FOL). The construction is simple, namely to run the FOL evaluation algorithm in counter mode. Details follow.

Let H be the given FOL function family, having output length $\mathsf{H.ol}$. For simplicity assume $\mathsf{H.IL}(\lambda) = \mathbb{N}$ for all $\lambda \in \mathbb{N}$, meaning inputs of any length are allowed. We build a VOL function family $\overline{\mathsf{H}} = \mathsf{Extend}[\mathsf{H}]$,

$\underline{\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, x, 1^\ell)}$
$l \leftarrow \lceil \ell / \mathsf{H.ol}(\lambda) \rceil$
For $i = 1, \ldots, l$ do
  $y_i \leftarrow 1^\ell \,\|\, 0 \,\|\, 1^i \,\|\, 0 \,\|\, x$; $h_i \leftarrow \mathsf{H.Ev}(1^\lambda, hk, y_i, 1^{\mathsf{H.ol}(\lambda)})$
$h \leftarrow h_1 \,\|\, \cdots \,\|\, h_l$; Return $h[1, \ell]$

Figure 7: **Construction of a VOL family $\overline{\mathsf{H}}$ from a FOL family H.**

also with $\overline{\mathsf{H}}.\mathsf{IL}(\lambda) = \mathbb{N}$, but now with $\overline{\mathsf{H}}.\mathsf{OL}(\lambda)$ also equal to $\mathbb{N}$ for all $\lambda \in \mathbb{N}$, meaning output lengths can be arbitrary. We let $\overline{\mathsf{H}}.\mathsf{Kg} = \mathsf{H.Kg}$, meaning keys for the new family are those of the old family. The new evaluation algorithm $\overline{\mathsf{H}}.\mathsf{Ev}$ is described in Fig. 7.

Theorem 4.6 (1) If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm sup}]$, then $\mathsf{Extend}[\mathsf{H}] \in \mathsf{UCE}[\mathcal{S}^{\rm sup}]$, and (2) If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$, then $\mathsf{Extend}[\mathsf{H}] \in \mathsf{UCE}[\mathcal{S}^{\rm srs}]$.

Proof: Let $\overline{\mathsf{H}} = \mathsf{Extend}[\mathsf{H}]$. Let $\overline{S}$ be a source and D a distinguisher. We construct a source S such that

$$\mathsf{Adv}^{\mathsf{uce}}_{\overline{\mathsf{H}},\overline{S},D}(\cdot) = \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) \ . \tag{3}$$

Without loss of generality, we assume that $\overline{S}$ never repeats an oracle query. We build S from $\overline{S}$ as shown below:

$\underline{S^{\text{HASH}}(1^\lambda)}$	$\underline{\text{HASHSIM}(x, 1^\ell)}$
$L \leftarrow\!\!\$\ \overline{S}^{\text{HASHSIM}}(1^\lambda)$	$l \leftarrow \lceil \ell / \mathsf{H.ol}(\lambda) \rceil$
Return L	For $i = 1, \ldots, l$ do
	$y_i \leftarrow 1^\ell \,\|\, 0 \,\|\, 1^i \,\|\, 0 \,\|\, x$; $h_i \leftarrow \text{HASH}(y_i, 1^{\mathsf{H.ol}(\lambda)})$
	$h \leftarrow h_1 \,\|\, \cdots \,\|\, h_l$; Return $h[1, \ell]$

The assumption that $\overline{S}$ never repeats an oracle query implies that the oracle queries made by S are all distinct. (This follows from the way these queries are encoded.) Letting $\overline{b}, b$ denote the challenge bits in games $\text{UCE}^{\overline{S},D}_{\overline{\mathsf{H}}}(\cdot)$ and $\text{UCE}^{S,D}_{\mathsf{H}}(\cdot)$ respectively, we thus have

$$\Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 0\,] = \Pr[\,\text{UCE}^{\overline{S},D}_{\overline{\mathsf{H}}}(\cdot) \,|\, \overline{b} = 0\,]$$
$$\Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 1\,] = \Pr[\,\text{UCE}^{\overline{S},D}_{\overline{\mathsf{H}}}(\cdot) \,|\, \overline{b} = 1\,] \ .$$

This yields Equation (3). For part (1), suppose that $\overline{S}$ is statistically unpredictable. We'll show that S is also statistically unpredictable. By Lemma 4.3 it suffices to show that S is simple statistically unpredictable. Given a simple predictor P' for S, we build a simple predictor $\overline{P}'$ for $\overline{S}$ such that $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le \mathsf{Adv}^{\mathsf{spred}}_{\overline{S},\overline{P}'}(\cdot)$. The conclusion follows because we assumed that $\overline{S}$ is statistically unpredictable and Lemma 4.3 says it is thus also simple statistically unpredictable. Predictor $\overline{P}'(1^\lambda, L)$ lets $w \leftarrow\!\!\$\ P'(1^\lambda, L)$. It then parses w as $1^\ell \| 0 \| 1^i \| 0 \| x$ and returns x. The claimed bound is easily verified. For part (2), suppose that $\overline{S}$ is statistically reset-secure. We'll show that S is also statistically reset-secure. Consider an arbitrary reset-adversary R. Wlog, assume that R doesn't repeat an oracle query. We'll build a reset-adversary $\overline{R}$ from R as shown below:

$\underline{\overline{R}^{\text{HASH}}(1^\lambda, L)}$	$\underline{\text{HASHSIM}(x, 1^{\mathsf{H.ol}(\lambda)})}$
$b' \leftarrow\!\!\$\ R^{\text{HASHSIM}}(1^\lambda, L)$	If $(x = 1^\ell \,\|\, 0 \,\|\, 1^i \,\|\, 0 \,\|\, v)$ and $(i \le \lceil \ell / \mathsf{H.ol}(\lambda) \rceil)$ then
Return b'	$l \leftarrow \lceil \ell / \mathsf{H.ol}(\lambda) \rceil$; $y \leftarrow \text{HASH}(v, 1^\ell)$; $r \leftarrow\!\!\$\ \{0,1\}^{l\mathsf{H.ol}(\lambda) - \ell}$
	If $i < l$ then return $y[(i-1)\mathsf{H.ol}(\lambda) + 1, i\mathsf{H.ol}(\lambda)]$
	Else return $y[(i-1)\mathsf{H.ol}(\lambda) + 1, \ell] \,\|\, r$
	$z \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$; Return z

Hence $\Pr[\,\mathrm{Reset}_S^R(\cdot)\,|\,a=0\,]=\Pr[\,\mathrm{Reset}_{\overline{S}}^{\overline{R}}(\cdot)\,|\,\overline{a}=0\,]$, where $\overline{a}, a$ denote the challenge bits in games $\mathrm{Reset}_{\overline{S}}^{\overline{R}}(\cdot)$ and $\mathrm{Reset}_S^R(\cdot)$ respectively. We claim that

$$\Pr[\,\mathrm{Reset}_S^R(\cdot)\,|\,a=1\,]=\Pr[\,\mathrm{Reset}_{\overline{S}}^{\overline{R}}(\cdot)\,|\,\overline{a}=1\,],$$

Subtracting, we have $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot)=\mathsf{Adv}^{\mathsf{reset}}_{\overline{S},\overline{R}}(\cdot)$. To justify this claim, consider games G_1 and G_2 below.

$\underline{\textsc{Main}\ \mathrm{G}_1^{\overline{S},R}(\lambda)}$
$L \leftarrow\!\!\$\ \overline{S}^{\textsc{Hash}}(1^\lambda)\ ;\ b' \leftarrow R^{\textsc{HashSim}}(1^\lambda, L)$
Return $(b'=1)$

$\underline{\textsc{Hash}(x, 1^\ell)}$
$y \leftarrow\!\!\$\ \{0,1\}^\ell\ ;\ l \leftarrow \lceil \ell/\mathsf{H.ol}(\lambda)\rceil\ ;\ r \leftarrow\!\!\$\ \{0,1\}^{l\mathsf{H.ol}(\lambda)-\ell}$
For $i=1$ to l do
  $y_i \leftarrow 1^\ell \,\|\, 0 \,\|\, 1^i \,\|\, 0 \,\|\, x$
  If $i<l$ then $T[y_i] \leftarrow y[(i-1)\mathsf{H.ol}(\lambda)+1, i\mathsf{H.ol}(\lambda)]$
  Else $T[y_i] \leftarrow y[(i-1)\mathsf{H.ol}(\lambda)+1, \ell] \,\|\, r$
Return y

$\underline{\textsc{HashSim}(x, 1^{\mathsf{H.ol}(\lambda)})}$
If $T[x]=\bot$ then $T[x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
Return $T[x]$

$\underline{\textsc{Main}\ \mathrm{G}_2^{\overline{S},R}(\lambda)}$
$L \leftarrow\!\!\$\ \overline{S}^{\textsc{Hash}}(1^\lambda)\ ;\ b' \leftarrow R^{\textsc{HashSim}}(1^\lambda, L)$
Return $(b'=1)$

$\underline{\textsc{Hash}(x, 1^\ell)}$
$y \leftarrow\!\!\$\ \{0,1\}^\ell\ ;\ l \leftarrow \lceil \ell/\mathsf{H.ol}(\lambda)\rceil$
For $i=1$ to l do
  $y_i \leftarrow 1^\ell \,\|\, 0 \,\|\, 1^i \,\|\, 0 \,\|\, x$
  If $i<l$ then $T[y_i] \leftarrow y[(i-1)\mathsf{H.ol}(\lambda)+1, i\mathsf{H.ol}(\lambda)]$
  Else $V[y_i] \leftarrow y[(i-1)\mathsf{H.ol}(\lambda)+1, \ell]$
Return y

$\underline{\textsc{HashSim}(x, 1^{\mathsf{H.ol}(\lambda)})}$
If $V[x]\neq\bot$ then
  $r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)-|V[x]|}\ ;\ T[x] \leftarrow V[x] \,\|\, r$
If $T[x]=\bot$ then $T[x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
Return $T[x]$

The two games are identical, as the source $\overline{S}$ never reads strings r, so we can postpone creating them until R makes queries to read them. Moreover, $\Pr[\mathrm{G}_1^{\overline{S},R}(\cdot)]=\Pr[\,\mathrm{Reset}_S^R(\cdot)\,|\,a=1\,]$, and $\Pr[\mathrm{G}_2^{\overline{S},R}(\cdot)]=\Pr[\,\mathrm{Reset}_{\overline{S}}^{\overline{R}}(\cdot)\,|\,\overline{a}=1\,]$. The claim then follows. ▌

4.6 mUCE security

In UCE, there is a single target key hk. Some of our applications will depend on an extension involving multiple keys. Here we define this mUCE extension of UCE.

Framework. Let H be a family of functions. Consider game $\mathrm{mUCE}_{\mathsf{H}}^{S,D}(\lambda)$ of Fig. 8 involving a multi-source S and distinguisher D. Adversary S now begins by returning a unary-encoded integer $n\geq 1$ indicating the number of instances, together with state information t. The game creates n, independent keys. The oracle Hash given to S now allows it to query any instance $i\in[1,n]$ of its choice. As before S returns leakage L based on which the distinguisher D, now given the entire vector $\mathbf{hk}$ of keys, returns its guess bit b'. The mUCE-advantage of (S,D) is defined for $\lambda\in\mathbb{N}$ by

$$\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\lambda)=2\Pr[\mathrm{mUCE}_{\mathsf{H}}^{S,D}(\lambda)]-1\ . \tag{4}$$

Let $\mathcal{S}$ be a class of multi-sources and $\mathcal{D}$ a class of distinguishers. Then we let $\mathsf{mUCE}[\mathcal{S},\mathcal{D}]$ be the set of all H such that $\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible for all $(S,D)\in\mathcal{S}\times\mathcal{D}$. We let $\mathsf{mUCE}[\mathcal{S}]=\mathsf{mUCE}[\mathcal{S},\mathcal{D}^{\mathsf{poly}}]$.

Classes. The classes defined in the single-key case have natural multi-key analogues, as we now explain. For $\lambda\in\mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{m\text{-}pred}}_{S,P}(\lambda)=\Pr[\mathrm{mPred}_S^P(\lambda)]$ where game $\mathrm{mPred}_S^P(\lambda)$ is in Fig. 8. We say that P is a computational predictor if it is PT, and we say it is a statistical predictor if there exist polynomials q, q' such that for all $\lambda\in\mathbb{N}$, predictor P makes at most $q(\lambda,n)$ oracle queries and outputs a set Q' of size at most $q'(\lambda,n)$ in game $\mathrm{Pred}_S^P(\lambda)$, where the number of keys n is defined via the output of S in the first line of the game. We stress that in this case the predictor need not be PT. We say S is computationally (resp., statistically) unpredictable if $\mathsf{Adv}^{\mathsf{m\text{-}pred}}_{S,P}(\cdot)$ is negligible for all computational (resp., statistical) predictors P. We let $\mathcal{S}^{\text{cup-m}}$ be the class of all PT, computationally unpredictable multi sources and $\mathcal{S}^{\text{sup-m}}\subseteq\mathcal{S}^{\text{cup-m}}$

$\underline{\text{MAIN mUCE}^{S,D}_{\mathsf{H}}(\lambda)}$	$\underline{\text{MAIN mPred}^{P}_{S}(\lambda)}$	$\underline{\text{MAIN mSPred}^{P'}_{S}(\lambda)}$
$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$	$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$	$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$
For $i = 1$ to n do $\mathbf{hk}[i] \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$\mathsf{done} \leftarrow \mathsf{false}$; $Q \leftarrow \emptyset$	$Q \leftarrow \emptyset$
$b \leftarrow\!\!\$\ \{0,1\}$; $L \leftarrow\!\!\$\ S^{\text{HASH}}(1^n, t)$	$L \leftarrow\!\!\$\ S^{\text{HASH}}(1^n, t)$; $\mathsf{done} \leftarrow \mathsf{true}$	$L \leftarrow\!\!\$\ S^{\text{HASH}}(1^n, t)$
$b' \leftarrow\!\!\$\ D(1^\lambda, \mathbf{hk}, L)$	$Q' \leftarrow\!\!\$\ P^{\text{HASH}}(1^\lambda, 1^n, L)$	$x \leftarrow\!\!\$\ P'(1^\lambda, 1^n, L)$
Return $(b' = b)$	Return $(Q \cap Q' \neq \emptyset)$	Return $(x \in Q)$
$\underline{\text{HASH}(x, 1^\ell, i)}$	$\underline{\text{HASH}(x, 1^\ell, i)}$	$\underline{\text{HASH}(x, 1^\ell, i)}$
If $T[x, \ell, i] = \perp$ then	If $\mathsf{done} = \mathsf{false}$ then $Q \leftarrow Q \cup \{x\}$	$Q \leftarrow Q \cup \{x\}$
If $b = 1$ then $T[x, \ell, i] \leftarrow \mathsf{H.Ev}(1^\lambda, \mathbf{hk}[i], x, 1^\ell)$	If $T[x, \ell, i] = \perp$ then	If $T[x, \ell, i] = \perp$ then
Else $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$	$T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$	$T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell, i]$	Return $T[x, \ell, i]$	Return $T[x, \ell, i]$

Figure 8: **Games** mUCE, mPred**, and** mSPred **used to define mUCE security of family of functions** H**, and game** mSPred **defining the simplified but equivalent form of unpredictability.** Here S is the multi-source, D is the distinguisher, P is the predictor and P' is the simple predictor.

---

the class of all PT, statistically unpredictable multi sources. The associated classes (assumptions) are $\mathsf{mUCE}[\mathcal{S}^{\text{cup-m}}] \subseteq \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$.

As with UCE, unpredictability is equivalent to simple unpredictability. In detail, consider game $\text{mSPred}^{P'}_{S}(\lambda)$ of Fig. 8 and let $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{S,P'}(\lambda) = \Pr[\text{mSPred}^{P'}_{S}(\lambda)]$. Say that P' is computational if it is PT, and say it is statistical if it may not be PT. We say that multi-source S is simple computationally (resp., statistically) unpredictable if $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{S,P'}(\cdot)$ is negligible for all simple computational (resp., statistical) predictors P'. The following analogue of Lemma 4.3 shows equivalence of simple unpredictability and unpredictability for multi-sources. The proof of Lemma 4.7 is similar to the proof of Lemma 4.3 and is omitted.

Lemma 4.7 Let S be a multi-source. Then S is computationally unpredictable if and only if it is simple computationally unpredictable and S is statistically unpredictable if and only if it is simple statistically unpredictable.

We likewise define reset security for multi-sources. Consider game $\text{mReset}^{R}_{S}(\lambda)$ of Fig. 4. Let $\mathsf{Adv}^{\mathsf{m\text{-}reset}}_{S,R}(\lambda) = 2\Pr[\text{mReset}^{R}_{S}(\lambda)] - 1$. We say R is a computational reset adversary if it is PT, and we say it is a statistical reset adversary if there exists a polynomial q such that for all $\lambda \in \mathbb{N}$, reset adversary R makes at most $q(\lambda)$ oracle queries in game $\text{mReset}^{R}_{S}(\lambda)$. We say multi source S is computationally (resp., statistically) reset-secure if $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot)$ is negligible for all computational (resp., statistical) reset adversaries R. We let $\mathcal{S}^{\text{crs-m}}$ be the class of all PT, computationally reset-secure sources and $\mathcal{S}^{\text{srs-m}} \subseteq \mathcal{S}^{\text{crs-m}}$ the class of all PT, statistically reset-secure sources. The associated classes (assumptions) are $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}] \subseteq \mathsf{mUCE}[\mathcal{S}^{\text{srs-m}}]$. The analogue of Proposition 4.2 holds also in the multi-key case.

One may also define the multi-key analogues of the split and parallel sources, leading to classes $\mathcal{S}^{\text{splt-m}}$ and $\mathcal{S}^{\text{prl-m}}_{\tau,\sigma,q}$. Since we don't use these classes in this paper, we omit their definitions.

5 Applications of UCE

We show how UCE- or mUCE-secure families can securely instantiate ROs to yield (new) standard-model solutions for a variety of goals. Specifically, we detail the claims of Section 1.4.

5.1 Hardcore functions for any OWF

A hardcore function for a one-way function f extracts from x bits that are indistinguishable from random even given $f(x)$ [28, 107, 77]. The concept has been central in the development of the theory of public-key

$\underline{\text{MAIN } \mathrm{HC}^A_{\mathsf{F},\mathsf{H}}(\lambda)}$
$b \leftarrow\!\!\$\ \{0,1\}$; $fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, x, 1^{\mathsf{F.ol}(\lambda)})$
If $b = 1$ then $r \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})$ else $r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$b' \leftarrow\!\!\$\ A(1^\lambda, fk, hk, y, r)$; Return $(b = b')$

Figure 9: **Game defining security of H as a hardcore function for F.**

encryption, and hardcore functions have been sought and found for many specific one-way functions [28, 107, 77, 82, 3]. Goldreich and Levin [74] present a hardcore function able to extract a single bit from any one-way function. But ROs are "ideal" hardcore functions, able to extract as many pseudorandom bits as desired from any one-way function. We show how the RO here can be UCE-instantiated. Thus, we show that UCE-secure families are hardcore functions for any one-way function f, allowing us to extract from x any number of bits that remain indistinguishable from random to an adversary given $f(x)$.

DEFINITIONS. Let F be a family of functions with input length $\mathsf{F.il}$ and output length $\mathsf{F.ol}$. We say that F is one-way if $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},I}(\cdot)$ is negligible for all PT I, where $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},I}(\lambda) = \Pr[I(1^\lambda, fk, y) = x]$ in the experiment $fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, x)$. Let OW be the set of all F that are one-way. Let H be a family of functions with the same input length as F and output length $\mathsf{H.ol}$. We say that H is hardcore for F if $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},A}(\cdot)$ is negligible for all PT A, where $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},A}(\lambda) = 2\Pr[\mathrm{HC}^A_{\mathsf{F},\mathsf{H}}(\lambda)] - 1$ and game $\mathrm{HC}^A_{\mathsf{F},\mathsf{H}}(\lambda)$ is in Fig. 9. Let $\mathsf{HC}[\mathsf{F}]$ be the set of all H that are hardcore for F.

RESULTS. The following says that if F is one-way and H is UCE-secure then H is hardcore for F. The assumption used is a weak version of $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}}]$-security, namely $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}]$-security, where $\mathcal{S}^{\mathrm{one}}$ is the class of sources that make at most one oracle query to HASH. We remark that the result could alternatively be obtained under a $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}]$ assumption where τ, σ depend only on F and $q = 1$, so that two, different assumptions suffice for security. The proof would be essentially the same, so we omit the details of this second result.

Theorem 5.1 If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}]$ and $\mathsf{F} \in \mathsf{OW}$ then $\mathsf{H} \in \mathsf{HC}[\mathsf{F}]$.

Proof of Theorem 5.1: Given a PT adversary A for game $\mathrm{HC}^A_{\mathsf{F},\mathsf{H}}(\cdot)$, we build a source $S \in \mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}$ and a distinguisher $D \in \mathcal{D}^{\mathsf{poly}}$ such that

$$\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},A}(\cdot) = \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)\ . \tag{5}$$

The assumption $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}]$ implies the right-hand side of Equation (5) is negligible, which yields the theorem. The constructions of S and D are shown below:

$\underline{S^{\mathrm{HASH}}(1^\lambda)}$	$\underline{D(1^\lambda, hk, L)}$	$\underline{I(1^\lambda, fk, y)}$
$fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}$	$((fk, y), r) \leftarrow L$	$r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, x)$; $r \leftarrow \mathrm{HASH}(x, 1^{\mathsf{H.ol}(\lambda)})$	$b' \leftarrow\!\!\$\ A(1^\lambda, fk, hk, y, r)$	$x' \leftarrow\!\!\$\ P'(1^\lambda, ((fk, y), r))$
$L \leftarrow ((fk, y), r)$; Return L	Return b'	Return x'

Equation (5) is easily verified, and S makes only a single query to HASH. Now we claim that S is computationally unpredictable. By Lemma 4.3 it suffices to show that S is simple computationally unpredictable. Given a simple computational predictor adversary P' we define I as shown above. Then we have

$$\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) = \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},I}(\cdot)\ .$$

But the assumption $\mathsf{F} \in \mathsf{OW}$ implies the right-hand-side is negligible, which shows that S is simple computationally unpredictable. Finally we exhibit, below, PT algorithms S_0, S_1 such that $S = \mathsf{Splt}[S_0, S_1]$—

$\underline{\text{MAIN IND-CPA}^A_{\mathsf{PKE}}(\lambda)}$	$\underline{\mathsf{PKE.Kg}(1^\lambda)}$	$\underline{\mathsf{PKE.Enc}(1^\lambda, (ek, hk), m)}$
$b \leftarrow\!\!\$\ \{0,1\}$	$(ek, dk) \leftarrow\!\!\$\ \mathsf{TF.EKg}(1^\lambda)$	$x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{TF.il}(\lambda)}$
$(ek, dk) \leftarrow\!\!\$\ \mathsf{PKE.Kg}(1^\lambda)$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$w \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^{\mathsf{H.ol}(\lambda)})$
$b' \leftarrow\!\!\$\ A^{\mathrm{LR}}(1^\lambda, ek)$	Return $((ek, hk), (dk, hk))$	Return $(\mathsf{TF.Ev}(1^\lambda, ek, x), w \oplus m)$
Return $(b = b')$		$\underline{\mathsf{PKE.Dec}(1^\lambda, (dk, hk), (y, z))}$
$\underline{\mathrm{LR}(m_0, m_1)}$		$x \leftarrow \mathsf{TF.Inv}(1^\lambda, dk, y)$
$c \leftarrow\!\!\$\ \mathsf{PKE.Enc}(1^\lambda, ek, m_b)$		Return $\mathsf{H.Ev}(1^\lambda, hk, x, 1^{\lvert z\rvert}) \oplus z$
Return c		

Figure 10: **Left:** The IND-CPA game. **Right:** PKE scheme $\mathsf{PKE} = \mathsf{BR93}[\mathsf{H}, \mathsf{TF}]$.

$\underline{S_0(1^\lambda)}$	$\underline{S_1(1^\lambda, \mathbf{y})}$
$fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)\ ;\ x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}$	$r \leftarrow \mathbf{y}[1]$
$y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, x)\ ;\ \mathbf{x}[1] \leftarrow x\ ;\ \boldsymbol{\ell}[1] \leftarrow \mathsf{H.ol}(\lambda)$	Return r
Return $((fk, y), \mathbf{x}, 1^{\boldsymbol{\ell}})$	

This shows that $S \in \mathcal{S}^{\mathrm{splt}}$, concluding the proof. ▮

5.2 Instantiating the BR93 PKE scheme

BR93 [23] gave a simple PKE scheme which encrypts m by picking x at random and returning $(f(x), \mathsf{RO}(x) \oplus m)$ where the public key f is an injective trapdoor function whose inverse is the secret key. They showed that this is IND-CPA when RO is a RO. We show that instantiating RO with a UCE family maintains IND-CPA security. We note that we show full (adaptive) IND-CPA, not IND-CPA-KI. This is because the points to which the RO is applied in this scheme do not depend on the messages.

DEFINITIONS. A family of functions TF with input length $\mathsf{TF.il}$ and output length $\mathsf{TF.ol}$ is said to be trapdoor if there are (additional) PT algorithms $\mathsf{TF.EKg}, \mathsf{TF.Inv}$, the second deterministic, such that the following hold. Extended key-generation algorithm $\mathsf{TF.EKg}(1^\lambda)$ returns a pair (ek, dk) of keys, the second called the trapdoor. The usual $\mathsf{TF.Kg}(1^\lambda)$ algorithm lets $(ek, dk) \leftarrow\!\!\$\ \mathsf{TF.EKg}(1^\lambda)$ and returns ek. Finally $\mathsf{TF.Inv}(1^\lambda, dk, \mathsf{TF.Ev}(1^\lambda, ek, x)) = x$ for all $\lambda \in \mathbb{N}$, all $(ek, dk) \in [\mathsf{TF.EKg}(1^\lambda)]$ and all $x \in \{0,1\}^{\mathsf{TF.il}(\lambda)}$.

A PKE scheme PKE as usual specifies a triple of PT algorithms, the last deterministic. 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 decrypt. We say that PKE is IND-CPA-secure if $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(\cdot)$ is negligible for all PT A, where $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(\lambda) = 2\Pr[\text{IND-CPA}^A_{\mathsf{PKE}}(\lambda)] - 1$ and game $\text{IND-CPA}^A_{\mathsf{PKE}}(\lambda)$ is shown in Fig. 10. Messages m_0, m_1 queried to LR are required to be of the same length and A may be assumed to make only one oracle query. Let $\mathsf{IND\text{-}CPA}$ be the set of all PKE that are IND-CPA secure.

RESULTS. Let TF be a trapdoor family of functions. Let H be a family of functions with the same input length as F and output length $\mathsf{H.ol}$. Our instantiated BR93 scheme is represented by a transform $\mathsf{BR93}$ that associates to H and TF the PKE scheme $\mathsf{PKE} = \mathsf{BR93}[\mathsf{H}, \mathsf{TF}]$ defined in Fig. 10. The message length of the scheme is $\mathsf{PKE.il} = \mathsf{H.ol}$. The assumption, as in Theorem 5.1, is $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}]$-security. Again, the result could also be obtained under a $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}]$ assumption where τ, σ depend only on TF and $q = 1$.

Theorem 5.2 If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{splt}} \cap \mathcal{S}^{\mathrm{one}}]$ and $\mathsf{TF} \in \mathsf{OW}$ then $\mathsf{BR93}[\mathsf{H}, \mathsf{TF}] \in \mathsf{IND\text{-}CPA}$.

Proof of Theorem 5.2: Theorem 5.2 is a simple corollary of Theorem 5.1, meaning we do not have to use UCE directly. Given an adversary A for game $\text{INDCPA}^A_{\mathsf{PKE}}(\lambda)$, where $\mathsf{PKE} = \mathsf{BR93}[\mathsf{H}, \mathsf{TF}]$, we build the following adversary B for game $\text{HC}^B_{\mathsf{TF},\mathsf{H}}(\lambda)$:

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

Figure 11: **Left:** The IND game. **Right:** D-PKE scheme $\mathsf{DE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$.

$\underline{B(1^\lambda, ek, hk, y, r)}$	$\underline{\text{LRSIM}(m_0, m_1)}$
$d \leftarrow\!\!\$\ \{0,1\}\ ;\ d' \leftarrow\!\!\$\ A^{\text{LRSIM}}(1^\lambda, (ek, hk))$	Return $(y, r \oplus m_d)$
If $(d = d')$ then $b' \leftarrow 1$ else $b' \leftarrow 0$	
Return b'	

Adversary A makes a single oracle query, consisting of a pair $m_0, m_1 \in \{0,1\}^{\mathsf{H.ol}(\lambda)}$ of messages, in response to which B returns the ciphertext shown. Letting b denote the challenge bit in game $\text{HC}^B_{\mathsf{TF},\mathsf{H}}(\lambda)$ we have

$$\Pr[\, d = d' \,|\, b = 1\,] = \frac{1}{2} + \frac{1}{2}\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(\cdot) \qquad \text{and} \qquad \Pr[\, d = d' \,|\, b = 0\,] = \frac{1}{2}\,.$$

Subtracting, we have $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{TF},\mathsf{H},A}(\lambda) = 0.5 \cdot \mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},A}(\lambda)$. ∎

5.3 Deterministic encryption

EwH is a simple and natural D-PKE scheme from [12] that deterministically encrypts m by using a randomized IND-CPA scheme with the coins derived by applying a RO to m. In the ROM the scheme is PRIV-secure [12] and equivalently IND-secure [15]. We show that instantiating the RO with a UCE hash family results in a scheme meeting the same notion of security in the standard model. Previous standard model schemes [30, 43] have met notions providing security only when one assumes messages are drawn from a blocksource [54], meaning each message has high min-entropy even given previous ones. Instantiated EwH however meets the original and full notions of [12, 15] which only make the necessary assumption that each individual 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.

DEFINITIONS. Let PKE be a PKE scheme as defined in Section 5.2. We say PKE is a D-PKE scheme if the encryption algorithm $\mathsf{PKE.Enc}$ is deterministic. The game defining the IND notion of security for D-PKE scheme DE, following [15], is in Fig. 11. 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 is a polynomial v, depending on the adversary, such that $|\mathbf{m}_0| = |\mathbf{m}_1| = v(\lambda)$ and $|\mathbf{m}_b[i]| = \mathsf{DE.il}(\lambda)$ for all $b \in \{0,1\}$ and $i \in [1, v(\lambda)]$. 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.

RESULTS. Let RE be a PKE scheme. Let $\mathsf{RE.ekl}\colon \mathbb{N} \to \mathbb{N}$ denote the length of its public keys, meaning $|ek| = \mathsf{RE.ekl}(\lambda)$ for all $(ek, dk) \in [\mathsf{RE.Kg}(1^\lambda)]$. 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 $\mathsf{H.il} = \mathsf{RE.il} + \mathsf{RE.ekl}$ and $\mathsf{H.ol} = \mathsf{RE.rl}$. Our standard-model instantiation of the ROM encrypt-with-hash transform of BBO07 [12] associates to RE and H the (standard-model) D-PKE scheme $\mathsf{DE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$ described in Fig. 11. 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 UCE-secure and RE is IND-CPA-secure. By $\mathsf{T}_{\mathsf{RE.Enc}}\colon \mathbb{N} \to \mathbb{N}$ we denote the running time of $\mathsf{RE.Enc}$, meaning $\mathsf{RE.Enc}(1^\lambda, ek, m; r)$ halts in $\mathsf{T}_{\mathsf{RE.Enc}}(\lambda)$ steps for all $\lambda \in \mathbb{N}$, all $(ek, dk) \in [\mathsf{RE.Kg}(1^\lambda)]$, all $m \in \{0,1\}^{\mathsf{RE.il}(\lambda)}$ and all $r \in \{0,1\}^{\mathsf{RE.rl}(\lambda)}$.

Theorem 5.3 Suppose $\mathsf{RE} \in \mathsf{IND\text{-}CPA}$. Suppose $\mathsf{H.il} = \mathsf{RE.il}+\mathsf{RE.ekl}$ and $\mathsf{H.ol} = \mathsf{RE.rl}$. Let $\tau = \mathcal{O}(\mathsf{T}_{\mathsf{RE.Enc}}+ \mathsf{RE.il} + \mathsf{RE.rl} + \mathsf{RE.ekl})$ and let $q = 1$. Let $\sigma = \mathcal{O}(\mathsf{RE.ekl})$. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}]$ then $\mathsf{EwH}[\mathsf{H}, \mathsf{RE}] \in \mathsf{IND}$.

We have let H have output length depending on RE so that its key length may be long compared to this output length. This seems important to prevent some attacks. Note that τ, σ depend only on the scheme RE, not on the running time of an adversary.

Proof of Theorem 5.3: Let $\mathsf{DE} = \mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$. Given a PT high min-entropy adversary $A = (A_1, A_2)$ for game $\mathrm{IND}^A_{\mathsf{DE}}(\cdot)$, we build a source $S \in \mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}$, a distinguisher $D \in \mathcal{D}^{\mathsf{poly}}$, and a PT adversary B_1 for game $\mathrm{INDCPA}^{B_1}_{\mathsf{RE}}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\cdot) \le 2\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + \mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{RE},B_1}(\cdot)\ . \tag{6}$$

The theorem follows from the assumptions $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q}]$ and $\mathsf{RE} \in \mathsf{IND\text{-}CPA}$. The constructions of S, D and B_1 are shown below, where v is the polynomial assumed associated to A as per the definitions:

$\underline{S^{\mathrm{HASH}}(1^\lambda)}$	$\underline{D(1^\lambda, hk, L)}$	$\underline{B_1^{\mathrm{LR}}(1^\lambda, ek)}$
$(ek, dk) \leftarrow\!\!\$\ \mathsf{RE.Kg}(1^\lambda)\ ;\ d \leftarrow\!\!\$\ \{0,1\}$	$((ek, d), \mathbf{c}) \leftarrow L$	$(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)$
$(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)$	$d' \leftarrow\!\!\$\ A_2(1^\lambda, (ek, hk), \mathbf{c})$	For $i = 1, \ldots, v(\lambda)$ do
For $i = 1, \ldots, v(\lambda)$ do	If $(d = d')$ then $b' \leftarrow 1$	$\quad \mathbf{c}[i] \leftarrow \mathrm{LR}(\mathbf{m}_0[i], \mathbf{m}_1[i])$
$\quad \mathbf{r}[i] \leftarrow\!\!\$\ \mathrm{HASH}(ek \Vert \mathbf{m}_d[i], 1^{\mathsf{RE.rl}(\lambda)})$	Else $b' \leftarrow 0$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$\quad \mathbf{c}[i] \leftarrow \mathsf{RE.Enc}(1^\lambda, ek, \mathbf{m}_d[i]; \mathbf{r}[i])$	Return b'	$b' \leftarrow\!\!\$\ A_2(1^\lambda, (ek, hk), \mathbf{c})$
$L \leftarrow ((ek, d), \mathbf{c})$		Return b'
Return L		

Letting b denote the challenge bit in game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot)$ we have

$$\Pr[\, d = d' \,|\, b = 1\,] = \frac{1}{2} + \frac{1}{2}\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{DE},A}(\cdot) \qquad \text{and} \qquad \Pr[\, d = d' \,|\, b = 0\,] = \frac{1}{2} + \frac{1}{2}\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{RE},B_1}(\cdot)\ .$$

The second equation above exploits the assumption that $\mathbf{m}_d[1], \ldots, \mathbf{m}_d[n]$ are all distinct. Subtracting and re-arranging terms, we have Equation (6).

We now show that S is computationally unpredictable. By Lemma 4.3 it suffices to show that S is simple computationally 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. Thus, letting P' be a simple computational predictor, we construct PT B_2 such that

$$\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le \mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{RE},B_2}(\cdot) + v(\cdot) \cdot \mathrm{Guess}_A(\cdot)\ . \tag{7}$$

The assumption that A has high min-entropy and that $\mathsf{RE} \in \mathsf{IND\text{-}CPA}$ mean the left-hand-side of Equation (7) is negligible, and thus S is simple computational unpredictable. We construct B_2 as follows:

$\underline{B_2^{\mathrm{LR}}(1^\lambda, ek)}$
$(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)\ ;\ d \leftarrow\!\!\$\ \{0,1\}$
For $i = 1, \ldots, v(\lambda)$ do $\mathbf{m}_2[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{RE.il}(\lambda)}\ ;\ \mathbf{c}[i] \leftarrow \mathrm{LR}(\mathbf{m}_2[i], \mathbf{m}_d[i])$
$L \leftarrow ((ek, d), \mathbf{c})\ ;\ x \leftarrow P'(1^\lambda, L)$
If $x \in \{\, ek \,\Vert\, \mathbf{m}_d[i]\ :\ 1 \le i \le v(\lambda)\,\}$ then $b' \leftarrow 1$ else $b' \leftarrow 0$
Return b'

$\underline{\text{MAIN IND\$-CDA}^{A}_{\mathsf{MLE}}(\lambda)}$	$\underline{\mathsf{CE.Pg}(1^\lambda)}$	$\underline{\mathsf{CE.Dec}(1^\lambda, hk, k, c)}$
$p \leftarrow\!\!\$\ \mathsf{MLE.Pg}(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$m \leftarrow \mathsf{SE.Dec}(1^\lambda, k, c)$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$	Return hk	Return m
For $i = 1$ to $\lvert\mathbf{m}\rvert$ do	$\underline{\mathsf{CE.Kg}(1^\lambda, hk, m)}$	$\underline{\mathsf{CE.Tag}(1^\lambda, hk, c)}$
$\quad \mathbf{k}[i] \leftarrow \mathsf{MLE.Kg}(1^\lambda, p, \mathbf{m}[i])$	$k \leftarrow \mathsf{H.Ev}(1^\lambda, hk, m, 1^{\mathsf{SE.kl}(\lambda)})$	Return c
$\quad \mathbf{c}_1[i] \leftarrow \mathsf{MLE.Enc}(1^\lambda, p, \mathbf{k}[i], \mathbf{m}[i])$	Return k	
$\quad \mathbf{c}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\lvert\mathbf{c}_1[i]\rvert}$	$\underline{\mathsf{CE.Enc}(1^\lambda, hk, k, m)}$	
$b' \leftarrow\!\!\$\ A_2(1^\lambda, p, \mathbf{c}_b)$	$c \leftarrow \mathsf{SE.Enc}(1^\lambda, k, m)$	
Return $(b' = b)$	Return c	

Figure 12: **Left:** The IND\$-CDA game. **Right:** MLE scheme $\mathsf{CE}[\mathsf{H}, \mathsf{SE}]$.

Letting b denote the challenge bit in game $\text{IND-CPA}^{B_2}_{\mathsf{RE}}(\cdot)$ we have

$$\Pr[\,b' = 1 \,|\, b = 1\,] = \mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \qquad \text{and} \qquad \Pr[\,b' = 1 \,|\, b = 0\,] \le v(\cdot) \cdot \text{Guess}_A(\cdot)\ .$$

Subtracting we obtain Equation (7).

Finally, we exhibit, below, PT algorithms S_0, S_1 such that $S = \mathsf{Prl}[S_0, S_1]$—

$\underline{S_0(1^\lambda)}$	$\underline{S_1^{\text{HASH}}(1^\lambda, (ek, m))}$
$(ek, dk) \leftarrow\!\!\$\ \mathsf{RE.Kg}(1^\lambda)\ ;\ d \leftarrow\!\!\$\ \{0,1\}$	$r \leftarrow\!\!\$\ \text{HASH}(ek \Vert m, 1^{\mathsf{RE.rl}(\lambda)})$
$(\mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda)$	$c \leftarrow \mathsf{RE.Enc}(1^\lambda, ek, m; r)$
Return $((ek, d), ((ek, \mathbf{m}_d[1]), \ldots, (ek, \mathbf{m}_d[v(\lambda)])))$	Return c

Note that S_1 makes only one oracle query and its running time is τ. Also the leakage communicated by S_0, namely (ek, d), has length σ. This shows that $S \in \mathcal{S}^{\text{prl}}_{\tau,\sigma,q}$, concluding the proof. ▌

An interesting open question is whether our $\mathsf{EwH}[\mathsf{H}, \mathsf{RE}]$ scheme can also be shown to meet the notions of security for D-PKE with respect to auxiliary inputs from [43], or, more generally, whether UCE allows one to achieve these goals in the standard model. (Here we refer to full auxiliary-input security rather than such security for block sources. The latter is already achieved without ROs in [43].)

D-PKE secure for adaptively-chosen plaintext distributions was considered in [102], who gave ROM solutions. It would be interesting to see if the RO here can be instantiated to obtain standard model schemes. The difficulty in doing this directly with UCE is that in the latter, the points being hashed may not depend on the key.

5.4 Message-locked encryption

Message-locked encryption (MLE) [19] is a form of symmetric encryption in which the key is derived from the message. It allows secure data deduplication. The convergent encryption (CE) MLE scheme of [63, 19] is in use by numerous providers of cloud storage. Its security is justified in the ROM by [19]. Here we instantiate the RO with a UCE family to get standard-model security.

DEFINITIONS. An MLE scheme MLE [19] specifies the following PT algorithms. Via $p \leftarrow\!\!\$\ \mathsf{MLE.Pg}(1^\lambda)$ one generates parameters. Via $k \leftarrow \mathsf{MLE.Kg}(1^\lambda, p, m)$, one deterministically derives a key k from a message $m \in \{0,1\}^{\mathsf{MLE.il}(\lambda)}$. Via $c \leftarrow \mathsf{MLE.Enc}(1^\lambda, p, k, m)$ one encrypts m under k to get ciphertext c. Via $m \leftarrow \mathsf{MLE.Dec}(1^\lambda, p, k, c)$ one deterministically decrypts c under k to get $m \in \{0,1\}^{\mathsf{MLE.il}(\lambda)} \cup \{\perp\}$. Via $t \leftarrow \mathsf{MLE.Tag}(1^\lambda, p, c)$ one deterministically generates a tag t for ciphertext c. Correctness requires the following for all $\lambda \in \mathbb{N}$, all $m \in \{0,1\}^{\mathsf{MLE.il}(\lambda)}$, all $p \in [\mathsf{MLE.Pg}(1^\lambda)]$ and all $k_1, k_2 \in [\mathsf{MLE.Kg}(1^\lambda, p, m)]$: (1) $\mathsf{MLE.Tag}(1^\lambda, p, c_1) = \mathsf{MLE.Tag}(1^\lambda, p, c_2)$ for all $c_1 \in [\mathsf{MLE.Enc}(1^\lambda, p, k_1, m)]$ and all $c_2 \in [\mathsf{MLE.Enc}(1^\lambda, p, k_2, m)]$, and (2) $\mathsf{MLE.Dec}(1^\lambda, p, k_2, c) = m$ for all $c \in [\mathsf{MLE.Enc}(1^\lambda, p, k_1, m)]$. The $\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\lambda)$ game defined in Fig. 12 is a simplification of the one of [19], without side-information. A IND\$-CDA adversary

$A = (A_1, A_2)$ is a pair of PT algorithms, where A_1 on input 1^λ returns $\mathbf{m}$, a $v(\lambda)$-vector of distinct $\mathsf{MLE.il}(\lambda)$-bit strings, where v depends on A. The guessing probability Guess_A of A is the function that on input $\lambda \in \mathbb{N}$ returns the maximum, over all i, m, of $\Pr[\mathbf{m}[i] = m]$, the probability over $\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$. We say that A has *high min-entropy* if $\mathrm{Guess}_A(\cdot)$ is negligible. We let $\mathsf{Adv}^{\mathsf{ind\$\text{-}cda}}_{\mathsf{MLE},A}(\lambda) = 2\Pr[\text{IND\$-CDA}^A_{\mathsf{MLE}}(\lambda)] - 1$ and say that MLE is IND\$-CDA-secure if $\mathsf{Adv}^{\mathsf{ind\$\text{-}cda}}_{\mathsf{MLE},A}(\cdot)$ is negligible for all PT A that have high min-entropy. We let $\mathsf{IND\$\text{-}CDA}$ be the set of all IND\$-CDA-secure MLE schemes.

A symmetric encryption (SE) scheme SE will be a tool in the construction. Such a scheme specifies the following PT algorithms. Via $k \leftarrow\!\!\$\ \mathsf{SE.Kg}(1^\lambda)$ one generates a key that is a random $\mathsf{SE.kl}(\lambda)$ bit string. Via $c \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, k, m)$, one encrypts message $m \in \{0,1\}^{\mathsf{SE.il}(\lambda)}$ under k to get a ciphertext $c \in \{0,1\}^{\mathsf{SE.cl}(\lambda)}$, where $\mathsf{SE.cl}\colon \mathbb{N} \to \mathbb{N}$ is the ciphertext-length function of SE. Via $m \leftarrow \mathsf{SE.Dec}(1^\lambda, k, c)$ one deterministically decrypts c under k to get back $m \in \{0,1\}^{\mathsf{SE.il}(\lambda)} \cup \{\perp\}$. Correctness requires that $\mathsf{SE.Dec}(1^\lambda, k, \mathsf{SE.Enc}(1^\lambda, k, m)) = m$ with probability 1 for all $m \in \{0,1\}^{\mathsf{SE.il}(\lambda)}$, all $k \in [\mathsf{SE.Kg}(1^\lambda)]$ and all $\lambda \in \mathbb{N}$, the probability being over the coins of $\mathsf{SE.Enc}$. We say that SE is a D-SE (deterministic SE) scheme if $\mathsf{SE.Enc}$ is deterministic. We will assume a D-SE scheme meeting the following definition of security. Game $\mathrm{ROR}^A_{\mathsf{SE}}(\lambda)$ starts by picking $b \leftarrow\!\!\$\ \{0,1\}$. Adversary A is then given access to an oracle Enc that, on input $m \in \{0,1\}^{\mathsf{SE.il}(\lambda)}$, picks a fresh key $k \leftarrow\!\!\$\ \mathsf{SE.Kg}(1^\lambda)$ and computes $c \leftarrow \mathcal{E}(1^\lambda, k, m)$. If $b = 1$ then it returns c, otherwise it returns $c' \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.cl}(\lambda)}$. When the adversary exits with output b', the game returns $(b' = b)$. We say that SE is ROR-secure if $\mathsf{Adv}^{\mathsf{ror}}_{\mathsf{SE},A}(\cdot)$ is negligible for all PT A, where $\mathsf{Adv}^{\mathsf{ror}}_{\mathsf{SE},A}(\lambda) = 2\Pr[\mathrm{ROR}^A_{\mathsf{SE}}(\lambda)] - 1$. We let ROR denote the set of all ROR-secure SE schemes.

Results. Let SE be a D-SE scheme and let H be a family of functions with input length $\mathsf{H.il} = \mathsf{SE.il}$ and output length $\mathsf{H.ol} = \mathsf{SE.kl}$. We describe a standard model instantiation $\mathsf{MLE} = \mathsf{CE}[\mathsf{H}, \mathsf{SE}]$ of the convergent encryption scheme of [63, 19] in Fig. 12 which has input (message) length $\mathsf{MLE.il} = \mathsf{SE.il}$. Correctness is easy to verify. MLE schemes also have an additional security requirement called tag consistency [19], and $\mathsf{CE}[\mathsf{H}, \mathsf{SE}]$ as described here has perfect tag consistency. The theorem below shows that $\mathsf{CE}[\mathsf{H}, \mathsf{SE}]$ is IND\$-CDA-secure if SE is ROR-secure and H is $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}]$-secure. By $\mathsf{T_{SE.Enc}}\colon \mathbb{N} \to \mathbb{N}$ we denote the running time of $\mathsf{SE.Enc}$, meaning $\mathsf{SE.Enc}(1^\lambda, k, m)$ halts in $\mathsf{T_{SE.Enc}}(\lambda)$ steps with probability one for all $\lambda \in \mathbb{N}$, all $k \in \{0,1\}^{\mathsf{SE.kl}(\lambda)}$ and all $m \in \{0,1\}^{\mathsf{SE.il}(\lambda)}$.

Theorem 5.4 Suppose $\mathsf{SE} \in \mathsf{ROR}$. Suppose $\mathsf{H.il} = \mathsf{SE.il}$ and $\mathsf{H.ol} = \mathsf{SE.kl}$. Let $\tau = \mathcal{O}(\mathsf{T_{SE.Enc}} + \mathsf{SE.kl} + \mathsf{SE.il})$ and let $q = 1$. Let $\sigma = 0$. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}]$, then $\mathsf{CE}[\mathsf{H}, \mathsf{SE}] \in \mathsf{IND\$\text{-}CDA}$.

Proof of Theorem 5.4: Let $\mathsf{MLE} = \mathsf{CE}[\mathsf{H}, \mathsf{SE}]$. Given a PT high min-entropy adversary $A = (A_1, A_2)$ for game $\text{IND\$-CDA}^A_{\mathsf{MLE}}(\lambda)$, we build a source $S \in \mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}$, a distinguisher $D \in \mathcal{D}^{\mathsf{poly}}$, and a PT adversary B_1 for game $\mathrm{ROR}^A_{\mathsf{SE}}(\lambda)$ such that

$$\mathsf{Adv}^{\mathsf{ind\$\text{-}cda}}_{\mathsf{MLE},A}(\cdot) \le \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + \mathsf{Adv}^{\mathsf{ror}}_{\mathsf{SE},B_1}(\cdot). \tag{8}$$

The theorem follows from the assumptions $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}]$ and $\mathsf{SE} \in \mathsf{ROR}$. The constructions of S, D, B_1 are shown below, where v is the function associated to A as per the definitions:

$\underline{S^{\mathrm{Hash}}(1^\lambda)}$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$
For $i = 1$ to $v(\lambda)$ do
  $\mathbf{k}[i] \leftarrow \mathrm{Hash}(\mathbf{m}[i], 1^{\mathsf{SE.kl}(\lambda)})$
  $\mathbf{c}[i] \leftarrow \mathsf{SE.Enc}(1^\lambda, \mathbf{k}[i], \mathbf{m}[i])$
$L \leftarrow (\varepsilon, \mathbf{c})$
Return L

$\underline{D(1^\lambda, L)}$
$(\varepsilon, \mathbf{c}[1], \ldots, \mathbf{c}[v(\lambda)]) \leftarrow L$
$a' \leftarrow A_2(1^\lambda, hk, \mathbf{c})$
Return a'

$\underline{B_1^{\mathrm{Enc}}(1^\lambda)}$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$
For $i = 1$ to $v(\lambda)$ do
  $\mathbf{c}[i] \leftarrow \mathrm{Enc}(\mathbf{m}[i])$
$d' \leftarrow\!\!\$\ A_2(1^\lambda, hk, \mathbf{c})$
Return d'

$\underline{B_2^{\mathrm{Enc}}(1^\lambda)}$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$
$Q \leftarrow \{\mathbf{m}[1], \ldots, \mathbf{m}[|\mathbf{m}|]\}$
For $i = 1$ to $v(\lambda)$ do
  $\mathbf{c}[i] \leftarrow \mathrm{Enc}(\mathbf{m}[i])$
$m \leftarrow\!\!\$\ P'(1^\lambda, (\varepsilon, \mathbf{c}))$
If $m \in Q$ then $b' \leftarrow 1$
Else $b' \leftarrow 0$
Return b'

Let a, c, d be the challenge bits of games $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot)$, $\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\cdot)$ and $\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot)$ respectively. Then

$$\begin{aligned}
\Pr[\,\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\cdot) \,|\, c=1\,] &= \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a=1\,] \\
\Pr[\,\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\cdot) \,|\, c=0\,] &= \Pr[\,\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot) \,|\, d=0\,] \\
\Pr[\,\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot) \,|\, d=1\,] &= 1 - \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a=0\,]\,. \qquad (9)
\end{aligned}$$

So

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ind\$\text{-}cda}}_{\mathsf{MLE},A}(\cdot) &= \Pr[\,\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\cdot) \,|\, c=1\,] + \Pr[\,\text{IND\$-CDA}^{A}_{\mathsf{MLE}}(\cdot) \,|\, c=0\,] - 1 \\
&= \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a=1\,] + \Pr[\,\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot) \,|\, d=0\,] - 1 \\
&= \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a=1\,] + \Big(\Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a=0\,] - 1 + \Pr[\,\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot) \,|\, d=1\,]\Big) \\
&\qquad + \Pr[\,\mathrm{ROR}^{B_1}_{\mathsf{SE}}(\cdot) \,|\, d=0\,] - 1 \\
&= \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + \mathsf{Adv}^{\mathsf{ror}}_{\mathsf{SE},B_1}(\cdot)
\end{aligned}$$

which yields Equation (8). The third equality above is justified by the fact that, by Equation (9), the term in parentheses is zero.

We now show that S is computationally unpredictable. By Lemma 4.3 it suffices to show simple computational unpredictability. Let P' be a simple computational predictor. Consider the PT adversary B_2 described above. Letting b denote the challenge bit in game $\mathrm{ROR}^{B_2}_{\mathsf{SE}}(\lambda)$, we have

$$\Pr[\,b'=1 \,|\, b=1\,] = \mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \qquad \text{and} \qquad \Pr[\,b'=1 \,|\, b=0\,] \le v \cdot \mathrm{Guess}_A(\cdot)\,.$$

Subtracting, we have $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le \mathsf{Adv}^{\mathsf{ror}}_{\mathsf{SE},B_2}(\cdot) + v \cdot \mathrm{Guess}_A(\cdot)$. The simple unpredictability of S then follows from the assumption that $\mathsf{SE} \in \mathsf{ROR}$ and that A has high min-entropy.

Finally, we exhibit, below, PT algorithms S_0, S_1 such that $S = \mathsf{Prl}[S_0, S_1]$—

$\underline{S_0(1^\lambda)}$	$\underline{S_1^{\mathrm{HASH}}(1^\lambda, m)}$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$	$k \leftarrow\!\!\$\ \mathrm{HASH}(m, 1^{\mathsf{SE.kl}(\lambda)})$
Return $(\varepsilon, \mathbf{m})$	$c \leftarrow \mathsf{SE.Enc}(1^\lambda, k, m)$
	Return c

Note that S_1 makes only one oracle query and its running time is τ. Also the leakage communicated by S_0 is ε, namely has length σ. This shows that $S \in \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}$, concluding the proof. ▌

The scheme above does not meet the MLE security definitions of [1] which allow the messages to depend on the public parameter, since the latter is the key for our UCE family. We remark that an IND\$-CDA-secure CE scheme can also be obtained under the $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$ assumption, by instantiating SE via H. We omit the details.

5.5 Point-function obfuscation

A *point function* has non-$\perp$ output on just one point. Lynn, Prabhakaran, and Sahai [95] showed how to obfuscate point functions in the ROM. We mUCE-instantiate their construction to obtain a standard-model point-function obfuscation scheme.

Definitions. For $(\alpha, \beta) \in \{0,1\}^* \times \{0,1\}^*$ we let $\Delta_{\alpha,\beta}\colon \{0,1\}^* \to \{\beta, \perp\}$ denote the function that on input $x \in \{0,1\}^*$ returns β if $x = \alpha$ and $\perp$ otherwise. A *point-function obfuscator* OS is defined as follows. Via $F \leftarrow\!\!\$\ \mathsf{OS.Obf}(1^\lambda, (\alpha, \beta))$, PT obfuscation algorithm $\mathsf{OS.Obf}$ creates a description F of an obfuscated version of $\Delta_{\alpha,\beta}$. Via $y \leftarrow\!\!\$\ \mathsf{OS.Ev}(1^\lambda, F, x)$, deterministic PT algorithm $\mathsf{OS.Ev}$ evaluates the obfuscated program F at $x \in \{0,1\}^*$ to get output y. Correctness requires that $\mathsf{OS.Ev}(1^\lambda, \mathsf{OS.Obf}(1^\lambda, (\alpha, \beta)), \alpha) = \beta$

$\underline{\textsc{Main}\ \mathrm{PFOB}^{A,T}_{\mathsf{OS}}(\lambda)}$	$\underline{\mathsf{OS.Obf}(1^\lambda,(\alpha,\beta))}$
$b \leftarrow\!\!\$\ \{0,1\}$; $(\boldsymbol{\alpha},\boldsymbol{\beta}) \leftarrow\!\!\$\ A_1(1^\lambda)$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
For $i = 1$ to $\lvert\boldsymbol{\alpha}\rvert$ do $\mathbf{F}[i] \leftarrow\!\!\$\ \mathsf{OS.Obf}(1^\lambda,(\boldsymbol{\alpha}[i],\boldsymbol{\beta}[i]))$	$\overline{\alpha} \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 0 \,\|\, \alpha, 1^\lambda)$
If $b = 1$ then $w \leftarrow\!\!\$\ A_2(1^\lambda,\mathbf{F})$ else $w \leftarrow\!\!\$\ T^{\textsc{Prog}}(1^\lambda,\lvert\boldsymbol{\alpha}\rvert)$	$\overline{\beta} \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 1 \,\|\, \alpha, 1^{\lvert\beta\rvert}) \oplus \beta$
$b' \leftarrow\!\!\$\ A_3(1^\lambda, w)$	Return $(hk, \overline{\alpha}, \overline{\beta})$
Return $(b = b')$	
$\underline{\textsc{Prog}(i,x)}$	$\underline{\mathsf{OS.Ev}(1^\lambda,(hk,\overline{\alpha},\overline{\beta}),x)}$
Return $\Delta_{\boldsymbol{\alpha}[i],\boldsymbol{\beta}[i]}(x)$	$\alpha^* \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 0 \,\|\, x, 1^\lambda)$
	If $(\alpha^* \neq \overline{\alpha})$ then return $\perp$
	Else return $\overline{\beta} \oplus \mathsf{H.Ev}(1^\lambda, hk, 1 \,\|\, x, 1^{\lvert\overline{\beta}\rvert})$

Figure 13: **Left:** The PFOB game defining security of point-function obfuscator OS. **Right:** Point-function obfuscation scheme $\mathsf{OS} = \mathsf{HTC}[\mathsf{H}]$.

for all $\alpha, \beta \in \{0,1\}^*$ and all $\lambda \in \mathbb{N}$. Security is defined via game $\mathrm{PFOB}^{A_1,A_2}_{\mathsf{OS}}(\lambda)$ of Fig. 13. It involves an adversary $A = (A_1, A_2, A_3)$ and a simulator T. Adversary A_1 outputs a pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of vectors of the same length, entries of both being strings, thereby describing a sequence of point functions. It is required that there is a function ℓ, called the function output-length of A, such that all entries of $\boldsymbol{\beta}$ have length $\ell(\lambda)$, and it is required that all entries of $\boldsymbol{\alpha}$ are distinct. We let $\mathsf{Adv}^{\mathsf{obf}}_{\mathsf{OS},A,T}(\lambda) = 2\Pr[\mathrm{PFOB}^{A,T}_{\mathsf{OS}}(\lambda)] - 1$. The guessing probability Guess_A of A is the function that on input $\lambda \in \mathbb{N}$ returns the maximum, over all i, α, of $\Pr[\boldsymbol{\alpha}[i] = \alpha]$, the probability over $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leftarrow\!\!\$\ A_1(1^\lambda)$. We say that A has *high min-entropy* if $\mathrm{Guess}_A(\cdot)$ is negligible. We say that OS is a secure point-function obfuscator if for all PT high min-entropy A there is a PT simulator T such that $\mathsf{Adv}^{\mathsf{obf}}_{\mathsf{OS},A,T}(\cdot)$ is negligible. Let PFOB denote the set of all secure point-function obfuscators. The high min-entropy condition makes the problem "interesting" in that without it the adversary knows α and thus there is nothing to gain by obfuscation. This definition is from [95, 51], adapted to our notation.

Results. Let H be a family of functions with $\mathsf{H.IL} = \mathsf{H.OL} = \mathbb{N}$. Our Hash-then-Compare point-obfuscation scheme $\mathsf{OS} = \mathsf{HTC}[\mathsf{H}]$ is described in Fig. 13. The following says that multi-key UCE security of H suffices for OS to be secure:

Theorem 5.5 If $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$ then $\mathsf{HTC}[\mathsf{H}] \in \mathsf{PFOB}$.

Proof of Theorem 5.5: Let $A = (A_1, A_2, A_3)$ be a PT adversary and let ℓ be its function output-length. We'll construct a PT statistically unpredictable multi-source S, a PT distinguisher D, and a PT simulator T such that

$$\mathsf{Adv}^{\mathsf{obf}}_{\mathsf{OS},A,T}(\cdot) = \mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\cdot) \ . \tag{10}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$. The constructions of S, D, and T are shown below.

$\underline{T^{\textsc{Prog}}(1^\lambda, a)}$	$\underline{S^{\textsc{Hash}}(1^\lambda)}$	$\underline{D(1^\lambda, \mathbf{hk}, L)}$
For $i = 1$ to a do	$(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leftarrow\!\!\$\ A_1(1^\lambda)$	$(\overline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\beta}}) \leftarrow L$
$\quad \mathbf{hk}[i] \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	For $i = 1$ to $\lvert\boldsymbol{\alpha}\rvert$ do	For $i = 1$ to $\lvert\overline{\boldsymbol{\alpha}}\rvert$ do
$\quad \overline{\boldsymbol{\alpha}}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$	$\quad \overline{\boldsymbol{\alpha}}[i] \leftarrow\!\!\$\ \textsc{Hash}(0 \Vert \boldsymbol{\alpha}[i], 1^\lambda, i)$	$\quad \mathbf{F}[i] \leftarrow (\mathbf{hk}[i], \overline{\boldsymbol{\alpha}}[i], \overline{\boldsymbol{\beta}}[i])$
$\quad \overline{\boldsymbol{\beta}}[i] \leftarrow\!\!\$\ \{0,1\}^{\ell(\lambda)}$	$\quad \overline{\boldsymbol{\beta}}[i] \leftarrow\!\!\$\ \boldsymbol{\beta}[i] \oplus \textsc{Hash}(1 \Vert \boldsymbol{\alpha}[i], 1^{\lvert\boldsymbol{\beta}[i]\rvert}, i)$	$w \leftarrow\!\!\$\ A_2(1^\lambda, \mathbf{F})$
$\quad \mathbf{F}[i] \leftarrow (\mathbf{hk}[i], \overline{\boldsymbol{\alpha}}[i], \overline{\boldsymbol{\beta}}[i])$	$L \leftarrow (\overline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\beta}})$	$b' \leftarrow\!\!\$\ A_3(1^\lambda, w)$
$w \leftarrow\!\!\$\ A_2(1^\lambda, \mathbf{F})$	Return L	Return b'
Return w		

Note that T does not call its oracle, the latter being in fact unnecessary to prove security. Let b and c be

$\underline{\text{MAIN KDM}^A_{\mathsf{SE}}(\lambda)}$	$\underline{\text{MAIN RKA}^A_{\mathsf{SE}}(\lambda)}$	$\underline{\mathsf{SE.Enc}(1^\lambda, k, m)}$
$(1^n, t) \leftarrow\!\!\$\ A_1(1^\lambda, \varepsilon)$	$(\mathbf{m}_0, \mathbf{m}_1, t) \leftarrow\!\!\$\ A_1(1^\lambda, \varepsilon)$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
For $i = 1$ to n do $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$	$k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$	$h \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k, 1^{\mathsf{H.ol}(\lambda)})$
$(\mathbf{s}, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda, (t, \mathbf{k}))$	$\mathbf{k} \leftarrow\!\!\$\ A_1(1^\lambda, (t, k))$; $b \leftarrow\!\!\$\ \{0,1\}$	$c \leftarrow (hk, h \oplus m)$; Return c
$b \leftarrow\!\!\$\ \{0,1\}$	For $i = 1$ to $\lvert\mathbf{m}_b\rvert$ do	$\underline{\mathsf{SE.Dec}(1^\lambda, k, (hk, z))}$
For $i = 1$ to $\lvert\mathbf{m}_b\rvert$ do	$\quad \mathbf{c}[i] \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, \mathbf{k}[i], \mathbf{m}_b[i])$	$h \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k, 1^{\lvert z\rvert})$
$\quad \mathbf{c}[i] \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, \mathbf{k}[\mathbf{s}[i]], \mathbf{m}_b[i])$	$b' \leftarrow\!\!\$\ A_2(1^\lambda, t, \mathbf{c})$; Return $(b = b')$	$m \leftarrow\!\!\$\ h \oplus z$; Return m
$b' \leftarrow\!\!\$\ A_2(1^\lambda, t, \mathbf{c})$; Return $(b = b')$		

Figure 14: **Left:** The KDM game. **Middle:** The RKA game. **Right:** The SE scheme $\mathsf{SE} = \mathsf{HtX}[\mathsf{H}]$.

the challenge bits of games $\text{mUCE}^{S,D}_{\mathsf{H}}(\lambda)$ and $\text{PFOB}^{A,T}_{\mathsf{OS}}(\lambda)$ respectively. Then

$$\begin{aligned}\Pr[\text{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \mid b = 1] &= \Pr[\text{PFOB}^{A,T}_{\mathsf{OS}}(\cdot) \mid c = 1] \\ \Pr[\text{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \mid b = 0] &= \Pr[\text{PFOB}^{A,T}_{\mathsf{OS}}(\cdot) \mid c = 0]\end{aligned}$$

Summing yields Equation (10). We now show that S is statistically unpredictable. By Lemma 4.7 it suffices to show that S is simple statistically unpredictable. Let P' be a simple predictor. Let v be a polynomial such that $|\boldsymbol{\alpha}| \le v(\lambda)$ in game $\text{PFOB}^{A,T}_{\mathsf{OS}}(\lambda)$, for all $\lambda \in \mathbb{N}$. Then $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le v \cdot \text{Guess}_A(\cdot)$, so the high min-entropy assumption on A implies that S is simple statistically unpredictable. ▌

5.6 Security for key-dependent messages

Black, Rogaway, and Shrimpton (BRS) [27] formalized security in the presence of key-dependent messages (KDM) and described a simple and efficient KDM-secure symmetric encryption scheme in the ROM. We now instantiate the RO in the BRS scheme with a mUCE family and obtain an efficient KDM-secure symmetric encryption scheme in the standard model. There are several other standard-model KDM-secure encryption schemes [41, 5, 10, 96, 4] but they are significantly more complex and less efficient than our instantiated BRS scheme.

DEFINITIONS. Let SE be a symmetric encryption (SE) scheme as defined in Section 5.4. In game $\text{KDM}^A_{\mathsf{SE}}(\lambda)$ of Fig. 14, an adversary $A = (A_1, A_2)$ is a pair of algorithms. Algorithm A_1, when invoked with $(1^\lambda, \varepsilon)$, returns $(1^n, t)$ where n is the number of keys it is requesting be created, and t is state information. Then when invoked with $(1^\lambda, (t, \mathbf{k}))$ where $\mathbf{k} \in (\{0,1\}^{\mathsf{SE.kl}(\lambda)})^n$ is a vector of keys, it outputs a triple of vectors $\mathbf{s}, \mathbf{m}_0, \mathbf{m}_1$ satisfying the following: (1) $|\mathbf{s}| = |\mathbf{m}_0| = |\mathbf{m}_1|$, and (2) $\mathbf{s}[i] \in [1, n]$ and $\mathbf{m}_0[i], \mathbf{m}_1[i] \in \mathsf{SE.il}(\lambda)$ for all $i \in [1, |\mathbf{s}|]$. We say that SE is KDM-secure if $\mathsf{Adv}^{\mathsf{kdm}}_{\mathsf{SE},A}(\cdot)$ is negligible for every PT KDM adversary A, where $\mathsf{Adv}^{\mathsf{kdm}}_{\mathsf{SE},A}(\lambda) = 2\Pr[\text{KDM}^A_{\mathsf{SE}}(\lambda)] - 1$. We let KDM denote the set of all KDM-secure schemes. Our definitions capture non-adaptive security, but this includes the cases that have been most prominent in past work, namely key cycles and cliques [41, 5, 2, 44].

RESULTS. BRS [27] showed that encrypting a message m under key k by picking a random r and returning $(r, \mathsf{RO}(r \,\|\, k) \oplus m)$ is KDM secure when RO is a random oracle. The natural first attempt to instantiate via a family H would be to add $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$ to the encryption key and then replace RO with $\mathsf{H.Ev}(1^\lambda, hk, \cdot, 1^{\mathsf{H.ol}(\lambda)})$, but this fails because in the KDM setting the messages are chosen by A_1 as a function of the encryption key(s), and UCE-security will not apply if the messages depend on hk. Instead, we leave the key unchanged relative to the BRS scheme and view the random value r of the BRS scheme as a key for H, so that a fresh key hk is chosen for each encryption. Given H with $\lambda \in \mathsf{H.IL}(\lambda)$ for all $\lambda \in \mathbb{N}$, our instantiated transform produces the SE scheme $\mathsf{SE} = \mathsf{HtX}[\mathsf{H}]$ whose encryption and decryption algorithms are described in Fig. 14. (Here "HtX" stands for "Hash-then-XOR.") Its key length is defined by $\mathsf{SE.kl}(\lambda) = \lambda$ for all $\lambda \in \mathbb{N}$ and its input length is $\mathsf{SE.il} = \mathsf{H.ol}$. The following theorem says that $\mathsf{HtX}[\mathsf{H}]$ is KDM secure if H is $\mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$-secure.

Theorem 5.6 If $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$, then $\mathsf{HtX}[\mathsf{H}] \in \mathsf{KDM}$.

Proof of Theorem 5.6: Let $\mathsf{SE} = \mathsf{HtX}[\mathsf{H}]$. Let $A = (A_1, A_2)$ be a PT KDM adversary. Assume that A_1 outputs messages of length $\mathsf{H.ol}(\lambda)$. We will construct a PT statistically unpredictable multi-source S and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathsf{kdm}}_{\mathsf{SE},A}(\cdot) \le 2 \cdot \mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\cdot) \ . \tag{11}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$. Let q and $\overline{n}$ be polynomials such that, in game $\mathrm{KDM}^A_{\mathsf{SE}}(\lambda)$, we have $|\mathbf{m}_0| \le q(\lambda)$ and $n \le \overline{n}(\lambda)$ for all $\lambda \in \mathbb{N}$. The constructions of S and D are shown below:

$\underline{S^{\text{HASH}}(1^\lambda, t)}$	$\underline{D(1^\lambda, \mathbf{hk}, L)}$
If $t = \varepsilon$ then	$(\mathbf{c}', t', d) \leftarrow L$
$\quad (1^n, t') \leftarrow\!\!\$\ A_1(1^\lambda, \varepsilon)$; Return $(1^{q(\lambda)}, (1^n, t'))$	For $i = 1$ to $\lvert\mathbf{c}'\rvert$ do
Else	$\quad \mathbf{c}[i] \leftarrow (\mathbf{hk}[i], \mathbf{c}'[i])$
$\quad (1^n, t') \leftarrow t$; $d \leftarrow\!\!\$\ \{0,1\}$	$d' \leftarrow\!\!\$\ A_2(1^\lambda, t', \mathbf{c})$
$\quad$ For $i = 1$ to n do $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$	If $(d = d')$ then $b' \leftarrow 1$ else $b' \leftarrow 0$
$\quad (\mathbf{s}, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A_1(1^\lambda, (t', \mathbf{k}))$	Return b'
$\quad$ For $i = 1$ to $\lvert\mathbf{m}_d\rvert$ do $\mathbf{c}'[i] \leftarrow \text{HASH}(\mathbf{k}[\mathbf{s}[i]], 1^{\mathsf{H.ol}(\lambda)}, i) \oplus \mathbf{m}_d[i]$	
$\quad L \leftarrow (\mathbf{c}', t', d)$; Return L	

Let b denote the challenge bit in game $\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\cdot)$. Then

$$\Pr[\,\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 1\,] = \Pr[\mathrm{KDM}^A_{\mathsf{SE}}(\cdot)]$$
$$\Pr[\,\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 0\,] = \frac{1}{2} \ .$$

Summing yields Equation (11). It remains to show that S is statistically unpredictable. By Lemma 4.7, it suffices to show that S is simple statistically unpredictable. Consider an arbitrary simple predictor P'. Given $|\mathbf{k}|$ and the leakage $(\mathbf{c}', t', d)$, the components of $\mathbf{k}$ are still uniformly and independently distributed λ-bit strings. Hence $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{S,P'}(\lambda) \le \overline{n}(\lambda)/2^\lambda$ for every $\lambda \in \mathbb{N}$. ▌

5.7 Security against related-key attack

Symmetric encryption schemes secure against related-key attack (RKA) must preserve security even when encryption is performed under keys $k' = \phi(k)$ derived from the original key by application of a key-deriving function ϕ [21, 6]. Previous schemes [6, 22] provided security for algebraic key-deriving functions ϕ 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. (If the output can be predicted, an adversary can guess the key k' and decrypt.) Furthermore, in our scheme, keys are binary strings rather than group elements, so we cover the most common practical transforms, such as XORing a constant to the key. The scheme itself is in fact the same $\mathsf{HtX}[\mathsf{H}]$ scheme that we showed KDM secure in Section 5.6 and is thus quite simple and natural. We continue to assume only a $\mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$-secure family of functions.

DEFINITIONS. Let SE be a symmetric encryption (SE) scheme as defined in Section 5.4. In game $\mathrm{RKA}^A_{\mathsf{SE}}(\lambda)$ of Fig. 14, an adversary $A = (A_1, A_2)$ is a pair of PT algorithms. Algorithm A_1, when invoked with $(1^\lambda, \varepsilon)$, returns a vectors $\mathbf{m}_0, \mathbf{m}_1$ of messages, along with state information t. It is required that $|\mathbf{m}_0| = |\mathbf{m}_1|$ and that $\mathbf{m}_0[i], \mathbf{m}_1[i] \in \{0,1\}^{\mathsf{SE.il}(\lambda)}$ for all $i \in [1, |\mathbf{m}_0|]$. Then, when invoked with $(1^\lambda, (t, k))$, it produces a vector $\mathbf{k} \in (\{0,1\}^{\mathsf{SE.kl}(\lambda)})^{|\mathbf{m}_0|}$, the entries of this vector being the derived, or related keys. RKA-security is not achievable if A_1 can produce arbitrary keys, as shown by impossibility results in [20], so, following the latter, one usually parametrizes security via a class Φ of transforms that the adversary is allowed to apply to the base key to obtain the related keys, and restricts this class appropriately to obtain results.

We will not take this Φ-parametrized approach because we can achieve security for key-deriving functions that are arbitrary subject only to the necessary condition of being unpredictable. Define the guessing probability Guess_A of A as the function that on input $\lambda \in \mathbb{N}$ returns the maximum, over all k', i and $(\mathbf{m}_0, \mathbf{m}_1, t) \in [A_1(1^\lambda, \varepsilon)]$, of $\Pr[\mathbf{k}[i] = k']$, the probability being over $k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$; $\mathbf{k} \leftarrow\!\!\$\ A_1(1^\lambda, t, k)$. We say that A has high min-entropy if $\text{Guess}_A(\cdot)$ is negligible. We say that SE is RKA-secure if $\mathsf{Adv}^{\mathsf{rka}}_{\mathsf{SE},A}(\cdot)$ is negligible for all PT A that have high min-entropy, where $\mathsf{Adv}^{\mathsf{rka}}_{\mathsf{SE},A}(\lambda) = 2\Pr[\text{RKA}^A_{\mathsf{SE}}(\lambda)] - 1$. We let RKA denote the set of all RKA-secure symmetric encryption schemes.

Results. Let H be a family of functions with $\lambda \in \mathsf{H.IL}(\lambda)$ for all $\lambda \in \mathbb{N}$ and with output length $\mathsf{H.ol}$. The following theorem states that the SE scheme $\mathsf{SE} = \mathsf{HtX}[\mathsf{H}]$, defined in Section 5.6 and depicted in Fig. 14, is RKA-secure, assuming only that H is $\mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$-secure. Recall that $\mathsf{SE.il} = \mathsf{H.ol}$.

Theorem 5.7 If $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$, then $\mathsf{HtX}[\mathsf{H}] \in \mathsf{RKA}$.

Proof: Let $\mathsf{SE} = \mathsf{HtX}[\mathsf{H}]$. Let $A = (A_1, A_2)$ be a PT RKA adversary of high min-entropy. We will construct a PT statistically unpredictable multi-source S and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathsf{rka}}_{\mathsf{HtX}[\mathsf{H}],A}(\cdot) \le 2 \cdot \mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\cdot) \ . \tag{12}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$. Let m be a polynomial such that, in game $\text{RKA}^A_{\mathsf{SE}}(\lambda)$, we have $|\mathbf{m}_0| \le m(\lambda)$ for all $\lambda \in \mathbb{N}$. The constructions of S and D are shown below:

$\underline{S^{\text{Hash}}(1^\lambda, t)}$	$\underline{D(1^\lambda, \mathbf{hk}, L)}$
If $t = \varepsilon$ then	$(\mathbf{c}', t', d) \leftarrow L$
$\quad(\mathbf{m}_0, \mathbf{m}_1, t') \leftarrow\!\!\$\ A_1(1^\lambda)$; $n \leftarrow \lvert\mathbf{m}_0\rvert$; Return $(1^n, (\mathbf{m}_0, \mathbf{m}_1, t'))$	For $i = 1$ to $\lvert\mathbf{c}'\rvert$ do
Else	$\quad\mathbf{c}[i] \leftarrow (\mathbf{hk}[i], \mathbf{c}'[i])$
$\quad(\mathbf{m}_0, \mathbf{m}_1, t') \leftarrow t$; $k \leftarrow\!\!\$\ \{0,1\}^\lambda$; $\mathbf{k} \leftarrow\!\!\$\ A_1(1^\lambda, t', k)$; $d \leftarrow\!\!\$\ \{0,1\}$	$d' \leftarrow\!\!\$\ A_2(1^\lambda, t', \mathbf{c})$
$\quad$For $i = 1$ to $\lvert\mathbf{m}_d\rvert$ do $\mathbf{c}'[i] \leftarrow \text{Hash}(\mathbf{k}[i], 1^{\mathsf{H.ol}(\lambda)}, i) \oplus \mathbf{m}_d[i]$	If $(d = d')$ then $b' \leftarrow 1$ else $b' \leftarrow 0$
$\quad L \leftarrow (\mathbf{c}', t', d)$; Return L	Return b'

Let b denote the challenge bit in game $\text{mUCE}^{S,D}_{\mathsf{H}}(\cdot)$. Then

$$\Pr[\,\text{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \mid b = 1\,] = \Pr[\text{RKA}^A_{\mathsf{SE}}(\cdot)]$$
$$\Pr[\,\text{mUCE}^{S,D}_{\mathsf{H}}(\cdot) \mid b = 0\,] = \frac{1}{2} \ .$$

Summing yields Equation (12). It remains to show that S is statistically unpredictable. By Lemma 4.7, it suffices to show that S is simple statistically unpredictable. Consider an arbitrary simple predictor P'. In the $\text{SPred}^{P'}_S(\lambda)$ game, P' receives leakage $(\mathbf{c}', t', d)$, and to win, it must output a component of $\mathbf{k}$ (along with a message length). Note that given $|\mathbf{k}|$, the variables $\mathbf{c}'$ and d are conditionally independent of $\mathbf{k}$ and t'. Since A is of high min-entropy, it follows that $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{S,P'}(\cdot) \le m \cdot \text{Guess}_A(\cdot)$. ▌

5.8 OAEP

OAEP [24] is a ROM transform of a trapdoor permutation to a PKE scheme. If the trapdoor permutation is one-way then the associated OAEP PKE scheme is IND-CPA in the ROM [24]. We would like to instantiate the RO in OAEP in a way that retains this result in the standard model. Here we describe two instantiations of OAEP. The first gets us (non-adaptive) IND-CPA-KI (IND-CPA for messages that do not depend on the public key) assuming the trapdoor permutation is partially one-way, and a hash function family secure w.r.t unpredictable parallel sources. The second requires the trapdoor permutation to be just one-way, but strengthens the requirement on the hash function family to reset-secure parallel sources. We note that for RSA, the most popular choice of trapdoor permutation, one-wayness implies partial one-wayness anyway [66].) Compared to KOS [89], we have relaxed the assumption on the trapdoor permutation from

MAIN IND-CPA-KI$^A_{\mathsf{PKE}}(\lambda)$	$\underline{\mathsf{OAEP.Kg}(1^\lambda)}$	$\underline{\mathsf{OAEP.Dec}(1^\lambda, (dk, hk), c)}$
$b \leftarrow\!\!\$\ \{0,1\}$	$(ek, dk) \leftarrow\!\!\$\ \mathsf{TF.EKg}(1^\lambda)$	$c' \leftarrow \mathsf{TF.Inv}(1^\lambda, dk, c, 1^{\mathsf{TF.ol}(\lambda)})$
$(ek, dk) \leftarrow\!\!\$\ \mathsf{PKE.Kg}(1^\lambda)$	$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$x \| y \leftarrow c'$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda, \varepsilon)$	Return $((ek, hk), (dk, hk))$	$t_2 \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 1\|x, 1^{\ell_3(\lambda)})$
For $i = 1$ to $\lvert\mathbf{m}_b\rvert$ do	$\underline{\mathsf{OAEP.Enc}(1^\lambda, (ek, hk), m)}$	$r \leftarrow y \oplus t_2$
$\quad \mathbf{c}[i] \leftarrow\!\!\$\ \mathsf{PKE.Enc}(1^\lambda, ek, \mathbf{m}_b[i])$	$r \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}$	$t_1 \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 0\|r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$
$b' \leftarrow\!\!\$\ A(1^\lambda, t, \mathbf{c}, ek)$	$t_1 \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 0\|r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$	$m \| z \leftarrow x \oplus t_1$
Return $(b = b')$	$x \leftarrow (m \,\|\, 0^{\ell_2(\lambda)}) \oplus t_1$	If $(z = 0^{\ell_2(\lambda)})$ then return m
	$t_2 \leftarrow \mathsf{H.Ev}(1^\lambda, hk, 1\|x, 1^{\ell_3(\lambda)})$	Else return $\perp$
	$y \leftarrow t_2 \oplus r$	
	$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})$	
	Return c	

Figure 15: **Left:** Game defining (non-adaptive) IND-CPA-KI security of public-key encryption scheme PKE. **Right:** $\mathsf{OAEP}[\mathsf{H}, \mathsf{TF}, \ell_1, \ell_2, \ell_3]$ scheme.

lossiness to plain one-wayness. In the particular case of RSA we have relaxed the assumption from Φ-hiding to standard one-wayness. We note that RSA-OAEP is a widely used and implemented standard.

DEFINITIONS. Let TF be a family of functions with input length TF.il and output length TF.ol. We say that TF is *p-partially one-way*, where p: $\mathbb{N} \to \mathbb{N}$, if $\mathsf{Adv}^{\mathsf{pow}}_{\mathsf{TF},p,I}(\cdot)$ is negligible for all PT I, where $\mathsf{Adv}^{\mathsf{pow}}_{\mathsf{TF},p,I}(\lambda) = \Pr[I(1^\lambda, ek, y) = x[1, p(\lambda)]]$ in the experiment $ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)$; $x \leftarrow\!\!\$\ \{0,1\}^{\mathsf{TF.il}(\lambda)}$; $y \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x, 1^{\mathsf{TF.ol}(\lambda)})$. We let OW_p be the set of all TF that are p-partially one-way.

We say that a PKE scheme PKE is IND-CPA-KI secure if $\mathsf{Adv}^{\mathsf{indcpa\text{-}ki}}_{\mathsf{PKE},A}(\cdot)$ is negligible for all PT adversaries A, where $\mathsf{Adv}^{\mathsf{indcpa\text{-}ki}}_{\mathsf{PKE},A}(\lambda) = 2\Pr[\text{IND-CPA-KI}^A_{\mathsf{PKE}}(\lambda)] - 1$ and game IND-CPA-KI$^A_{\mathsf{PKE}}(\lambda)$ is in Fig. 15. Here, we require that A, when invoked with $1^\lambda, \varepsilon$ outputs $(t, \mathbf{m}_0, \mathbf{m}_1)$ where $\mathbf{m}_0, \mathbf{m}_1$ are vectors of length $q(\lambda)$ and $\mathbf{m}_0[i], \mathbf{m}_1[i] \in \{0,1\}^{\mathsf{PKE.il}(\lambda)}$ for $i \in [1, q(\lambda)]$, where q is a polynomial associated to A. We assume that $|t| = \lambda$. This is wlog because, by using a PRG, we can assume that $A(1^\lambda, \varepsilon)$ uses randomness of length λ, and then we can assume that t consists of exactly this randomness. We let IND-CPA-KI denote the set of IND-CPA-KI-secure PKE schemes.

RESULTS. Let TF be a trapdoor family of functions (as defined in Section 5.2) with TF.il = TF.ol. Let ℓ_1, ℓ_2, ℓ_3: $\mathbb{N} \to \mathbb{N}$ satisfy $\ell_1 + \ell_2 + \ell_3 = \mathsf{TF.il}$. Let H be a family of functions such that $1 + \ell_1(\lambda) + \ell_2(\lambda), 1 + \ell_3(\lambda) \in \mathsf{H.IL}(\lambda)$ and $\ell_3(\lambda), \ell_1(\lambda) + \ell_2(\lambda) \in \mathsf{H.OL}(\lambda)$ for all $\lambda \in \mathbb{N}$. Our instantiated OAEP transform associates to these the PKE scheme $\mathsf{PKE} = \mathsf{OAEP}[\mathsf{H}, \mathsf{TF}, \ell_1, \ell_2, \ell_3]$ whose algorithms are described in Fig. 15. The scheme has message-length function $\mathsf{PKE.il} = \ell_1$. We note that we use the fact that our family H allows variable output lengths, but we need only two different output lengths, allowing a key for the family to be long compared to either output length, which may be important in preventing attacks. The following says that if TF is $(\ell_1 + \ell_2)$-partially one-way and H is UCE-secure then $\mathsf{OAEP}[\mathsf{H}, \mathsf{TF}, \ell_1, \ell_2, \ell_3]$ is IND-CPA-KI secure as long as ℓ_2, ℓ_3 are super-logarithmic.

Theorem 5.8 Let $\mathsf{TF}, \mathsf{H}, \ell_1, \ell_2, \ell_3$ be as above, and let $\mathsf{PKE} = \mathsf{OAEP}[\mathsf{H}, \mathsf{TF}, \ell_1, \ell_2, \ell_3]$. Assume $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible. Let $\tau = \mathcal{O}(\mathsf{T}_{\mathsf{TF.Ev}} + \ell_1 + \ell_2 + \ell_3)$ and let $q' = 2$. Let σ be defined by $\sigma(\lambda) = \lambda + \mathsf{PKE.ekl}(\lambda) + 1$ for all $\lambda \in \mathbb{N}$. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q'}]$ and $\mathsf{TF} \in \mathsf{OW}_{\ell_1+\ell_2}$, then $\mathsf{PKE} \in \mathsf{IND\text{-}CPA\text{-}KI}$.

Proof of Theorem 5.8: Let A be a PT adversary for game IND-CPA-KI$^A_{\mathsf{PKE}}(\lambda)$. Let q be the polynomial associated to A as per the definitions. Let

$$\epsilon(\lambda) = \frac{q(\lambda)^2}{2^{1+\ell_3(\lambda)}} + \frac{q(\lambda)^2}{2^{1+\ell_1(\lambda)+\ell_2(\lambda)}}$$

for all $\lambda \in \mathbb{N}$. We'll construct a source $S \in \mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q'}$ and a distinguisher $D \in \mathcal{D}^{\mathsf{poly}}$ such that

$$\mathsf{Adv}^{\mathsf{indcpa\text{-}ki}}_{\mathsf{PKE},A}(\cdot) \le 2\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) + 2\epsilon \ . \tag{13}$$

The theorem follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q'}]$ and that $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible. The constructions of S and D are shown below:

$\underline{S^{\textsc{Hash}}(1^\lambda)}$
$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)\ ;\ d \leftarrow\!\!\$\ \{0,1\}$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda, \varepsilon)$
For $i = 1$ to $|\mathbf{m}_d|$ do
  $r \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ t_1 \leftarrow \textsc{Hash}(0\|r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$
  $x \leftarrow (\mathbf{m}_d[i] \,\|\, 0^{\ell_2(\lambda)}) \oplus t_1\ ;\ t_2 \leftarrow \textsc{Hash}(1\|x, 1^{\ell_3(\lambda)})$
  $y \leftarrow t_2 \oplus r\ ;\ \mathbf{c}[i] \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})$
$L \leftarrow ((ek, t, d), \mathbf{c})$
Return L

$\underline{D(1^\lambda, hk, L)}$
$((ek, t, d), \mathbf{c}) \leftarrow L$
$d' \leftarrow\!\!\$\ A(1^\lambda, t, \mathbf{c}, (ek, hk))$
If $(d = d')$ then $b' \leftarrow 1$ else $b' \leftarrow 0$
Return b'

Let b denote the challenge bit in game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\lambda)$. We claim that

$$\Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 1\,] = \frac{1}{2} + \frac{1}{2}\mathsf{Adv}^{\mathsf{indcpa\text{-}ki}}_{\mathsf{PKE},A}(\cdot) \tag{14}$$

$$1 - \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 0\,] \leq \frac{1}{2} + \epsilon\ . \tag{15}$$

Subtracting and re-arranging terms, we have Equation (13). We turn to justifying the two equations above. The first follows from the adversary definitions. For the second, consider the following games, where G_1 includes the boxed code and G_2 does not:

$\underline{\textsc{Main}\ \boxed{\mathrm{G}^A_1(\lambda)},\ \mathrm{G}^A_2(\lambda)}$
$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$d \leftarrow\!\!\$\ \{0,1\}$
$Q_0, Q_1 \leftarrow \emptyset$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda, \varepsilon)$
For $i = 1$ to $|\mathbf{m}_d|$ do
  $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_d[i])$
$d' \leftarrow\!\!\$\ A(1^\lambda, t, \mathbf{c}, (ek, hk))$
Return $(d = d')$

$\underline{\textsc{Enc}(m)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}\ ;\ z \leftarrow m \,\|\, 0^{\ell_2(\lambda)}\ ;\ t_1 \leftarrow z \oplus x$
If $(r \in Q_0)$ then $\mathsf{bad} \leftarrow \mathsf{true}\ ;\ \boxed{t_1 \leftarrow T_0[r]\ ;\ x \leftarrow t_1 \oplus z}$
$T_0[r] \leftarrow t_1\ ;\ Q_0 \leftarrow Q_0 \cup \{r\}$
$y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ t_2 \leftarrow y \oplus r$
If $(x \in Q_1)$ then $\mathsf{bad} \leftarrow \mathsf{true}\ ;\ \boxed{t_2 \leftarrow T_1[x]\ ;\ y \leftarrow t_2 \oplus r}$
$T_1[x] \leftarrow t_2\ ;\ Q_1 \leftarrow Q_1 \cup \{x\}$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})\ ;$ Return c

The games are identical-until-bad so by the Fundamental Lemma of Game Playing [26] we have

$$\begin{aligned}
1 - \Pr[\,\mathrm{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, b = 0\,] &= \Pr[\mathrm{G}^A_1(\cdot)] \\
&\leq \Pr[\mathrm{G}^A_2(\cdot)] + \Pr[\mathrm{G}^A_2(\cdot) \text{ sets } \mathsf{bad}] \\
&= \frac{1}{2} + \Pr[\mathrm{G}^A_2(\cdot) \text{ sets } \mathsf{bad}] \\
&\leq \frac{1}{2} + \frac{q(q-1)}{2^{1+\ell_3}} + \frac{q(q-1)}{2^{1+\ell_1+\ell_2}}\ ,
\end{aligned}$$

which establishes Equation (15). It remains to show that S is computationally unpredictable. By Lemma 4.3, it suffices to show that S is simple computationally unpredictable. Let P' be a PT simple predictor and consider the invertor defined below:

$\underline{I(1^\lambda, ek, c')}$	$\underline{\textsc{Enc}(m)}$
$d \leftarrow\$ \{0,1\}$; $j \leftarrow\$ \{1,\ldots,q(\lambda)\}$; $i \leftarrow 0$	$i \leftarrow i+1$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\$ A(1^\lambda, \varepsilon)$	If $(i = j)$ then return c'
For $i = 1$ to $\lvert\mathbf{m}_d\rvert$ do	$x_i \leftarrow\$ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}$; $y_i \leftarrow\$ \{0,1\}^{\ell_3(\lambda)}$
$\quad \mathbf{c}[i] \leftarrow\$ \textsc{Enc}(\mathbf{m}_d[i])$	$c_i \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x_i \,\|\, y_i, 1^{\mathsf{TF.ol}(\lambda)})$
$L \leftarrow ((ek, t, d), \mathbf{c})$; $u \leftarrow\$ P'(1^\lambda, L)$	Return c_i
Return $u[2, \lvert u\rvert]$	

Then we claim that

$$\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le q \cdot \mathsf{Adv}^{\mathsf{pow}}_{\mathsf{TF},\ell_1+\ell_2,I}(\cdot) + \epsilon + \frac{q}{2^{\ell_3}} \ . \tag{16}$$

The simple computational unpredictability of S follows from the assumption that $\mathsf{TF} \in \mathsf{OW}_{\ell_1+\ell_2}$ and $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible. To justify Equation (16), consider the games whose Main procedures are below, with Enc continuing to be the same as for games $\mathrm{G}^A_1(\lambda), \mathrm{G}^A_2(\lambda)$ above:

$\underline{\textsc{Main}\ \boxed{\mathrm{G}^{A,P'}_3(\lambda)},\ \mathrm{G}^{A,P'}_4(\lambda)}$	$\underline{\textsc{Main}\ \mathrm{G}^{A,P'}_5(\lambda)}$
$ek \leftarrow\$ \mathsf{TF.Kg}(1^\lambda)$; $hk \leftarrow\$ \mathsf{H.Kg}(1^\lambda)$	$ek \leftarrow\$ \mathsf{TF.Kg}(1^\lambda)$; $hk \leftarrow\$ \mathsf{H.Kg}(1^\lambda)$
$d \leftarrow\$ \{0,1\}$; $Q_0, Q_1 \leftarrow \emptyset$; $(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\$ A(1^\lambda, \varepsilon)$	$d \leftarrow\$ \{0,1\}$; $Q_0, Q_1 \leftarrow \emptyset$; $(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\$ A(1^\lambda, \varepsilon)$
For $i = 1$ to $\lvert\mathbf{m}_d\rvert$ do $\mathbf{c}[i] \leftarrow\$ \textsc{Enc}(\mathbf{m}_d[i])$	For $i = 1$ to $\lvert\mathbf{m}_d\rvert$ do $\mathbf{c}[i] \leftarrow\$ \textsc{Enc}(\mathbf{m}_d[i])$
$L \leftarrow ((ek, t, d), \mathbf{c})$; $u \leftarrow\$ P'(1^\lambda, L)$	$L \leftarrow ((ek, t, d), \mathbf{c})$; $u \leftarrow\$ P'(1^\lambda, L)$
Return $(u[2, \lvert u\rvert] \in Q_0 \cup Q_1)$	Return $(u[2, \lvert u\rvert] \in Q_1)$

Then

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) &\le \Pr[\mathrm{G}^{A,P'}_3(\cdot)] \\
&\le \Pr[\mathrm{G}^{A,P'}_4(\cdot)] + \Pr[\mathrm{G}^{A,P'}_4(\cdot) \text{ sets } \mathsf{bad}] \\
&\le \Pr[\mathrm{G}^{A,P'}_4(\cdot)] + \epsilon \\
&\le \Pr[\mathrm{G}^{A,P'}_5(\cdot)] + \epsilon + \frac{q}{2^{\ell_3}} \\
&\le q \cdot \mathsf{Adv}^{\mathsf{pow}}_{\mathsf{TF},\ell_1+\ell_2,I}(\cdot) + \epsilon + \frac{q}{2^{\ell_3}} \ ,
\end{aligned}$$

establishing Equation (16).

Finally we exhibit PT algorithms S_0, S_1 such that $S = \mathsf{Prl}[S_0, S_1]$—

$\underline{S_0(1^\lambda)}$	$\underline{S_1^{\textsc{Hash}}(1^\lambda, (ek, m))}$
$ek \leftarrow\$ \mathsf{TF.Kg}(1^\lambda)$; $d \leftarrow\$ \{0,1\}$	$r \leftarrow\$ \{0,1\}^{\ell_3(\lambda)}$; $t_1 \leftarrow \textsc{Hash}(0\|r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\$ A(1^\lambda, \varepsilon)$	$x \leftarrow (m \,\|\, 0^{\ell_2(\lambda)}) \oplus t_1$; $t_2 \leftarrow \textsc{Hash}(1\|x, 1^{\ell_3(\lambda)})$
Return $((ek, t, d), ((ek, \mathbf{m}_d[1]), \ldots, (ek, \mathbf{m}_d[v(\lambda)])))$	$y \leftarrow t_2 \oplus r$; $c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})$
	Return c

Note that S_1 makes only two oracle queries and its running time is τ. As per our wlog assumption on A discussed above, the length of t is λ, so the length of the leakage (ek, t, d) output by S_0 is σ. This shows that $S \in \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q'}$, concluding the proof. ∎

We remark that the assumption that TF is partially one-way, rather than merely one-way, is crucial to the proof above. This is because there is a one-way TF that makes the source S above predictable. For example, let $\mathsf{TF.EKg} = \mathsf{F.EKg}$ and $\mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)}) = x \,\|\, \mathsf{F.Ev}(1^\lambda, ek, y, 1^{\mathsf{F.ol}(\lambda)})$, where F is a trapdoor permutation family with $\mathsf{F.il} = \ell_3$. This counter-example TF is one-way but not $(\ell_1+\ell_2)$-partially one-way, and makes S predictable. Theorem 5.9 shows that a version of reset-security will allow us to relax the assumption on TF to plain one-wayness.

Theorem 5.9 Let $\mathsf{TF}, \mathsf{H}, \ell_1, \ell_2, \ell_3$ be as in Theorem 5.8, and let $\mathsf{PKE} = \mathsf{OAEP}[\mathsf{H}, \mathsf{TF}, \ell_1, \ell_2, \ell_3]$. Assume $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible. Let $\tau = \mathcal{O}(\mathsf{T}_{\mathsf{TF.Ev}} + \ell_1 + \ell_2 + \ell_3)$ and let $q' = 2$. Let σ be defined by $\sigma(\lambda) = \lambda + \mathsf{PKE.ekl}(\lambda) + 1$ for all $\lambda \in \mathbb{N}$. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q'}]$ and $\mathsf{TF} \in \mathsf{OW}$, then $\mathsf{PKE} \in \mathsf{IND\text{-}CPA\text{-}KI}$.

Proof of Theorem 5.9: Let A be a PT adversary for game $\text{IND-CPA-KI}^A_{\mathsf{PKE}}(\lambda)$. Let q, ϵ be as in the proof of Theorem 5.8. We construct source S and distinguisher D exactly as in the proof of Theorem 5.8, so that Equation (13) continues to be true. The difference is that, rather than showing $S \in \mathcal{S}^{\rm cup} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q'}$ assuming $\mathsf{TF} \in \mathsf{OW}_{\ell_1+\ell_2}$, we will now show that $S \in \mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q'}$ assuming only that $\mathsf{TF} \in \mathsf{OW}$. Since we are using the same S as in Theorem 5.8, we can carry over the argument that $S \in \mathcal{S}^{\rm prl}_{\tau,\sigma,q'}$. Thus, it remains to show that $S \in \mathcal{S}^{\rm crs}$. The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm crs} \cap \mathcal{S}^{\rm prl}_{\tau,\sigma,q'}]$ and that $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible.

Let R be an arbitrary PT reset adversary. Wlog, assume that R never repeats a query to its HASH oracle. Assume further that each of R's queries $(u, 1^\ell)$ to HASH satisfies the following: (i) If $u[1] = 0$ then $|u| = \ell_3(\lambda)+1$ and $\ell = \ell_1(\lambda)+\ell_2(\lambda)$, and (ii) If $u[1] = 1$ then $|u| = \ell_1(\lambda)+\ell_2(\lambda)+1$ and $\ell = \ell_3(\lambda)$. Let p be a polynomial such that the number of HASH queries of R in game $\text{Reset}^R_S(\lambda)$ is at most $p(\lambda)$ for all $\lambda \in \mathbb{N}$. We'll construct a PT adversary I such that

$$\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot) \le q \cdot \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{TF},I}(\cdot) + \frac{p \cdot q}{2^{\ell_3}} + 2\epsilon \ . \tag{17}$$

Then, from the assumption that $\mathsf{TF} \in \mathsf{OW}$ and that $2^{-\ell_1-\ell_2}, 2^{-\ell_3}$ are negligible, it follows that $S \in \mathcal{S}^{\rm crs}$. Proceeding, consider games G_1–G_7 in Fig. 16. Let a be the challenge bit of game $\text{Reset}^R_S(\lambda)$. Then

$$\Pr[\text{G}_1^{A,R}(\lambda)] = \Pr[\,\text{Reset}^R_S(\lambda) \,|\, a = 1\,] \text{ and } \Pr[\text{G}_7^{A,R}(\lambda)] = 1 - \Pr[\,\text{Reset}^R_S(\lambda) \,|\, a = 0\,]$$

We explain the game chain up to the terminal one. In game $\text{G}_2^{A,R}(\lambda)$, we no longer maintain consistency among queries: in each ENC operation, the strings x and y are uniformly random, independent of anything else. The two games $\text{G}_1^{A,R}(\lambda)$ and $\text{G}_2^{A,R}(\lambda)$ are identical-until-bad. Then

$$\Pr[\text{G}_1^{A,R}(\cdot)] - \Pr[\text{G}_2^{A,R}(\cdot)] \le \Pr[\text{G}_2^{A,R}(\cdot) \text{ sets } \mathsf{bad}] \le \sum_{i=0}^{q-1} \frac{i}{2^{\ell_3}} + \frac{i}{2^{\ell_1+\ell_2}} \le \epsilon \ .$$

In game $\text{G}_3^{A,R}(\lambda)$, we delay the writing to $T_0[r]$ until R queries HASH at this point. Then

$$\Pr[\text{G}_2^{A,R}(\cdot)] = \Pr[\text{G}_3^{A,R}(\cdot)] \ .$$

In game $\text{G}_4^{S,R}(\lambda)$, we won't write to $T_0[r]$ if R queries $(0 \,\|\, r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$ before querying $(1 \,\|\, x, 1^{\ell_3(\lambda)})$. Games $\text{G}_3^{S,R}(\lambda)$ and $\text{G}_4^{S,R}(\lambda)$ are identical-until-bad. Before R queries $(1 \,\|\, x, 1^{\ell_3(\lambda)})$, the string r is independent of whatever R receives, and thus

$$\Pr[\text{G}_3^{S,R}(\cdot)] - \Pr[\text{G}_4^{S,R}(\cdot)] \le \Pr[\text{G}_4^{S,R}(\cdot) \text{ sets } \mathsf{bad}] \le \frac{q \cdot p}{2^{\ell_3}}.$$

In game $\text{G}_5^{A,R}(\lambda)$, we drop the writing to $T_0[r]$ entirely. This may cause inconsistency with game $\text{G}_4^{S,R}(\lambda)$ only if R queries $(1 \,\|\, x, 1^{\ell_3(\lambda)})$ and then $(0 \,\|\, r, 1^{\ell_1(\lambda)+\ell_2(\lambda)})$ to HASH. Games $\text{G}_4^{S,R}(\lambda)$ and $\text{G}_5^{S,R}(\lambda)$ are identical-until-coll. Then

$$\Pr[\text{G}_4^{S,R}(\cdot)] - \Pr[\text{G}_5^{S,R}(\cdot)] \le \Pr[\text{G}_5^{S,R}(\cdot) \text{ sets } \mathsf{coll}] \ .$$

Note that in game $\text{G}_5^{S,R}(\lambda)$, the answers of HASH and the values returned by ENC are independent. We now construct adversary I such that

$$\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{TF},I}(\cdot) \ge \frac{1}{q} \Pr[\text{G}_5^{S,R}(\cdot) \text{ sets } \mathsf{coll}] \ .$$

The construction of I is shown below:

MAIN $\boxed{\mathrm{G}_1^{A,R}(\lambda)}$, $\mathrm{G}_2^{A,R}(\lambda)$

$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ M, Q_0, Q_1 \leftarrow \emptyset$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda)$
For $i = 1$ to $|\mathbf{m}_b|$ do $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_b[i])$
$L \leftarrow ((ek, t, b), \mathbf{c})\ ;\ d' \leftarrow\!\!\$\ R^{\textsc{Hash}}(1^\lambda, L)$; Return $(d' = 1)$

$\textsc{Enc}(m)$
$r \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}$
$y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ z \leftarrow m \,\|\, 0^{\ell_2(\lambda)}\ ;\ t_1 \leftarrow x \oplus z$
If $r \in Q_0$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{t_1 \leftarrow T_0[r]\ ;\ x \leftarrow t_1 \oplus z}$
$T_0[r] \leftarrow t_1\ ;\ Q_0 \leftarrow Q_0 \cup \{r\}\ ;\ t_2 \leftarrow y \oplus r$
If $x \in Q_1$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{t_2 \leftarrow T_1[x]\ ;\ y \leftarrow t_2 \oplus r}$
$Q_1 \leftarrow Q_1 \cup \{x\}\ ;\ T_1[x] \leftarrow t_2$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})\ ;$ Return c

$\textsc{Hash}(u, 1^\ell)$
$v \leftarrow u[2, |u|]\ ;\ s \leftarrow u[1]$
If $T_s[v] = \perp$ then $T_s[v] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T_s[v]$

MAIN $\boxed{\mathrm{G}_3^{A,R}(\lambda)}$, $\mathrm{G}_4^{A,R}(\lambda)$

$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ M, Q_0, Q_1 \leftarrow \emptyset$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda)$
For $i = 1$ to $|\mathbf{m}_b|$ do $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_b[i])$
$L \leftarrow ((ek, t, b), \mathbf{c})\ ;\ d' \leftarrow\!\!\$\ R^{\textsc{Hash}}(1^\lambda, L)$; Return $(d' = 1)$

$\textsc{Enc}(m)$
$x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}\ ;\ y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}$
$z \leftarrow m \,\|\, 0^{\ell_2(\lambda)}\ ;\ T_1[x] \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ t_2 \leftarrow T_1[x]$
$t_1 \leftarrow x \oplus z\ ;\ r \leftarrow y \oplus t_2$
$V_0[r] \leftarrow t_1\ ;\ Q_0 \leftarrow Q_0 \cup \{r\}$
$V_1[x] \leftarrow r\ ;\ Q_1 \leftarrow Q_1 \cup \{x\}$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})\ ;$ Return c

$\textsc{Hash}(u, 1^\ell)$
$v \leftarrow u[2, |u|]\ ;\ s \leftarrow u[1]$
If $s = 0$ and $v \in Q_0$ then
  If $v \in M$ then $\mathsf{coll} \leftarrow \mathsf{true}$; $T_0[v] \leftarrow V_0[v]$
  Else $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T_0[v] \leftarrow V_0[v]}$
If $s = 1$ and $v \in Q_1$ then $r \leftarrow V_1[v]\ ;\ M \leftarrow M \cup \{r\}$
If $T_s[v] = \perp$ then $T_s[v] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T_s[v]$

MAIN $\mathrm{G}_5^{A,R}(\lambda)$

$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ M, Q_0, Q_1 \leftarrow \emptyset$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda)$
For $i = 1$ to $|\mathbf{m}_b|$ do $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_b[i])$
$L \leftarrow ((ek, t, b), \mathbf{c})\ ;\ d' \leftarrow\!\!\$\ R^{\textsc{Hash}}(1^\lambda, L)$; Return $(d' = 1)$

$\textsc{Enc}(m)$
$x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}\ ;\ y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}$
$z \leftarrow m \,\|\, 0^{\ell_2(\lambda)}\ ;\ T_1[x] \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ t_2 \leftarrow T_1[x]$
$t_1 \leftarrow x \oplus z\ ;\ r \leftarrow y \oplus t_2$
$V_0[r] \leftarrow t_1\ ;\ Q_0 \leftarrow Q_0 \cup \{r\}\ ;\ V_1[x] \leftarrow r\ ;\ Q_1 \leftarrow Q_1 \cup \{x\}$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})\ ;$ Return c

$\textsc{Hash}(u, 1^\ell)$
$v \leftarrow u[2, |u|]\ ;\ s \leftarrow u[1]$
If $s = 0$ and $v \in Q_0$ then
  If $v \in M$ then $\mathsf{coll} \leftarrow \mathsf{true}$
If $s = 1$ and $v \in Q_1$ then $r \leftarrow V_1[v]\ ;\ M \leftarrow M \cup \{r\}$
If $T_s[v] = \perp$ then $T_s[v] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T_s[v]$

MAIN $\mathrm{G}_6^{A,R}(\lambda)$, $\boxed{\mathrm{G}_7^{A,R}(\lambda)}$

$ek \leftarrow\!\!\$\ \mathsf{TF.Kg}(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ M, Q_0, Q_1 \leftarrow \emptyset$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda)$
For $i = 1$ to $|\mathbf{m}_b|$ do $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_b[i])$
$L \leftarrow ((ek, t, b), \mathbf{c})\ ;\ d' \leftarrow\!\!\$\ R^{\textsc{Hash}}(1^\lambda, L)$
Return $(d' = 1)$

$\textsc{Enc}(m)$
$r \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}$
$y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}\ ;\ z \leftarrow m \,\|\, 0^{\ell_2(\lambda)}\ ;\ t_1 \leftarrow x \oplus z$
If $r \in Q_0$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{t_1 \leftarrow T_0[r]\ ;\ x \leftarrow t_1 \oplus z}$
$T_0[r] \leftarrow t_1\ ;\ Q_0 \leftarrow Q_0 \cup \{r\}\ ;\ t_2 \leftarrow y \oplus r$
If $x \in Q_1$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{t_2 \leftarrow T_1[x]\ ;\ y \leftarrow t_2 \oplus r}$
$Q_1 \leftarrow Q_1 \cup \{x\}\ ;\ T_1[x] \leftarrow t_2$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})\ ;$ Return c

$\textsc{Hash}(u, 1^\ell)$
$z \leftarrow\!\!\$\ \{0,1\}^\ell\ ;$ Return z

Figure 16: **Games for the proof of Theorem 5.9.** Games $\mathrm{G}_1, \mathrm{G}_3$, and G_7 contain the corresponding boxed statements but the other games do not.

$I(1^\lambda, ek, c')$
$b \leftarrow\!\!\$\ \{0,1\}\ ;\ j \leftarrow\!\!\$\ \{1, \ldots q(\lambda)\}\ ;\ i \leftarrow 0$
$(t, \mathbf{m}_0, \mathbf{m}_1) \leftarrow\!\!\$\ A(1^\lambda)\ ;\ Q_0, Q_1 \leftarrow \emptyset$
For $i = 1$ to $|\mathbf{m}_b|$ do $\mathbf{c}[i] \leftarrow\!\!\$\ \textsc{Enc}(\mathbf{m}_b[i])$
$L \leftarrow ((ek, t, b), \mathbf{c})\ ;\ R^{\textsc{HashSim}}(1^\lambda, (ek, b, t))$
For $r \in Q_0, x \in Q_1$ do
  $y \leftarrow T_1[x] \oplus r\ ;\ c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})$
  If $c = c'$ then return $x \,\|\, y$

$\textsc{Enc}(m)$
$i \leftarrow i + 1\ ;\ x \leftarrow\!\!\$\ \{0,1\}^{\ell_1(\lambda)+\ell_2(\lambda)}\ ;\ y \leftarrow\!\!\$\ \{0,1\}^{\ell_3(\lambda)}$
$c \leftarrow \mathsf{TF.Ev}(1^\lambda, ek, x \,\|\, y, 1^{\mathsf{TF.ol}(\lambda)})$
If $i = j$ then $c \leftarrow c'$
Return c

$\textsc{HashSim}(u, 1^\ell)$
$v \leftarrow u[2, |u|]\ ;\ s \leftarrow u[1]$
$Q_s \leftarrow Q_s \cup \{v\}\ ;\ T_s[v] \leftarrow\!\!\$\ \{0,1\}^\ell\ ;$ Return $T_s[v]$

In game $\mathrm{G}_6^{A,R}(\lambda)$, procedure HASH explicitly returns random answers, independent of ENC. Then

$$\Pr[\mathrm{G}_5^{A,R}(\cdot)] = \Pr[\mathrm{G}_6^{A,R}(\cdot)] \ .$$

In game $\mathrm{G}_7^{A,R}(\lambda)$, the calls to ENC still need to maintain consistency among its answers, but these are still independent of HASH. Games $\mathrm{G}_6^{A,R}(\lambda)$ and $\mathrm{G}_7^{A,R}(\lambda)$ are identical-until-bad. Then

$$\Pr[\mathrm{G}_6^{A,R}(\cdot)] - \Pr[\mathrm{G}_7^{A,R}(\cdot)] \leq \Pr[\mathrm{G}_7^{A,R}(\cdot) \text{ sets } \mathsf{bad}] \leq \epsilon \ .$$

Hence

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\cdot) &= \Pr[\,\mathrm{Reset}_S^R(\lambda) \,|\, a = 1\,] + \Pr[\,\mathrm{Reset}_S^R(\lambda) \,|\, a = 0\,] - 1 \\
&= \Pr[\mathrm{G}_1^{A,R}(\cdot)] - \Pr[\mathrm{G}_7^{A,R}(\cdot)] \\
&= \sum_{i=1}^{6} \Pr[\mathrm{G}_i^{A,R}(\cdot)] - \Pr[\mathrm{G}_{i+1}^{A,R}(\cdot)] \\
&\leq 2\epsilon + \frac{p \cdot q}{2^{\ell_3}} + q \cdot \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{TF},I}(\cdot)
\end{aligned}$$

yielding Equation (17). ∎

5.9 Proofs of storage

Client Alice has uploaded her file x to a server in the cloud. She is worried that the server is malicious and, to save space, is not actually storing x. A storage-auditing protocol [8, 87, 104] allows Alice to efficiently verify that the server stores her file. A particularly natural and canonical protocol for this task, embodied in the SafeStore system [91], is for Alice to send the server a random challenge hk and expect in response $H(x\|hk)$ where H is a hash function. Ristenpart, Shacham and Shrimpton (RSS) [103] show that this protocol is secure if H is a RO. Interestingly, however, they show that H being indifferentiable from a RO [97] is *not* enough for the protocol to be secure. We show that $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{splt}}]$-security *is* enough. Our instantiation interprets hk as a key for a UCE family H, meaning we let $H(x\|hk) = \mathsf{H.Ev}(1^\lambda, hk, x, 1^\lambda)$. This results in a natural, simple standard model, secure storage-auditing protocol.

DEFINITIONS. The security definition of RSS [103] assumes the message (file) x is a uniformly random string as a function of which the adversary has computed some information s that it stores. In asymptotic terms, the requirement is then that the adversary cannot defeat the audit (meaning, provide the correct hash value) as long as $2^{|s|-|x|}$ is negligible. Real files are, however, not uniformly random strings. We accordingly provide a stronger and more general definition that implies that of RSS.

Let H be a family of functions with $\mathsf{H.IL}(\lambda) = \mathbb{N}$ and $\lambda \in \mathsf{H.OL}(\lambda)$ for all $\lambda \in \mathbb{N}$. In game $\mathrm{Store}^A_{\mathsf{H}}(\lambda)$ of Fig. 17, the adversary $A = (A_1, A_2)$ is a pair of algorithms. Algorithm A_1 produces the message x together with the information s about x that it will store. (So s can be a function of x.) Then a challenge hk is picked at random, and, to win, A_2, given s and hk, must be able to provide $y = \mathsf{H.Ev}(1^\lambda, hk, x, 1^\lambda)$. We let $\mathsf{Adv}^{\mathsf{store}}_{\mathsf{H},A}(\lambda) = \Pr[\mathrm{Store}^A_{\mathsf{H}}(\lambda)]$ for all $\lambda \in \mathbb{N}$. This advantage cannot be small for all A, for A_1 can let $s = x$ and then A_2 can compute the correct hash, but this corresponds to a server who does, indeed, store the file. So let us define the guessing probability $\mathrm{Guess}_A(\cdot)$ of A via

$$\mathrm{Guess}_A(\lambda) = \sum_{s'} \Pr[s = s'] \cdot \max_{x'} \Pr[\,x = x' \,|\, s = s'\,]$$

for all $\lambda \in \mathbb{N}$, where the probability is over $(x, s) \leftarrow\!\!\$\ A_1(1^\lambda)$. We say that A has *high min-entropy* if $\mathrm{Guess}_A(\cdot)$ is negligible. Having high min-entropy corresponds to A cheating, meaning not storing information sufficient to recover the file. We say that a hash family H is *storage-auditing secure* if $\mathsf{Adv}^{\mathsf{store}}_{\mathsf{H},A}(\cdot)$ is negligible for all PT adversaries A of high min-entropy. This captures the requirement that adversaries that fail to store enough information to recover the file will also fail to pass the audit protocol. Let STORE denote the set of all storage-auditing secure hash families.

$\underline{\text{MAIN Store}^{A}_{\mathsf{H}}(\lambda)}$
$(x, s) \leftarrow\!\!\$\ A_1(1^\lambda)\ ;\ hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)\ ;\ y' \leftarrow\!\!\$\ A_2(1^\lambda, hk, s)$
$y \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^\lambda)\ ;$ Return $(y = y')$

Figure 17: **Game defining storage-auditing security.**

RESULTS. The following says that UCE security implies storage-auditing security. The assumption of $\mathsf{UCE}[\mathcal{S}^{\rm sup}]$-security would suffice, but in fact we can use an even weaker assumption, namely $\mathsf{UCE}[\mathcal{S}^{\rm sup} \cap \mathcal{S}^{\rm splt} \cap \mathcal{S}^{\rm one}]$-security, where $\mathcal{S}^{\rm one}$ is the class of sources that make at most one oracle query to HASH.

Theorem 5.10 Let H be a hash family. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm sup} \cap \mathcal{S}^{\rm splt} \cap \mathcal{S}^{\rm one}]$ then $\mathsf{H} \in \mathsf{STORE}$.

Proof: Let A be a PT adversary of high min-entropy. We'll construct a PT source $S \in \mathcal{S}^{\rm sup} \cap \mathcal{S}^{\rm splt} \cap \mathcal{S}^{\rm one}$ and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathsf{store}}_{\mathsf{H},A}(\lambda) \le \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\lambda) + 2^{-\lambda} \tag{18}$$

for every $\lambda \in \mathbb{N}$. The theorem follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\rm sup} \cap \mathcal{S}^{\rm splt} \cap \mathcal{S}^{\rm one}]$. The constructions of S and D are shown below:

$\underline{S^{\text{HASH}}(1^\lambda)}$	$\underline{D(1^\lambda, hk, L)}$
$(x, s) \leftarrow\!\!\$\ A_1(1^\lambda)\ ;\ y \leftarrow \text{HASH}(x, 1^\lambda)$	$(y, s) \leftarrow L\ ;\ y' \leftarrow A_2(1^\lambda, hk, s)$
$L \leftarrow (y, s)\ ;$ Return L	If $(y = y')$ then $b' \leftarrow 1$ else $b' \leftarrow 0\ ;$ Return b'

The source S indeed makes a single query to HASH. Let b denote the challenge bit in game $\text{UCE}^{S,D}_{\mathsf{H}}(\cdot)$. Then for all $\lambda \in \mathbb{N}$ we have

$$\Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\lambda) \,|\, b = 1\,] = \mathsf{Adv}^{\mathsf{store}}_{\mathsf{H},A}(\lambda)$$
$$\Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\lambda) \,|\, b = 0\,] = 1 - 2^{-\lambda}\ .$$

Summing yields Equation (18). We claim that S is statistically unpredictable. By Lemma 4.3, it suffices to show that S is simple statistically unpredictable. Consider an arbitrary simple predictor P'. Then $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le \text{Guess}_A(\cdot)$. From the assumption that A is of high min-entropy, the claim follows.

Finally we exhibit, below, PT algorithms S_0, S_1 such that $S = \mathsf{Splt}[S_0, S_1]$—

$\underline{S_0(1^\lambda)}$	$\underline{S_1(1^\lambda, \mathbf{y})}$
$(x, s) \leftarrow\!\!\$\ A_1(1^\lambda)\ ;\ \boldsymbol{\ell}[1] \leftarrow \mathsf{H.ol}(\lambda)$	Return $\mathbf{y}[1]$
Return $(s, x, 1^{\boldsymbol{\ell}})$	

This shows that $S \in \mathcal{S}^{\rm splt}$, concluding the proof. ▌

5.10 Correlated-input hash functions

Goyal, O'Neill, and Rao (GOR) [79] introduced the notion of correlated-input hash (CIH) function families and proposed several notions of security for them. They provided constructions achieving limited CIH security from the q-DHI assumption of [35] and from RKA secure blockciphers. But achieving full CIH security in the standard model has remained open. Here, we show that UCE-secure function families achieve this goal, being selective (pseudorandomness) CIH secure in the terminology of GOR.

DEFINITIONS. Let H be a function family with input length $\mathsf{H.il}$ and output length $\mathsf{H.ol}$. Game $\text{CIH}^{A}_{\mathsf{H}}(\lambda)$ of Fig. 18 captures the selective pseudorandom correlated-input hash security notion of [79] adapted to our setting and notation. An adversary $A = (A_1, A_2)$ is a pair of algorithms. Algorithm A_1 on input

MAIN $\mathrm{CIH}^A_{\mathsf{H}}(\lambda)$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$; $\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$
For $i = 1$ to $\lvert\mathbf{m}\rvert$ do $\mathbf{h}_1[i] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, \mathbf{m}[i], 1^{\mathsf{H.ol}(\lambda)})$; $\mathbf{h}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$b' \leftarrow\!\!\$\ A_2(1^\lambda, hk, \mathbf{h}_b)$; Return $(b' = b)$

Figure 18: **The** CIH **game.**

1^λ returns $\mathbf{m}$, a $v(\lambda)$-vector of distinct $\mathsf{H.il}(\lambda)$-bit strings, where the polynomial v depends on A. The guessing probability Guess_A of A is the function that on input $\lambda \in \mathbb{N}$ returns the maximum, over all i, m, of $\Pr[\mathbf{m}[i] = m]$, the probability over $\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$. We say that A has *high min-entropy* if $\mathrm{Guess}_A(\cdot)$ is negligible. We let $\mathsf{Adv}^{\mathsf{cih}}_{\mathsf{H},A}(\lambda) = 2\Pr[\mathrm{CIH}^A_{\mathsf{H}}(\lambda)] - 1$. We say that H is CIH-secure if $\mathsf{Adv}^{\mathsf{cih}}_{\mathsf{H},A}$ is negligible for all PT A that have high min-entropy. Note that the high min-entropy assumption on A is necessary to achieve security. We let CIH be the set of all CIH-secure function families.

RESULTS. Let H be a function family with input length $\mathsf{H.il}$ and output length $\mathsf{H.ol}$. The following theorem says that if H is UCE-secure then it is also CIH-secure. With regard to the specific UCE assumption, $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$ would suffice, but in fact the weaker $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{splt}}]$ is enough.

Theorem 5.11 Let H be a FOL function family. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{splt}}]$ then $\mathsf{H} \in \mathsf{CIH}$.

Proof: Let $A = (A_1, A_2)$ be a PT CIH adversary. We will construct a PT source $S \in \mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{splt}}$ and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathsf{cih}}_{\mathsf{H},A}(\cdot) = \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)\ . \tag{19}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{splt}}]$. The constructions of S and D are shown below:

$S^{\mathrm{HASH}}(1^\lambda)$	$D(1^\lambda, hk, L)$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$	$\mathbf{h} \leftarrow L$
For $i = 1$ to $\lvert\mathbf{m}\rvert$ do $\mathbf{h}[i] \leftarrow \mathrm{HASH}(\mathbf{m}[i], 1^{\mathsf{H.ol}(\lambda)})$	$b' \leftarrow\!\!\$\ A_2(1^\lambda, hk, \mathbf{h})$
$L \leftarrow \mathbf{h}$; Return L	Return b'

We now show that S is statistically unpredictable. By Lemma 4.3, it suffices to show that S is simple statistically unpredictable. Consider an arbitrary simple predictor P'. The leakage it receives consists of random strings unrelated to the messages that S queries to its oracle, so $\mathsf{Adv}^{\mathsf{spred}}_{S,P'}(\cdot) \le v \cdot \mathrm{Guess}_A(\cdot)$ where v is the polynomial associated to A as per the definition. Finally we exhibit, below, PT algorithms S_0, S_1 such that $S = \mathsf{Splt}[S_0, S_1]$—

$S_0(1^\lambda)$	$S_1(1^\lambda, \mathbf{y})$
$\mathbf{m} \leftarrow\!\!\$\ A_1(1^\lambda)$	Return $\mathbf{y}$
For $i = 1, \ldots, \lvert\mathbf{m}\rvert$ do $\boldsymbol{\ell}[i] \leftarrow \mathsf{H.ol}(\lambda)$	
Return $(\varepsilon, \mathbf{m}, 1^{\boldsymbol{\ell}})$	

This shows that $S \in \mathcal{S}^{\mathrm{splt}}$, concluding the proof. ▌

We note that UCE security does not imply adaptive CIH security, where the inputs can depend on the key.

5.11 Adaptively secure garbling with short tokens

BHR1 [17] introduce garbling schemes as an abstraction of the garbled circuit technique that originates with Yao [106]. They provide definitions for privacy, obliviousness and authenticity. These however capture the standard, static-security requirements met by the standard Yao-style constructions. BHR2 [16] point

out that applications like one-time programs [76] and secure-outsourcing [70] require adaptive security, and provide corresponding definitions of adaptive security (again for privacy, obliviousness and authenticity) for garbling schemes.

Ideally, we want garbling schemes where the garbled inputs are short. (Garbling schemes take a function to produce a garbled function, and also associate to an input a garbled input. We say that the scheme has short garbled inputs if the size of the garbled inputs depends only on the security parameter and the input and output lengths of the original function. We note that the secure-outsourcing protocol of [70] requires short garbled inputs in order to be non-trivial.) Statically-secure schemes naturally have this feature, but retaining it while providing adaptive security is challenging. Schemes with this feature have been provided in the ROM [16, 7]. But in the standard model, known constructions achieving adaptive security [16, 78] have long garbled inputs, meaning ones of size proportional to the size of the circuit being garbled. Standard-model adaptively-secure garbling with short tokens has remained open.

Here we resolve this problem with schemes based on a $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$-secure family. We give two garbling schemes, GaP and GaAO, that have short tokens. The former provides adaptive privacy, while the latter provides adaptive obliviousness and authenticity. Both assume only a $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$-secure hash family.

We note that the secure-outsourcing protocol of [70] uses Fully Homomorphic Encryption [72], requiring the garbling scheme it (also) uses to be secure in the standard model. (A ROM garbling scheme does not work for them.) Thus we obtain the first non-trivial instantiation of the secure-outsourcing protocol of [70].

Definitions. BHR1 [17] give the following formalization of circuits. A *circuit* is defined as a 6-tuple $f = (n, m, q, A, B, G)$ where $n \geq 2$ is the number of *inputs*, $m \geq 1$ is the number of *outputs*, $q \geq 1$ is the number of *gates*, and $r = n + q$ the number of *wires*. We let $\mathrm{Inputs} = [1\,..\,n]$, $\mathrm{Wires} = [1\,..\,n+q]$, $\mathrm{OutputWires} = [n+q-m+1\,..\,n+q]$, and $\mathrm{Gates} = [n+1\,..\,n+q]$. Then A: Gates $\to$ Wires\OutputWires is a function to identify each gate's *first* incoming wire and B: Gates $\to$ Wires\OutputWires is a function to identify each gate's *second* incoming wire. Finally G: Gates $\times \{0,1\}^2 \to \{0,1\}$ is a function that determines the *functionality* of each gate. We require $A(g) < B(g) < g$ for all $g \in \mathrm{Gates}$.

The conventions above embody all of the following. Gates have two inputs, arbitrary functionality, and arbitrary fan-out. The wires are numbered 1 to $n+q$. Every non-input wire is the outgoing wire of some gate. The ith bit of input is presented along wire i. The ith bit of output is collected off wire $n+q-m+i$. The outgoing wire of each gate serves as the name of that gate. Circuit output wires, i.e., wires in OutputWires, cannot be circuit input wires and cannot be incoming wires to gates. No output wire can be used twice in the output. Requiring $A(g) < B(g) < g$ ensures that the directed graph corresponding to f is acyclic, and that no wire feeds a gate twice. Numbering the gates produces a topological sort when the circuit is viewed as a directed graph.

A *garbling scheme* GS is specified via algorithms $\mathsf{GS.Gb}, \mathsf{GS.En}, \mathsf{GS.De}, \mathsf{GS.Ev}$, and $\mathsf{GS.ev}$ [17]. The first algorithm $\mathsf{GS.Gb}$ is probabilistic; the others are deterministic. Algorithm $\mathsf{GS.Gb}$ takes input a string f describing a function and outputs a triple of strings (F, e, d). Here, f describes a function $\mathsf{GS.ev}(f, \cdot) : \{0,1\}^n \to \{0,1\}^m$. The values $n = f.n$ and $m = f.m$ should be efficiently computable from f. The strings e and d describe functions $\mathsf{GS.En}(e, \cdot) : \{0,1\}^n \to \{0,1\}^*$ and $\mathsf{GS.De}(d, \cdot) : \{0,1\}^* \to \{0,1\}^m \cup \{\perp\}$ respectively. We call λ, f, F, e, d the *security parameter, initial function, garbled function, encoding function* and *decoding function*, respectively. For any $x \in \{0,1\}^n$, we require that $\mathsf{GS.ev}(f, x) = \mathsf{GS.De}(d, Y)$, where $(F, e, d) \xleftarrow{\$} \mathsf{GS.Gb}(1^\lambda, f), X \leftarrow \mathsf{GS.En}(e, x)$ and $Y \leftarrow \mathsf{GS.Ev}(F, X)$. The strings X and Y are called *garbled input* and *garbled output* respectively.

We restrict attention to *projective circuit-garbling schemes* following the terminology of BHR1. Evaluation $\mathsf{GS.ev}$ is the canonical circuit-evaluation function cev below. Encoding $\mathsf{GS.En}(e, \cdot)$ uses the bits of $x = x_1 \cdots x_n$ to select from $e = (X_1^0, X_1^1, \ldots, X_n^0, X_n^1)$ the subvector $X = (X_1^{x_1}, \ldots, X_n^{x_n})$. We say that a garbling scheme has *short* garbled inputs if garbled input lengths depend only on the security parameter λ, the input length n, and output length m of f.

$\underline{\mathsf{cev}(f, x)}$
$(n, m, q, A, B, G) \leftarrow f$; $x_1 \cdots x_n \leftarrow x$
For $g \leftarrow n+1$ to $n+q$ do
  $a \leftarrow A(g)$; $b \leftarrow B(g)$; $x_g \leftarrow G_g(x_a, x_b)$
Return $x_{n+q-m+1} \cdots x_{n+q}$

$\underline{\textsc{Main}\ \mathrm{Prv2}_{\mathsf{GS},\Phi,\mathrm{Sim}}(\lambda)}$ $b \leftarrow\!\!\$\ \{0,1\}\ ;\ b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Garble},\textsc{Input}}(1^\lambda)\ ;$ Return $(b=b')$ $\underline{\textsc{Garble}(f)}$ $n \leftarrow f.n\ ;\ Q \leftarrow \emptyset\ ;\ \tau \leftarrow \varepsilon$ If $b=1$ then $\quad (F,(X_1^0,X_1^1,\ldots,X_n^0,X_n^1),d) \leftarrow\!\!\$\ \mathsf{GS.Gb}(1^\lambda,f)$ Else $(F,d) \leftarrow\!\!\$\ \mathrm{Sim}(1^\lambda,\Phi(f),0)$ Return (F,d)	$\underline{\textsc{Input}(i,c)}$ If $i \notin \{1,\ldots,n\} \setminus Q$ then return $\bot$ $x_i \leftarrow c\ ;\ Q \leftarrow Q \cup \{i\}$ If $\lvert Q\rvert = n$ then $\quad x \leftarrow x_1 \cdots x_n\ ;\ y \leftarrow\!\!\$\ \mathsf{GS.ev}(f,x)\ ;\ \tau \leftarrow y$ If $b=1$ then $X_i \leftarrow X_i^{x_i}$ Else $X_i \leftarrow\!\!\$\ \mathrm{Sim}(1^\lambda,\tau,i,\lvert Q\rvert)$ Return X_i
$\underline{\textsc{Main}\ \mathrm{Obv2}_{\mathsf{GS},\Phi,\mathrm{Sim}}(\lambda)}$ $b \leftarrow\!\!\$\ \{0,1\}\ ;\ b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Garble},\textsc{Input}}(1^\lambda)\ ;$ Return $(b=b')$ $\underline{\textsc{Garble}(f)}$ $n \leftarrow f.n;\ \ Q \leftarrow \emptyset;\ \ \sigma \leftarrow \varepsilon$ If $b=1$ then $\quad (F,(X_1^0,X_1^1,\ldots,X_n^0,X_n^1),d) \leftarrow \mathsf{GS.Gb}(1^\lambda,f)$ Else $F \leftarrow \mathrm{Sim}(1^\lambda,\Phi(f),0)$ Return F	$\underline{\textsc{Input}(i,c)}$ If $i \notin \{1,\ldots,n\} \setminus Q$ then return $\bot$ $x_i \leftarrow c;\ \ Q \leftarrow Q \cup \{i\}$ If $b=1$ then $X_i \leftarrow X_i^{x_i}$ Else $X_i \leftarrow \mathrm{Sim}(1^\lambda,i,\lvert Q\rvert)$ Return X_i
$\underline{\textsc{Main}\ \mathrm{Aut2}_{\mathsf{GS}}(\lambda)}$ $Y \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Garble},\textsc{Input}}(1^\lambda)$ Return $(\mathsf{GS.De}(d,Y) \neq \bot$ and $Y \neq \mathsf{GS.Ev}(F,X))$ $\underline{\textsc{Garble}(f)}$ $n \leftarrow f.n\ ;\ Q \leftarrow \emptyset\ ;\ \sigma \leftarrow \varepsilon$ $(F,(X_1^0,X_1^1,\ldots,X_n^0,X_n^1),d) \leftarrow \mathsf{GS.Gb}(1^\lambda,f)\ ;$ Return F	$\underline{\textsc{Input}(i,c)}$ If $i \notin \{1,\ldots,n\} \setminus Q$ then return $\bot$ $x_i \leftarrow c;\ \ Q \leftarrow Q \cup \{i\},\ \ X_i \leftarrow X_i^{x_i}$ If $\lvert Q\rvert = n$ then $X \leftarrow (X_1,\ldots,X_n)$ Return X_i

Figure 19: **Security notions of a garbling scheme GS.** The scheme is required to be projective. The adversary $\mathcal{A}$ makes a single query to Garble, followed by multiple queries to Input.

We parametrize privacy by a "knob" that measures what we allow to be revealed. The *side-information function* Φ maps f to some information about it, $\Phi(f)$. We require that $f.n$ and $f.m$ be efficiently computable from $\Phi(f)$. In this work, we consider only the side-information function Φ_{topo} that maps a circuit $f=(n,m,q,A,B,G)$ to its *topological circuit* (n,m,q,A,B).

BHR2 [16] identify *privacy, obliviousness,* and *authenticity* as relevant security notions for garbling schemes, and go on to show that privacy is sufficient for one-time programs [76] while obliviousness and authenticity suffice for secure-outsourcing [70].

Let GS be a garbling scheme and let Φ be a side-information function. The first notion *privacy* is defined via game $\mathrm{Prv2}_{\mathsf{GS},\Phi,\mathrm{Sim}}$ of Fig. 19, where Sim, the *simulator*, is an always-terminating algorithm that maintains state across invocations. We define $\mathsf{Adv}^{\mathrm{prv2},\Phi,\mathrm{Sim}}_{\mathsf{GS},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{Prv2}^{\mathcal{A}}_{\mathsf{GS},\Phi,\mathrm{Sim}}(\lambda)] - 1$. Garbling scheme GS is prv2-secure with respect to Φ if for any PT adversary $\mathcal{A}$ there exists a simulator Sim such that $\mathsf{Adv}^{\mathrm{prv2},\Phi,\mathrm{Sim}}_{\mathsf{GS},\mathcal{A}}(\lambda)$ is negligible. We let $\mathsf{PRV2}[\Phi]$ denote the set of all garbling schemes prv2-secure over Φ.

The next notion, *obliviousness*, is defined through game $\mathrm{Obv2}_{\mathsf{GS},\Phi,\mathrm{Sim}}$ of Fig. 19. Advantage is defined through $\mathsf{Adv}^{\mathrm{obv2},\Phi,\mathrm{Sim}}_{\mathsf{GS},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{Obv2}^{\mathcal{A}}_{\mathsf{GS},\Phi,\mathrm{Sim}}(\lambda)] - 1$. Garbling scheme GS is obv2-secure with respect to Φ if for all PT adversaries $\mathcal{A}$ there exists a simulator Sim such that $\mathsf{Adv}^{\mathrm{obv2},\Phi,\mathrm{Sim}}_{\mathsf{GS},\mathcal{A}}(\lambda)$ is negligible. We let $\mathsf{OBV2}[\Phi]$ denote the set of all garbling schemes obv2-secure over Φ.

The third notion *authenticity* is defined through game $\mathrm{Obv2}_{\mathsf{GS},\Phi,\mathrm{Sim}}$ of Fig. 19. We define $\mathsf{Adv}^{\mathrm{aut2}}_{\mathsf{GS},\mathcal{A}}(\lambda) = \Pr[\mathrm{Aut2}^{\mathcal{A}}_{\mathsf{GS}}(\lambda)]$. Garbling scheme GS is aut2-secure if $\mathsf{Adv}^{\mathrm{aut2}}_{\mathsf{GS},\mathcal{A}}(\lambda)$ is negligible for any PT adversary $\mathcal{A}$. We let $\mathsf{AUT2}$ denote the set of all aut2 secure garbling schemes.

Results. Let H be a hash family such that $\mathsf{H.IL}(\lambda) = \mathbb{N}$ and $\mathsf{H.ol}(\lambda) = \lambda$ for every $\lambda \in \mathbb{N}$. Fig. 20 describes the $\mathsf{GaP}[\mathsf{H}]$ and $\mathsf{GaAO}[\mathsf{H}]$ garbling schemes, based on the RO-model scheme $\mathsf{Ga}[\mathsf{h}]$ of Pinkas, Schneider, Smart, and Williams [101] that depends on a keyless hash h such that $\mathsf{h}(x,1^\ell) \in \{0,1\}^\ell$ for every

$\underline{\mathsf{Ga}[\mathsf{h}].\mathsf{Gb}(1^\lambda, f)}$
$(n, m, q, A', B', G) \leftarrow f$
For $i \leftarrow 1$ to $n+q$ do
  $t \leftarrow\!\!\$\ \{0,1\}$; $X_i^0 \leftarrow\!\!\$\ \{0,1\}^{\lambda-1} t$; $X_i^1 \leftarrow\!\!\$\ \{0,1\}^{\lambda-1} \bar{t}$
For $g \leftarrow n+1$ to $n+q$, $i \leftarrow 0$ to 1, $j \leftarrow 0$ to 1 do
  $a \leftarrow A'(g)$; $b \leftarrow B'(g)$
  $A \leftarrow X_a^i$; $\mathsf{a} \leftarrow \mathrm{lsb}(A)$; $B \leftarrow X_b^j$; $\mathsf{b} \leftarrow \mathrm{lsb}(B)$
  $T[g, \mathsf{a}, \mathsf{b}] \leftarrow \mathsf{h}(A \,\|\, B \,\|\, g, 1^\lambda) \oplus X_g^{G_g(i,j)}$
$F \leftarrow (n, m, q, A', B', T)$
$e \leftarrow (X_1^0, X_1^1, \ldots, X_n^0, X_n^1)$
$d \leftarrow (X_{n+q-m+1}^0, X_{n+q-m+1}^1, \ldots, X_{n+q}^0, X_{n+q}^1)$
Return (F, e, d)

$\underline{\mathsf{Ga}[\mathsf{h}].\mathsf{Ev}(F, X)}$
$(n, m, q, A', B', T) \leftarrow F$, $(X_1, \ldots, X_n) \leftarrow X$
For $g \leftarrow n+1$ to $n+q$ do
  $a \leftarrow A'(g)$; $b \leftarrow B'(g)$
  $A \leftarrow X_a$; $\mathsf{a} \leftarrow \mathrm{lsb}(A)$; ; $B \leftarrow X_b$; $\mathsf{b} \leftarrow \mathrm{lsb}(B)$
  $X_g \leftarrow \mathsf{h}(X_a \,\|\, X_b \,\|\, g, 1^\lambda) \oplus T[g, \mathsf{a}, \mathsf{b}]$
Return $(X_{n+q-m+1}, \ldots, X_{n+q})$

$\underline{\mathsf{Ga}[\mathsf{h}].\mathsf{De}(d, Y)}$
$(Y_1, \ldots, Y_m) \leftarrow Y$; $(Y_1^0, Y_1^1, \ldots, Y_m^0, Y_m^1) \leftarrow d$
For $i \leftarrow 1$ to m do
  If $Y_i = Y_i^0$ then $y_i \leftarrow 0$
  Elsif $Y_i = Y_i^1$ then $y_i \leftarrow 1$ else return $\perp$
Return $y \leftarrow y_1 \cdots y_m$

$\underline{\mathsf{GaAO}[\mathsf{H}].\mathsf{Gb}(1^\lambda, f)}$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$(F, e, d) \leftarrow\!\!\$\ \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Gb}(1^\lambda, f)$
$(X_1^0, X_1^1, \ldots, X_n^0, X_n^1) \leftarrow e$
For $i \leftarrow 1$ to $n-1$ do $V_i \leftarrow\!\!\$\ \{0,1\}^{|hk|}$
$V_n \leftarrow V_1 \oplus \cdots \oplus V_{n-1} \oplus hk$
For $i \leftarrow 1$ to n do $T_i^0 \leftarrow (X_i^0, V_i)$; $T_i^1 \leftarrow (X_i^1, V_i)$
Return $\big(F, (T_1^0, T_1^1, \ldots, T_n^0, T_n^1), d\big)$

$\underline{\mathsf{GaAO}[\mathsf{H}].\mathsf{Ev}(F, X^*)}$
$\big((X_1, V_1), \ldots, (X_n, V_n)\big) \leftarrow X^*$
$X \leftarrow (X_1, \ldots, X_n)$
$hk \leftarrow V_1 \oplus \cdots V_n$
$Y \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Ev}(F, X)$
Return Y

$\underline{\mathsf{GaP}[\mathsf{H}].\mathsf{Gb}(1^\lambda, f)}$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$(F, e, d) \leftarrow\!\!\$\ \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Gb}(1^\lambda, f)$
$(X_1^0, X_1^1, \ldots, X_n^0, X_n^1) \leftarrow e$
$(Y_1^0, Y_1^1, \ldots, Y_m^0, Y_m^1) \leftarrow d$
For $i \leftarrow 1$ to m do $u_i \leftarrow \mathrm{lsb}(Y_i^0)$
$U \leftarrow u_1 \cdots u_m$; $K \leftarrow (U, hk)$
For $i \leftarrow 1$ to $n-1$ do $V_i \leftarrow\!\!\$\ \{0,1\}^{|K|}$
$V_n \leftarrow V_1 \oplus \cdots \oplus V_{n-1} \oplus K$
For $i \leftarrow 1$ to n do $T_i^0 \leftarrow (X_i^0, V_i)$; $T_i^1 \leftarrow (X_i^1, V_i)$
Return $\big(F, (T_1^0, T_1^1, \ldots, T_n^0, T_n^1), \varepsilon\big)$

$\underline{\mathsf{GaP}[\mathsf{H}].\mathsf{Ev}(F, X^*)}$
$\big((X_1, V_1), \ldots, (X_n, V_n)\big) \leftarrow X^*$
$X \leftarrow (X_1, \ldots, X_n)$
$(hk, U) \leftarrow V_1 \oplus \cdots V_n$
$Y \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Ev}(F, X)$
Return (Y, U)

$\underline{\mathsf{GaP}[\mathsf{H}].\mathsf{De}(d^*, Y^*)}$
$((Y_1, \ldots, Y_m), U) \leftarrow Y^*$; $u_1 \ldots u_m \leftarrow U$
For $i \leftarrow 1$ to m do $y_i \leftarrow \mathrm{lsb}(Y_i) \oplus u_i$
Return $y \leftarrow y_1 \cdots y_m$

Figure 20: **Top:** A conventional circuit-garbling scheme Ga. Here $\bar{t}$ denotes the complement bit of t. **Middle:** Scheme GaAO that achieves aut2 and obv2 security. **Bottom:** Scheme GaP that achieves prv2 security. We omit their encoding functions, as they are projective. The evaluation functions of these schemes are the canonical circuit evaluation cev.

$x \in \{0,1\}^*$ and $\ell \in \mathbb{N}$.

In scheme Ga, wires carry λ-bit tokens (strings) the last bit of each is the token's *type*. To garble a circuit, we begin selecting two tokens for each wire, one of each type. One of these will represent 0—the token is said to have *semantics* of 0—while the other will represent 1. The variable X_i^b names the token of wire i with semantics of b. For every wire i, we select random tokens of opposite type, making the association between a token's type and its semantics random. We then compute q garbled tables, one for each gate g. Table $T[g, \cdot, \cdot]$ has four rows, entry a, b the row to use when the left incoming token is of type a and the right incoming token is of type b. The token that gets encrypted for this row is the token for the outgoing-wire with the correct semantics. Given two tokens X_a and X_b we use their types to determine which entry of the garbled table we need to decrypt. The description of the decoding function d is the list of the tokens at output wires.

Informally, scheme GaAO generates a key hk of a UCE-secure hash H and runs $\mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Gb}$

$\underline{\textsc{Main}\ \mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda)}$	$\underline{\textsc{Main}\ \mathrm{mReset}^{R}_{S}(\lambda)}$
$(1^n, t) \leftarrow\!\!\$\ S^{\mathsf{RO}}(1^\lambda, \varepsilon)$	$\mathsf{Dom} \leftarrow \emptyset\ ;\ (1^n, t) \leftarrow\!\!\$\ S^{\mathsf{RO}}(1^\lambda, \varepsilon)$
For $i = 1$ to n do $\mathbf{hk}[i] \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$L \leftarrow\!\!\$\ S^{\mathsf{RO},\textsc{Hash}}(1^n, t)\ ;\ b \leftarrow\!\!\$\ \{0,1\}$
$b \leftarrow\!\!\$\ \{0,1\}\ ;\ L \leftarrow\!\!\$\ S^{\mathsf{RO},\textsc{Hash}}(1^n, t)$	If $b = 0$ then // reset the array T
$b' \leftarrow\!\!\$\ D^{\mathsf{RO}}(1^\lambda, \mathbf{hk}, L)$	For $(x, \ell, i) \in \mathsf{Dom}$ do
Return $(b' = b)$	$T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
$\underline{\textsc{Hash}(x, 1^\ell, i)}$	$b' \leftarrow\!\!\$\ R^{\mathsf{RO},\textsc{Hash}}(1^\lambda, L)\ ;$ Return $(b = b')$
If $T[x, \ell, i] = \perp$ then	$\underline{\textsc{Hash}(x, 1^\ell, i)}$
If $b = 1$ then $T[x, \ell, i] \leftarrow \mathsf{H.Ev}^{\mathsf{RO}}(1^\lambda, \mathbf{hk}[i], x, 1^\ell)$	$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{(x, \ell, i)\}$
Else $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$	If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell, i]$	Return $T[x, \ell, i]$
$\underline{\mathsf{RO}(v, 1^\ell)}$	$\underline{\mathsf{RO}(v, 1^\ell)}$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$	If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$	Return $H[v, \ell]$

Figure 21: **Left:** Game mUCE defines multi-key UCE security in the ROM. **Right:** Game mReset defines multi-key reset-security in the ROM.

to produce (F, e, d). We then secret-share the key hk to n shares, distributing the shares among the input tokens. Scheme GaP also creates a hash key hk and runs $\mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Gb}$ to generate (F, e, d). However, it instead uses a vacuous decoding function. Let $d = (Y_1^0, Y_1^1, \ldots, Y_m^0, Y_m^1)$, $u_i = \mathrm{lsb}(Y_i^0)$ for every $i \le m$, and $U = u_1 \ldots u_m$, where function lsb maps a string to its last bit. It then secret-shares (U, hk) to n shares and distribute them among the input tokens. Later, to decode the garbled output $Y = (Y_1, \ldots, Y_m)$ from procedure $\mathsf{Ga.Ev}$, it will output $y_1 \cdots y_m$ as the final output, where $y_i = \mathrm{lsb}(Y_i) \oplus u_i$ for every $i \le m$.

The following says that $\mathsf{GaP}[\mathsf{H}]$ is prv2-secure, and $\mathsf{GaAO}[\mathsf{H}]$ is obv2 and aut2-secure. The proof is in Appendix B.

Theorem 5.12 Let H be a hash function family such that $\mathsf{H.IL}(\lambda) = \mathbb{N}$ and $\mathsf{H.ol}(\lambda) = \lambda$ for every $\lambda \in \mathbb{N}$. If $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$, then (1) $\mathsf{GaAO}[\mathsf{H}] \in \mathsf{AUT2}$, (2) $\mathsf{GaAO}[\mathsf{H}] \in \mathsf{OBV2}[\Phi_{\mathrm{topo}}]$, and (3) $\mathsf{GaP}[\mathsf{H}] \in \mathsf{PRIV2}[\Phi_{\mathrm{topo}}]$.

6 Constructions of UCE families

In this section, we describe constructions of UCE-secure function families. We provide a ROM construction and prove that it is $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$-secure. This means is also secure under all other UCE notions that we considered: $\mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}], \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}], \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}], \mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}], \mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}], \mathsf{mUCE}[\mathcal{S}^{\text{cup-m}}], \mathsf{mUCE}[\mathcal{S}^{\text{srs-m}}]$, and supersets of these formed by further constraining the sources. We go on to explore practical instantiations of UCE-secure functions. We refer the reader to Section 2 for a discussion of the value of validating UCE in the ROM as part of the layered-cryptography approach for ROM-based design.

6.1 Achieving UCE in the ROM

A random oracle RO is a stateful algorithm that maintains a table H, initially empty. When invoked with inputs $(m, 1^\ell)$, it returns $H[m, \ell]$ if $H[m, \ell]$ is already defined. Otherwise it picks $y \leftarrow\!\!\$\ \{0,1\}^\ell$, sets $H[m, \ell]$ to y and then returns y. A game in the ROM would implement RO and present an interface to access RO for its routines as well as adversaries.

Definitions. The first step is to extend the syntax. In a ROM family of functions H, the algorithm $\mathsf{H.Ev}$ has oracle access to RO. The rest is as before. We now define $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$ security of H in the ROM. The multi-source S now has access to RO in addition to Hash, and the distinguisher gets access to RO as well. We continue to define m-uce advantage via Equation (4), with game $\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda)$ now being that of Fig. 21.

$\underline{\text{MAIN } \mathrm{G}_1^{S,D}(\lambda),\ \boxed{\mathrm{G}_2^{S,D}(\lambda)}}$
$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$; $M \leftarrow \emptyset$
For $i = 1$ to n do
  $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$
  If $\mathbf{k}[i] \in M$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda \backslash M}$
  $M \leftarrow M \cup \{\mathbf{k}[i]\}$
$L \leftarrow\!\!\$\ S^{\text{HASH},\mathsf{RO}_1}(1^n, t)$; $b \leftarrow\!\!\$\ D^{\mathsf{RO}_2}(1^\lambda, \mathbf{k}, L)$
Return $(b = 1)$

$\underline{\text{HASH}(x, 1^\ell, i)}$
$v \leftarrow \mathbf{k}[i] \,\|\, x$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\mathsf{RO}_1(v, 1^\ell)}$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\mathsf{RO}_2(v, 1^\ell)}$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\text{MAIN } \boxed{\mathrm{G}_3^{S,D}(\lambda)},\ \mathrm{G}_4^{S,D}(\lambda)}$
$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$; $M \leftarrow \emptyset$
For $i = 1$ to n do
  $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$
  If $\mathbf{k}[i] \in M$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda \backslash M}$
  $M \leftarrow M \cup \{\mathbf{k}[i]\}$
$L \leftarrow\!\!\$\ S^{\text{HASH},\mathsf{RO}_1}(1^n, t)$; $b \leftarrow\!\!\$\ D^{\mathsf{RO}_2}(1^\lambda, \mathbf{k}, L)$; Return $(b = 1)$

$\underline{\text{HASH}(x, 1^\ell, i)}$
If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell, i]$

$\underline{\mathsf{RO}_1(v, 1^\ell)}$
$x \leftarrow v[\lambda + 1, |v|]$; $K \leftarrow v[1, \lambda]$
For $i = 1$ to n do
  If $K = \mathbf{k}[i]$ then
    $\mathsf{coll} \leftarrow \mathsf{true}$
    If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
    Return $T[x, \ell, i]$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\mathsf{RO}_2(v, 1^\ell)}$
$x \leftarrow v[\lambda + 1, |v|]$; $K \leftarrow v[1, \lambda]$
For $i = 1$ to n do
  If $K = \mathbf{k}[i]$ then
    If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
    Return $T[x, \ell, i]$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

Figure 22: **Games G_1–G_4 for the proof of Theorem 6.1.** Games $\mathrm{G}_2, \mathrm{G}_3$ include the corresponding boxed statement, while the other games do not.

A reset adversary now gets oracle access to RO in addition to HASH. We say that S is computationally reset-secure if $\mathsf{Adv}^{\mathsf{m\text{-}reset}}_{S,R}(\cdot)$ is negligible for any PT reset adversary R, where $\mathsf{Adv}^{\mathsf{m\text{-}reset}}_{S,R}(\lambda) = 2\Pr[\mathrm{mReset}^S_R(\lambda)] - 1$ and game $\mathrm{mReset}^S_R(\lambda)$ is in Fig. 21. We say that H is $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$-secure in the ROM if $\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},S,D}(\cdot)$ is negligible for every PT reset-secure multi-source S and every PT D. Let $\mathsf{mUCE}^{\mathsf{ro}}[\mathcal{S}^{\text{crs-m}}]$ denote the set of all ROM function families H that are $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$-secure in the ROM.

RESULTS. We now describe a $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$-secure ROM family of functions. The construction H is as follows. Let $\mathsf{H.Kg}(1^\lambda)$ return $hk \leftarrow\!\!\$\ \{0,1\}^\lambda$ for every $\lambda \in \mathbb{N}$. Let $\mathsf{H.IL} = \mathbb{N}$ and $\mathsf{H.OL} = \mathbb{N}$. Let $\mathsf{H.Ev}^{\mathsf{RO}}(1^\lambda, hk, m, 1^\ell)$ return $\mathsf{RO}(hk \,\|\, m, 1^\ell)$ for every $hk \in \{0,1\}^\lambda$, every $m \in \{0,1\}^*$, every $\ell \in \mathbb{N}$ and every $\lambda \in \mathbb{N}$. The following says that H is $\mathsf{mUCE}[\mathcal{S}^{\text{crs-m}}]$-secure in the ROM.

Theorem 6.1 Let H be the ROM function family defined above. Then $\mathsf{H} \in \mathsf{mUCE}^{\mathsf{ro}}[\mathcal{S}^{\text{crs-m}}]$.

Proof: Let S be a PT, computationally reset-secure multi-source and let D be a PT distinguisher. Let $\overline{n}, q$ be polynomials such that $n \leq \overline{n}(\lambda)$ and S, D between them make at most $q(\lambda)$ RO-queries in game $\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda)$, for all $\lambda \in \mathbb{N}$. Assume $\overline{n}(\lambda) < 2^\lambda$ for all $\lambda \in \mathbb{N}$. Wlog, assume that S doesn't repeat a query to HASH or RO, and D does not repeat a query to RO. We'll construct a PT reset-adversary R such that

$\underline{\text{MAIN } \mathrm{G}_5^{S,D}(\lambda)}$
$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$
For $i = 1$ to n do $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$L \leftarrow\!\!\$\ S^{\text{HASH},\mathsf{RO}_1}(1^n, t)$; $b \leftarrow\!\!\$\ D^{\mathsf{RO}_2}(1^\lambda, \mathbf{k}, L)$
Return $(b = 1)$

$\underline{\text{HASH}(x, 1^\ell, i)}$
If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell, i]$

$\underline{\mathsf{RO}_1(v, 1^\ell)}$
$x \leftarrow v[\lambda + 1, |v|]$; $K \leftarrow v[1, \lambda]$
For $i = 1$ to n do
  If $K = \mathbf{k}[i]$ then $\mathsf{coll} \leftarrow \mathsf{true}$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\mathsf{RO}_2(v, 1^\ell)}$
$x \leftarrow v[\lambda + 1, |v|]$; $K \leftarrow v[1, \lambda]$
For $i = 1$ to n do
  If $K = \mathbf{k}[i]$ then
    If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
    Return $T[x, \ell, i]$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\text{MAIN } \mathrm{G}_6^{S,D}(\lambda),\ \boxed{\mathrm{G}_7^{S,D}(\lambda)}}$
$(1^n, t) \leftarrow\!\!\$\ S(1^\lambda, \varepsilon)$
For $i = 1$ to n do $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$L \leftarrow\!\!\$\ S^{\text{HASH},\mathsf{RO}_1}(1^n, t)$; $b \leftarrow\!\!\$\ D^{\mathsf{RO}_2}(1^\lambda, \mathbf{k}, L)$
Return $(b = 1)$

$\underline{\text{HASH}(x, 1^\ell, i)}$
If $T[x, \ell, i] = \perp$ then $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $T[x, \ell, i]$

$\underline{\mathsf{RO}_1(v, 1^\ell)}$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

$\underline{\mathsf{RO}_2(v, 1^\ell)}$
$x \leftarrow v[\lambda + 1, |v|]$; $K \leftarrow v[1, \lambda]$
For $i = 1$ to n do
  If $K = \mathbf{k}[i]$ then
    If $H[v, \ell] \neq \perp$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\text{Return } H[v, \ell]}$
    $T[x, \ell, i] \leftarrow\!\!\$\ \{0,1\}^\ell$; Return $T[x, \ell, i]$
If $H[v, \ell] = \perp$ then $H[v, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[v, \ell]$

Figure 23: **Games $\mathrm{G}_5, \mathrm{G}_6$, and G_7 for the proof of Theorem 6.1.** Game G_7 includes the corresponding boxed statement, while the other games do not.

for all $\lambda \in \mathbb{N}$ we have

$$\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{S,D,\mathsf{H}}(\lambda) \leq \mathsf{Adv}^{\mathsf{mreset}}_{S,R}(\lambda) + \frac{2\overline{n}(\lambda) \cdot q(\lambda) + \overline{n}(\lambda)^2}{2^\lambda} \ . \tag{20}$$

The theorem follows from the assumption that S is computationally reset secure.

Consider games G_1–G_7 in Fig. 22. We let RO_1 be the interface of S to access to the random oracle, and RO_2 be that of D. Let d be the challenge bit of game $\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\cdot)$. Then

$$\Pr[\mathrm{G}_1^{S,D}(\lambda)] = \Pr[\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda) \mid d = 1] \text{ and } \Pr[\mathrm{G}_7^{S,D}(\lambda)] = 1 - \Pr[\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda) \mid d = 0] \ .$$

We explain the game chain up to the terminal one. In game $\mathrm{G}_2^{S,D}(\lambda)$, we sample the hash keys so that they are distinct. The two games $\mathrm{G}_1^{S,D}(\lambda)$ and $\mathrm{G}_2^{S,D}(\lambda)$ are identical-until-bad. Then, for all $\lambda \in \mathbb{N}$, we have

$$\Pr[\mathrm{G}_1^{S,D}(\lambda)] - \Pr[\mathrm{G}_2^{S,D}(\lambda)] \leq \Pr[\mathrm{G}_2^{S,D}(\lambda) \text{ sets } \mathsf{bad}] \leq \frac{\overline{n}(\lambda)^2}{2^{\lambda+1}} \ .$$

In game $\mathrm{G}_3^{S,D}(\lambda)$, for each string v, if there is $i \leq n$ such that $v[1, \lambda] = \mathbf{k}[i]$, instead of reading/writing to $H[v, \ell]$, we'll use $T\big[v[\lambda + 1, |v|], \ell, i\big]$. Since the keys are distinct, for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathrm{G}_2^{S,D}(\lambda)] = \Pr[\mathrm{G}_3^{S,D}(\lambda)] \ .$$

In game $\mathrm{G}_4^{S,D}(\lambda)$, the keys now are sampled independently. The two games $\mathrm{G}_3^{S,D}(\lambda)$ and $\mathrm{G}_4^{S,D}(\lambda)$ are identical-until-bad. Then for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathrm{G}_3^{S,D}(\lambda)] - \Pr[\mathrm{G}_4^{S,D}(\lambda)] \leq \Pr[\mathrm{G}_4^{S,D}(\lambda) \text{ sets } \mathsf{bad}] \leq \frac{\overline{n}(\lambda)^2}{2^{\lambda+1}} \ .$$

In game $\mathrm{G}_5^{S,D}(\lambda)$, replies from RO_2 are no longer consistent with HASH replies. The two games $\mathrm{G}_4^{S,D}(\lambda)$ and $\mathrm{G}_5^{S,D}(\lambda)$ are identical-until-coll. Then for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathrm{G}_4^{S,D}(\lambda)] - \Pr[\mathrm{G}_5^{S,D}(\lambda)] \leq \Pr[\mathrm{G}_5^{S,D}(\lambda) \text{ sets } \mathsf{coll}] \leq \frac{\overline{n}(\lambda) \cdot q(\lambda)}{2^\lambda},$$

where the last inequality is due to the fact that the keys now are uniformly random, and independent of whatever S receives. In game $\mathrm{G}_6^{S,D}(\lambda)$, in procedure RO_2, we reset the entries of T before giving answers to D. Now consider the reset-adversary R constructed below:

$\underline{R^{\mathrm{HASH},\mathsf{RO}}(1^\lambda, 1^n, L)}$	$\underline{\mathsf{ROSim}(v, 1^\ell)}$
For $i = 1$ to n do $\mathbf{k}[i] \leftarrow\!\!\$\ \{0,1\}^\lambda$	$x \leftarrow v[\lambda+1, \lvert v\rvert]$; $K \leftarrow v[1,\lambda]$
$b \leftarrow\!\!\$\ D^{\mathsf{ROSim}}(1^\lambda, \mathbf{k}, L)$	For $i = 1$ to n do
Return b	If $K = \mathbf{k}[i]$ then then return $\mathrm{HASH}(x, 1^\ell, i)$
	Else return $\mathsf{RO}(v, 1^\ell)$

Let a be the challenge bit of game $\mathrm{mReset}_S^R(\lambda)$. Then for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathrm{G}_5^{S,D}(\lambda)] = \Pr[\,\mathrm{mReset}_S^P(\lambda) \,|\, a = 1\,] \text{ and } \Pr[\mathrm{G}_6^{S,D}(\lambda)] = 1 - \Pr[\,\mathrm{mReset}_S^P(\lambda) \,|\, a = 0\,] \ .$$

In game $\mathrm{G}_7^{S,D}(\lambda)$, replies from RO_1 and RO_2 are consistent. Games $\mathrm{G}_6^{S,D}(\lambda)$ and $\mathrm{G}_7^{S,D}(\lambda)$ are identical-until-bad. The flag bad is set only if S queries $(\mathbf{k}[i] \,\|\, x, 1^\ell)$ to RO_1, for some $i \leq n$. Then

$$\Pr[\mathrm{G}_6^{S,D}(\lambda)] - \Pr[\mathrm{G}_7^{S,D}(\lambda)] \leq \Pr[\mathrm{G}_7^{S,D}(\lambda) \text{ sets } \mathsf{bad}] \leq \frac{\overline{n}(\lambda) \cdot q(\lambda)}{2^\lambda} \ .$$

Hence, for every $\lambda \in \mathbb{N}$,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{S,D,\mathsf{H}}(\lambda) &= \Pr[\,\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda) \,|\, d = 1\,] + \Pr[\,\mathrm{mUCE}^{S,D}_{\mathsf{H}}(\lambda) \,|\, d = 0]\,] - 1 \\
&= \Pr[\mathrm{G}_1^{S,D}(\lambda)] - \Pr[\mathrm{G}_7^{S,D}(\lambda)] \\
&\leq \sum_{i=1}^{6} \Pr[\mathrm{G}_i^{S,D}(\lambda)] - \Pr[\mathrm{G}_{i+1}^{S,D}(\lambda)] \\
&\leq \mathsf{Adv}^{\mathsf{mreset}}_{S,R}(\lambda) + \frac{2\overline{n}(\lambda) \cdot q(\lambda) + \overline{n}(\lambda)^2}{2^\lambda}
\end{aligned}$$

yielding Equation (20). ∎

6.2 Practical constructions

We consider practical, heuristic instantiations for families of functions assumed UCE secure. The UCE framework and security definitions are asymptotic, while these real-world instantiations are based on non-asymptotic blockciphers and hash functions, so we make no formal claims about security. We ignore the security parameter and consider FOL families, so that we view $\mathsf{H.Kg}$ as taking no inputs and we view $\mathsf{H.Ev}$ as taking only a key and an input.

A natural instantiation is via a block cipher, for example AES. Here, $\mathsf{H.Kg}$ would pick a random 128-bit string K, and $\mathsf{H.Ev}(K, X)$ would return $\mathsf{AES}(K, X)$. However, as we saw in part (2) of the proof of Proposition 4.4, this construction fails to provide $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$ security since AES is efficiently invertible given the key and this can be exploited to mount an attack. This is interesting in the light of the fact that it is standard to use AES as a PRF or PRP.

One could consider instantiations based on cryptographic hash functions such as $\mathsf{SHA256}$, but UCE security requires a keyed function, and $\mathsf{SHA256}$ is not keyed. This suggests that we use the HMAC construction of [14, 93]. This is indeed our leading suggestion for a practical way to instantiate families assumed UCE secure, for example for $\mathsf{mUCE}[\mathcal{S}]$ where $\mathcal{S}$ is $\mathcal{S}^{\text{sup-m}}, \mathcal{S}^{\text{srs-m}}$ or subsets of these, or for $\mathsf{UCE}[\mathcal{S}^{\mathrm{splt}}]$. However, in the case that $\mathcal{S}$ is $\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{\tau,\sigma,q}$, the larger the values of τ, σ, the less recommended is HMAC as an

instantiation. The issue is that HMAC is efficient, and once τ, σ are large enough to create circuits for indistinguishability obfuscation (iO) of HMAC, security will fail due to the attack of [45].

An interesting open question is whether the assumption that HMAC provides (say) $\mathsf{mUCE}[\mathcal{S}^{\text{sup-m}}]$-security can be validated in an idealized model where one assumes the compression function is ideal. (If not, the suggestion that it be used to instantiate UCE families in practice should be reconsidered.) Since we have provided in Section 6.1 a ROM-based construct, one might hope to validate HMAC based on its indifferentiability from a RO [62], but as per [103] indifferentiability may not be enough because UCE is a multi-stage game, so a different approach or a direct analysis may be needed.

Acknowledgments

We thank the Crypto 2013 PC for their many valuable comments and suggestions. We thank Dan Boneh, Adam O'Neill, Phillip Rogaway, Ananth Raghunathan and Amit Sahai for their comments. We thank Peter Gaži for pointing out a bug in the proof of Theorem 6.1 in a prior version of this paper. We thank Christina Brzuska, Pooya Farshim and Arno Mittelbach for their personal communication [45].

References

[1] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev. Message-locked encryption for lock-dependent messages. In R. Canetti and J. Garay, editors, *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 374–391. Springer Berlin Heidelberg, 2013. 28

[2] T. Acar, M. Belenkiy, M. Bellare, and D. Cash. Cryptographic agility and its relation to circular encryption. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 403–422. Springer, May 2010. 30

[3] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, Mar. 2009. 22

[4] B. Applebaum. Key-dependent message security: Generic amplification and completeness. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, May 2011. 6, 30

[5] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Aug. 2009. 6, 30

[6] B. Applebaum, D. Harnik, and Y. Ishai. Semantic security under related-key attacks and applications. In A. Yao, editor, *ICS 2011*. Tsinghua University Press, 2011. 6, 31

[7] B. Applebaum, Y. Ishai, E. Kushilevitz, and B. Waters. Encoding functions with constant online rate or how to compress garbled circuits keys. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 166–184. Springer Berlin Heidelberg, 2013. 41

[8] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. Song. Provable data possession at untrusted stores. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 598–609. ACM Press, Oct. 2007. 38

[9] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Aug. 2001. 4, 9, 12

[10] B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 423–444. Springer, May 2010. 6, 30

[11] M. Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619. Springer, Aug. 2006. 7, 14

[12] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Aug. 2007. 4, 5, 8, 24

[13] M. Bellare, A. Boldyreva, and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 171–188. Springer, May 2004. 3

[14] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Aug. 1996. 7, 9, 14, 47

[15] M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 360–378. Springer, Aug. 2008. 5, 9, 24

[16] M. Bellare, V. Hoang, and P. Rogaway. Adaptively secure garbling with applications to one-time programs and secure outsourcing. In *ASIACRYPT 2012*, LNCS, pages 134–153. Springer, Dec. 2012. 6, 40, 41, 42

[17] M. Bellare, V. Hoang, and P. Rogaway. Foundations of garbled circuits. In *ACM Computer and Communications Security (CCS'12)*. ACM, 2012. 40, 41

[18] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via uces. In *Advances in Cryptology–CRYPTO 2013*, pages 398–415. Springer, 2013. Full paper available as Report 2013/424 on the IACR Cryptology ePrint Archive. 4, 12, 13

[19] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology–EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 296–312. Springer, 2013. 5, 26, 27

[20] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, May 2003. 31

[21] M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 401–418. Springer, May 2004. 31

[22] M. Bellare, K. Paterson, and S. Thomson. RKA Security beyond the Linear Barrier: IBE, Encryption and Signatures. In X. Wang and K. Sako, editors, *Advances in Cryptology–ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 331–348. Springer, 2012. 6, 31

[23] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. 3, 5, 23

[24] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, May 1994. 4, 5, 6, 32

[25] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, May 1996. 8, 9

[26] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. 10, 15, 17, 34, 53

[27] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Aug. 2003. 5, 30

[28] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. 21, 22

[29] A. Boldyreva, D. Cash, M. Fischlin, and B. Warinschi. Foundations of non-malleable hash and one-way functions. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 524–541. Springer, Dec. 2009. 4

[30] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Aug. 2008. 5, 9, 24

[31] A. Boldyreva and M. Fischlin. Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 412–429. Springer, Aug. 2005. 4, 6

[32] A. Boldyreva and M. Fischlin. On the security of OAEP. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 210–225. Springer, Dec. 2006. 4, 6

[33] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Aug. 2004. 9

[34] D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr. 2008. 7

[35] D. Boneh and X. Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, Oct. 2011. 6, 39

[36] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, May 2005. 7

[37] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Aug. 2004. 7

[38] D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. 7

[39] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Aug. 2005. 7

[40] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 325–341. Springer, Feb. 2005. 7

[41] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision diffie-hellman. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Aug. 2008. 6, 30

[42] Z. Brakerski and G. N. Rothblum. Obfuscating conjunctions. In *Advances in Cryptology–CRYPTO 2013*, pages 416–434. Springer, 2013. 13

[43] Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer, Aug. 2011. 5, 9, 24, 26

[44] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Aug. 2011. 30

[45] C. Brzuska, P. Farshim, and A. Mittelbach. Personal communication, Sept. 2013. 4, 9, 12, 13, 14, 48

[46] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, May 1999. 6

[47] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469. Springer, Aug. 1997. 4

[48] R. Canetti and R. R. Dakdouk. Extractable perfectly one-way functions. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 449–460. Springer, July 2008. 4

[49] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998. 3, 8, 14

[50] R. Canetti, O. Goldreich, and S. Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 40–57. Springer, Feb. 2004. 3

[51] R. Canetti, Y. T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, Feb. 2010. 29

[52] R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th ACM STOC*, pages 131–140. ACM Press, May 1998. 4

[53] R. Canetti and V. Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. Technical report, Cryptology ePrint Archive, 2013. 13

[54] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *26th FOCS*, pages 429–442. IEEE Computer Society Press, Oct. 1985. 9, 24

[55] J.-S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, Aug. 2000. 9

[56] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård revisited: How to construct a hash function. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Aug. 2005. 6, 9

[57] D. Dachman-Soled, R. Gennaro, H. Krawczyk, and T. Malkin. Computational extractors and pseudorandomness. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 383–403. Springer, Mar. 2012. 8, 14

[58] G. Demay, P. Gaži, M. Hirt, and U. Maurer. Resource-restricted indifferentiability. In *Advances in Cryptology–EUROCRYPT 2013*, pages 664–683. Springer, 2013. 9

[59] A. W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Y. Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Dec. 2002. 8

[60] Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of the full domain hash. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 449–466. Springer, Aug. 2005. 3, 8, 9

[61] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. 14

[62] Y. Dodis, T. Ristenpart, J. P. Steinberger, and S. Tessaro. To hash or not to hash again? (in)differentiability results for h^2 and HMAC. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366. Springer, Aug. 2012. 9, 48

[63] J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 617–624. IEEE, 2002. 5, 26, 27

[64] M. Fischlin. A note on security proofs in the generic model. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 458–469. Springer, Dec. 2000. 8

[65] P.-A. Fouque, D. Pointcheval, and S. Zimmer. HMAC is a randomness extractor and applications to TLS. In M. Abe and V. Gligor, editors, *ASIACCS 08*, pages 21–32. ACM Press, Mar. 2008. 8, 14

[66] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. *Journal of Cryptology*, 17(2):81–104, Mar. 2004. 6, 32

[67] B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 582–599. Springer, Mar. 2012. 5, 9

[68] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *Proc. of FOCS (to appear)*, 2013. 4, 9, 12, 13

[69] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 467–476. ACM, 2013. 13

[70] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Aug. 2010. 6, 41, 42

[71] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 123–139. Springer, May 1999. 9

[72] C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. 41

[73] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986. 3

[74] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989. 22

[75] S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, Oct. 2003. 3

[76] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. One-time programs. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56. Springer, Aug. 2008. 6, 41, 42

[77] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. 21, 22

[78] V. Goyal, Y. Ishai, A. Sahai, R. Venkatesan, and A. Wadia. Founding cryptography on tamper-proof hardware tokens. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 308–326. Springer, Feb. 2010. 6, 41

[79] V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Mar. 2011. 5, 6, 39

[80] J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, May / June 2006. 7

[81] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 8

[82] J. Håstad and M. Näslund. The security of individual RSA bits. In *39th FOCS*, pages 510–521. IEEE Computer Society Press, Nov. 1998. 22

[83] D. Hofheinz. Possibility and impossibility results for selective decommitments. *Journal of Cryptology*, 24(3):470–516, July 2011. 8

[84] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38. Springer, Aug. 2008. 9

[85] S. Hohenberger, A. Sahai, and B. Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In *Advances in Cryptology–CRYPTO 2013*, pages 494–512. Springer, 2013. 8

[86] C.-Y. Hsiao, C.-J. Lu, and L. Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In M. Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 169–186. Springer, May 2007. 8

[87] A. Juels and B. S. Kaliski Jr. Pors: proofs of retrievability for large files. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 584–597. ACM Press, Oct. 2007. 38

[88] S. A. Kakvi and E. Kiltz. Optimal security proofs for full domain hash, revisited. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 537–553. Springer, Apr. 2012. 9

[89] E. Kiltz, A. O'Neill, and A. Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313. Springer, Aug. 2010. 4, 6, 32

[90] E. Kiltz and K. Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 389–406. Springer, Apr. 2009. 3

[91] R. Kotla, L. Alvisi, and M. Dahlin. Safestore: A durable and practical storage system. In *Usenix Technical 2007*, pages 331–348. USENIX, 2007. 38

[92] H. Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 631–648. Springer, Aug. 2010. 8, 14

[93] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. IETF Internet Request for Comments 2104, Feb. 1997. 9, 47

[94] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988. 17

[95] B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, May 2004. 5, 28, 29

[96] T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526. Springer, May 2011. 6, 30

[97] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Feb. 2004. 3, 6, 8, 9, 38

[98] J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, Aug. 2002. 3

[99] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996. 14

[100] R. Pass. Limits of provable security from standard assumptions. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 109–118. ACM Press, June 2011. 8

[101] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure two-party computation is practical. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 250–267. Springer, Dec. 2009. 6, 42

[102] A. Raghunathan, G. Segev, and S. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology–EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 93–110. Springer, 2013. 26

[103] T. Ristenpart, H. Shacham, and T. Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer, May 2011. 5, 6, 9, 10, 38, 48

[104] H. Shacham and B. Waters. Compact proofs of retrievability. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 90–107. Springer, Dec. 2008. 38

[105] D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In *ITCS 2013*, 2013. 8

[106] A. C. Yao. Protocols for secure computations. In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, Nov. 1982. 40

[107] A. C. Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, Nov. 1982. 21, 22

A Proof of Proposition 4.2

Proof of Proposition 4.2: For part (1), let $\mathcal{P}^{\mathsf{stat}}$ and $\mathcal{R}^{\mathsf{stat}}$ denote the classes of all predictors and reset adversaries respectively. Recall that $\mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}] = \mathsf{UCE}[\mathcal{S}^{\mathsf{poly}} \cap \mathsf{Pred}[\mathcal{P}^{\mathsf{stat}}], \mathcal{D}^{\mathsf{poly}}]$ and $\mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}] = \mathsf{UCE}[\mathcal{S}^{\mathsf{poly}} \cap \mathsf{Reset}[\mathcal{R}^{\mathsf{stat}}], \mathcal{D}^{\mathsf{poly}}]$. By Proposition 4.1 it suffices to show that $\mathsf{Pred}[\mathcal{P}^{\mathsf{stat}}] \subseteq \mathsf{Reset}[\mathcal{R}^{\mathsf{stat}}]$. In other words, we want to show that every statistically unpredictable source S is also statistically reset secure. Intuitively, this is because a reset adversary will be unable to query its oracle at any point where the source queried its oracle, except with negligible probability, and hence will usually have the same view both when its oracle is reset and when it is not. Formally, given a reset adversary R we build a predictor P such that $\mathsf{Adv}^{\mathsf{reset}}_{R,S}(\cdot) \leq \mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot)$. Predictor P is on the left below:

$\underline{P^{\mathrm{HASH}}(1^\lambda, L)}$	$\underline{\text{MAIN } \mathrm{G}_0^{S,R}(\lambda) \ / \ \boxed{\mathrm{G}_1^{S,R}(\lambda)}}$
$Q' \leftarrow \emptyset \ ; \ b' \leftarrow\!\!\$ \ R^{\mathrm{HASHSIM}}(1^\lambda, L)$	$Q \leftarrow \emptyset \ ; \ L \leftarrow\!\!\$ \ S^{\mathrm{HASH}}(1^\lambda) \ ; \ b' \leftarrow R^{\mathrm{HASH}}(1^\lambda, L)$
Return Q'	Return $(b' = 1)$
$\underline{\mathrm{HASHSIM}(x, \ell)}$	$\underline{\mathrm{HASH}(x, \ell)}$
$Q' \leftarrow Q' \cup \{x\} \ ; \ T[x, \ell] \leftarrow\!\!\$ \ \{0,1\}^\ell$	If $T[x, \ell] = \perp$ then $T[x, \ell] \leftarrow\!\!\$ \ \{0,1\}^\ell$
Return $T[x, \ell]$	Else $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T[x, \ell] \leftarrow\!\!\$ \ \{0,1\}^\ell}$
	Return $T[x, \ell]$

For the analysis, assume neither S nor R repeat an oracle query, and consider games $\mathrm{G}_0^{S,R}(\lambda), \mathrm{G}_1^{S,R}(\lambda)$ above, where $\mathrm{G}_1^{S,R}(\lambda)$ includes the boxed code and $\mathrm{G}_0^{S,R}(\lambda)$ does not. The games are identical-until-bad, so by the Fundamental Lemma of Game Playing [26],

$$\mathsf{Adv}^{\mathsf{reset}}_{R,S}(\cdot) = \Pr[\mathrm{G}_0^{S,R}(\cdot)] - \Pr[\mathrm{G}_1^{S,R}(\cdot)] \leq \Pr[\mathrm{G}_1^{S,R}(\cdot) \text{ sets } \mathsf{bad}] \leq \mathsf{Adv}^{\mathsf{pred}}_{S,P}(\cdot)$$

as desired. Moreover, P makes no HASH queries, and the size of the set Q' returned by P is equal to the number of HASHSIM queries made by R, which in turn is bounded by a polynomial. We also have that $\mathsf{UCE}[\mathcal{S}^{\mathrm{crs}}] \subseteq \mathsf{UCE}[\mathcal{S}^{\mathrm{cup}}]$ by noting that if R is a PT algorithm, then so is P.

For part (2), consider $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$. Define a hash function family $\overline{\mathsf{H}}$ as follows: $\overline{\mathsf{H}}.\mathsf{Kg} = \mathsf{H}.\mathsf{Kg}$ and $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, 0^\lambda, 1^\lambda) = 0^\lambda$, and $\overline{\mathsf{H}}.\mathsf{Ev}(1^\lambda, hk, x, 1^\ell) = \mathsf{H}.\mathsf{Ev}(1^\lambda, hk, x, 1^\ell)$ otherwise. It follows that $\overline{\mathsf{H}} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{sup}}]$, as an unpredictable source cannot query $\mathrm{HASH}(0^\lambda, 1^\lambda)$. However, $\overline{\mathsf{H}} \notin \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$. A source S can get $y = \mathrm{HASH}(0^\lambda, 1^\lambda)$, and leak 1 if $y = 0^\lambda$, and leak 0 otherwise. The distinguisher simply echoes the leakage. We claim that the source above is reset-secure. Let R be an arbitrary reset adversary.

In game $\mathrm{Reset}_S^R(\lambda)$, the leakage is 0 with probability $1-2^{-\lambda}$, regardless of the challenge bit. Moreover, when the challenge bit is 0, if R queries $(0^\lambda, 1^\lambda)$ to HASH, the chance that it gets 0^λ is at most $2^{-\lambda}$. Hence $\mathsf{Adv}^{\mathsf{reset}}_{R,S}(\lambda) \le 2^{1-\lambda}$. ∎

B Proof of Theorem 5.12

For part (1), consider an arbitrary PT adversary $\mathcal{A}$ attacking the aut2 security of $\mathsf{GaAO}[\mathsf{H}]$. Assume that $\mathcal{A}(1^\lambda)$ uses $\rho(\lambda)$ coins. We'll construct a PT statistically reset-secure source S and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathrm{aut2}}_{\mathsf{GaAO}[\mathsf{H}],\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\lambda) + 2^{1-\lambda} \tag{21}$$

for every $\lambda \in \mathbb{N}$. The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$. The constructions of S and D are shown below.

$\underline{S^{\mathrm{HASH}}(1^\lambda)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \bot$
$\mathcal{A}^{\mathrm{GARBLE},\mathrm{INPUT}}(1^\lambda; r)$; Return L

$\underline{\mathrm{GARBLE}(f)}$
$(F, (X_1^0, X_1^1, \ldots, X_n^0, X_n^1), d) \leftarrow\!\!\$\ \mathsf{Ga}[\mathrm{HASH}].\mathsf{Gb}(1^\lambda, f)$
Return F

$\underline{\mathrm{INPUT}(i, c)}$
$\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i \leftarrow X_i^c$; $V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$
If $\mathsf{cnt} < n$ then return (X_i, V_i)
$V_i \leftarrow \bot$; $X \leftarrow (X_1, \ldots, X_n)$; $V \leftarrow (V_1, \ldots, V_n)$
$(Y_1^0, Y_1^1, \ldots, Y_m^0, Y_m^1) \leftarrow d$
For $j = 1$ to m do $d_j \leftarrow Y_j^{1-y_j}[1, \lambda - 1]$
$L \leftarrow (F, r, X, V, (d_1, \ldots, d_m))$; Return $\bot$

$\underline{D(1^\lambda, hk, L)}$
$(F, r, X, V, (d_1, \ldots, d_m)) \leftarrow L$; $(V_1, \ldots, V_n) \leftarrow V$
$(Y_1, \ldots, Y_m) \leftarrow Y \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Ev}(F, X)$
$(X_1, \ldots, X_n) \leftarrow X$; $Y' \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{GARBLE},\mathrm{INPUT}}(1^\lambda; r)$
$x \leftarrow x_1 \cdots x_n$; $y_1 \cdots y_m \leftarrow y \leftarrow \mathsf{cev}(f, x)$
For $j = 1$ to m do
  $t_j \leftarrow \mathrm{lsb}(Y_j)$; $Y_j^{y_j} \leftarrow Y_j$; $Y_j^{1-y_j} \leftarrow d_j \,\|\, (1 - t_j)$
$d \leftarrow (Y_1^0, Y_1^1, \ldots, Y_m^0, Y_m^1)$
$y' \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{De}(d, Y')$
If $y' \neq \bot$ and $Y' \neq Y$ then return 1 else return 0

$\underline{\mathrm{GARBLE}(f)}$
Return F

$\underline{\mathrm{INPUT}(i, c)}$
If $V_i = \bot$ then $V_i \leftarrow 0^{\mathsf{H.kl}(\lambda)}$; $V_i \leftarrow V_1 \oplus \cdots \oplus V_n \oplus hk$
$x_i \leftarrow c$; Return (X_i, V_i)

Let a be the challenge bit of game $\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda)$. Then

$$\mathsf{Adv}^{\mathrm{aut2},\mathcal{A}}_{\mathsf{GaAO}[\mathsf{H}]}(\lambda) = \Pr[\,\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda) \,|\, a = 1\,] \ .$$

On the other hand, in game $\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda)$ with $a = 0$, only if $\mathcal{A}$ can specify some $j \le m$ and the correct d_j does D output 1. However, since each d_j is a uniformly random string that is independent of whatever $\mathcal{A}$ receives, $\Pr[\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda) \mid a = 0] \ge 1 - 2^{-\lambda}$. Hence

$$\begin{aligned} \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\lambda) &= \Pr[\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda) \mid a = 1] + \Pr[\mathrm{UCE}_{\mathsf{H}}^{S,D}(\lambda) \mid a = 0] - 1 \\ &\ge \mathsf{Adv}^{\mathrm{aut2}}_{\mathsf{GaAO}[\mathsf{H}],\mathcal{A}}(\lambda) - 2^{-\lambda}, \end{aligned}$$

yielding Equation (21). Finally, Lemma B.1 below shows that S is statistically reset-secure.

Lemma B.1 For any reset-adversary R and any polynomial p that bounds the number of R's HASH queries, $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\lambda) \le 2p(\lambda)/2^\lambda$ for every $\lambda \in \mathbb{N}$.

Proof: Consider games G_1–G_4 in Fig. 24. Let c be the challenge bit of game $\mathrm{Reset}_S^P(\lambda)$. Then

$$\Pr[\mathrm{G}_1^{\mathcal{A},R}(\lambda)] = \Pr[\,\mathrm{Reset}_S^P(\lambda) \,|\, c = 1\,] \text{ and } \Pr[\mathrm{G}_7^{\mathcal{A},R}(\lambda)] = 1 - \Pr[\,\mathrm{Reset}_S^P(\lambda) \,|\, c = 0\,]$$

for every $\lambda \in \mathbb{N}$. We explain the game chains up until the terminal game. In game $\mathrm{G}_2^{\mathcal{A},R}(\lambda)$, we postpone sampling the tokens and defining $4q$ points of the array H until they are needed. Hence

$$\Pr[\mathrm{G}_1^{\mathcal{A},R}(\lambda)] = \Pr[\mathrm{G}_2^{\mathcal{A},R}(\lambda)] \ .$$

$\underline{\text{MAIN } \mathrm{G}_1^{\mathcal{A},R}(\lambda)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \bot$
$\mathcal{A}^{\text{GARBLE},\text{INPUT}}(1^\lambda)$; $b' \leftarrow R^{\text{HASH}}(1^\lambda, L)$
Return $(b' = 1)$

$\underline{\text{INPUT}(i, c)}$
If $i > n$ or $x_i \neq \bot$ then return $\bot$
$V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i \leftarrow X_i^c$
If $\mathsf{cnt} < n$ then return (X_i, V_i)
For $j \leftarrow 1$ to m do $d_j \leftarrow\!\!\$\ \{0,1\}^{\lambda-1}$
$V_i \leftarrow \bot$; $V \leftarrow (V_1, \ldots, V_n)$
$X \leftarrow (X_1, \ldots, X_n)$
$L \leftarrow (F, r, X, V, (d_1, \ldots, d_m))$
Return $\bot$

$\underline{\text{GARBLE}(1^\lambda, f)}$
$(n, m, q, A', B', G) \leftarrow f$; $\mathsf{cnt} \leftarrow 0$
For $i \leftarrow 1$ to $n + q$ do
  $t \leftarrow\!\!\$\ \{0,1\}$; $X_i^0 \leftarrow\!\!\$\ \{0,1\}^{\lambda-1} t$; $X_i^1 \leftarrow\!\!\$\ \{0,1\}^{\lambda-1} \bar{t}$
For $g \leftarrow n + 1$ to $n + q$, $i \leftarrow 0$ to 1, $j \leftarrow 0$ to 1 do
  $a \leftarrow A'(g)$; $b \leftarrow B'(g)$
  $A \leftarrow X_a^i$; $\mathsf{a} \leftarrow \mathrm{lsb}(A)$; $B \leftarrow X_b^j$; $\mathsf{b} \leftarrow \mathrm{lsb}(B)$
  $T[g, \mathsf{a}, \mathsf{b}] \leftarrow\!\!\$\ \{0,1\}^\lambda$; $H[A \| B \| g, \lambda] \leftarrow T[g, \mathsf{a}, \mathsf{b}] \oplus X_g^{G_g(i,j)}$
$F \leftarrow (n, m, q, A', B', T)$
Return F

$\underline{\text{HASH}(w, 1^\ell)}$
If $H[w, \ell] = \bot$ then $H[w, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[w, \ell]$

$\underline{\text{MAIN } \boxed{\mathrm{G}_2^{\mathcal{A},R}(\lambda)}, \mathrm{G}_3^{\mathcal{A},R}(\lambda)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \bot$
$\mathcal{A}^{\text{GARBLE},\text{INPUT}}(1^\lambda)$; $b' \leftarrow R^{\text{HASH}}(1^\lambda, L)$
Return $(b' = 1)$

$\underline{\text{INPUT}(i, c)}$
If $i > n$ or $x_i \neq \bot$ then return $\bot$
$x_i \leftarrow c$; $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i^{x_i} \leftarrow X_i \leftarrow\!\!\$\ \{0,1\}^\lambda$
If $\mathsf{cnt} < n$ then
  $V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; Return (X_i, V_i)
For $g \leftarrow n + 1$ to $n + q$ do
  $a \leftarrow A'(g)$; $b \leftarrow B'(g)$; $x_g \leftarrow G_g(x_a, x_b)$
  $Z \leftarrow H[X_a \| X_b \| g, \lambda] \leftarrow\!\!\$\ \{0,1\}^\lambda$
  $X_g^{x_g} \leftarrow X_g \leftarrow Z \oplus T[g, x_a, x_b]$
For $j \leftarrow 1$ to $n + q - m$ do
  $t \leftarrow \mathrm{lsb}(X_j)$; $X_j^{1-x_j} \leftarrow\!\!\$\ \{0,1\}^{\lambda-1} \| \bar{t}$
For $j \leftarrow 1$ to m do
  $g \leftarrow n + q - m + j$; $d_j \leftarrow\!\!\$\ \{0,1\}^{\lambda-1}$
  $t \leftarrow \mathrm{lsb}(X_g)$; $X_g^{1-x_g} \leftarrow d_j \| \bar{t}$
$V_i \leftarrow \bot$; $V \leftarrow (V_1, \ldots, V_n)$
$X \leftarrow (X_1, \ldots, X_n)$
$L \leftarrow (F, r, X, V, (d_1, \ldots, d_m))$; Return $\bot$

$\underline{\text{GARBLE}(1^\lambda, f)}$
$(n, m, q, A', B', G) \leftarrow f$; $\mathsf{cnt} \leftarrow 0$
For $g \leftarrow n + 1$ to $n + q$, $\mathsf{a} \leftarrow 0$ to 1, $\mathsf{b} \leftarrow 0$ to 1 do
  $T[g, \mathsf{a}, \mathsf{b}] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$F \leftarrow (n, m, q, A', B', T)$
Return F

$\underline{\text{HASH}(w, 1^\ell)}$
If $H[w, \ell] \neq \bot$ then return $H[w, \ell]$
$H[w, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
If $w = A \| B \| g$ and $\ell = \lambda$ then
  $a \leftarrow A'(g)$; $b \leftarrow B'(g)$; $\mathsf{a} \leftarrow \mathrm{lsb}(A)$; $\mathsf{b} \leftarrow \mathrm{lsb}(B)$
  If $A = X_a^i$ and $B = X_b^j$ then
    $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{H[w, \ell] \leftarrow T[g, \mathsf{a}, \mathsf{b}] \oplus X_g^{G_g(i,j)}}$
Return $H[w, \ell]$

$\underline{\text{MAIN } \mathrm{G}_4^{\mathcal{A},R}(\lambda)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \bot$
$\mathcal{A}^{\text{GARBLE},\text{INPUT}}(1^\lambda)$; $b' \leftarrow R^{\text{HASH}}(1^\lambda, L)$
Return $(b' = 1)$

$\underline{\text{INPUT}(i, c)}$
If $i > n$ or $x_i \neq \bot$ then return $\bot$
$x_i \leftarrow c$; $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i \leftarrow\!\!\$\ \{0,1\}^\lambda$
If $\mathsf{cnt} < n$ then $V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; Return (X_i, V_i)
For $j \leftarrow 1$ to m do $d_j \leftarrow\!\!\$\ \{0,1\}^{\lambda-1}$
$V_i \leftarrow \bot$; $V \leftarrow (V_1, \ldots, V_n)$; $X \leftarrow (X_1, \ldots, X_n)$
$L \leftarrow (F, r, X, V, (d_1, \ldots, d_m))$; Return $\bot$

$\underline{\text{GARBLE}(1^\lambda, f)}$
$(n, m, q, A', B', G) \leftarrow f$; $\mathsf{cnt} \leftarrow 0$
For $g \leftarrow n + 1$ to $n + q$, $\mathsf{a} \leftarrow 0$ to 1, $\mathsf{b} \leftarrow 0$ to 1 do
  $T[g, \mathsf{a}, \mathsf{b}] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$F \leftarrow (n, m, q, A', B', T)$
Return F

$\underline{\text{HASH}(w, 1^\ell)}$
If $H[w, \ell] = \bot$ then $H[w, \ell] \leftarrow\!\!\$\ \{0,1\}^\ell$
Return $H[w, \ell]$

Figure 24: **Games for the proof of Lemma B.1.** Game G_2 contains the boxed statement but game G_3 does not.

$\underline{\mathrm{Sim}(1^\lambda, \phi, 0)}$	$\underline{\mathrm{Sim}(1^\lambda, i, \mathsf{cnt})}$
$(n, m, q, A, B) \leftarrow \phi$	$X_i \leftarrow\!\!\$\ \{0,1\}^\lambda$; $V_i \leftarrow 0^{\mathsf{H.kl}(\lambda)}$
For $g \leftarrow n+1$ to $n+q$, $\mathtt{a} \leftarrow 0$ to 1, $\mathtt{b} \leftarrow 0$ to 1 do	If $\mathsf{cnt} < n$ then $V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$
$\quad T[g, \mathtt{a}, \mathtt{b}] \leftarrow\!\!\$\ \{0,1\}^\lambda$	Else $V_i \leftarrow V_1 \oplus \cdots \oplus V_n \oplus hk$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$, $F \leftarrow (n, m, q, A, B, T)$; Return F	Return (X_i, V_i)

Figure 25: **Constructed simulator for part (2) of Theorem 5.12.**

After last INPUT query, one can run $\mathsf{Ga}[\textsc{Hash}(\cdot, 1^\lambda)].\mathsf{Ev}(F, X)$ to obtain a token per wire. We say that these tokens are *visible*, and the other tokens are *invisible*. A query $(w, 1^\lambda)$ is *illegitimate* if (i) $w = A \,\|\, B \,\|\, g$, (ii) A and B are tokens of the left- and right-incoming wires of g respectively, and (iii) at least one of A and B is invisible. In game $\mathrm{G}_3^{\mathcal{A},R}(\lambda)$, for illegitimate queries, procedure HASH gives answers independent of GARBLE and INPUT. The two games $\mathrm{G}_2^{\mathcal{A},R}(\lambda)$ and $\mathrm{G}_3^{\mathcal{A},R}(\lambda)$ are identical-until-bad. Triggering bad means specifying a non-output wire and its invisible token. As long as bad is not set, the first $\lambda - 1$ bits of each invisible token are uniformly random and independent of whatever $(\mathcal{A}, R)$ receives. Hence

$$\Pr[\mathrm{G}_2^{\mathcal{A},R}(\lambda)] - \Pr[\mathrm{G}_3^{\mathcal{A},R}(\lambda)] \le \Pr[\mathrm{G}_3^{\mathcal{A},R}(\lambda) \text{ sets } \mathsf{bad}] \le \frac{2p(\lambda)}{2^\lambda} \ .$$

In game $\mathrm{G}_4^{\mathcal{A},R}(\lambda)$, the visible tokens at non-input wires and all invisible tokens are unused, and thus the code generating them can be removed. Hence

$$\Pr[\mathrm{G}_3^{\mathcal{A},R}(\lambda)] = \Pr[\mathrm{G}_4^{\mathcal{A},R}(\lambda)] \ .$$

Hence

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\lambda) &= \Pr[\,\mathrm{Reset}_S^P(\lambda) \,|\, c = 1\,] + \Pr[\,\mathrm{Reset}_S^P(\lambda) \,|\, c = 0\,] - 1 \\
&= \Pr[\mathrm{G}_1^{\mathcal{A},R}(\lambda)] - \Pr[\mathrm{G}_4^{\mathcal{A},R}(\lambda)] \\
&= \Pr[\mathrm{G}_2^{\mathcal{A},R}(\lambda)] - \Pr[\mathrm{G}_3^{\mathcal{A},R}(\lambda)] \le \frac{2p(\lambda)}{2^\lambda}
\end{aligned}$$

for every $\lambda \in \mathbb{N}$. ▌

For part (2), let Sim be the simulator constructed in Fig. 25. We'll construct a PT statistically reset-secure source S and a PT distinguisher D such that

$$\mathsf{Adv}^{\mathrm{obv2},\,\Phi_{\mathrm{topo}},\,\mathrm{Sim}}_{\mathsf{GaAO[H]},\mathcal{A}}(\cdot) \le \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot) \ . \tag{22}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\mathrm{srs}}]$. The constructions of S and D are shown below.

$\underline{S^{\textsc{Hash}}(1^\lambda)}$	$\underline{D(1^\lambda, hk, L)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \perp$	$(F, r, X, V) \leftarrow L$; $(V_1, \ldots, V_n) \leftarrow V$
$\mathcal{A}^{\textsc{Garble},\textsc{Input}}(1^\lambda; r)$; Return L	$(X_1, \ldots, X_n) \leftarrow X$
	$b' \leftarrow \mathcal{A}^{\textsc{Garble},\textsc{Input}}(1^\lambda; r)$; Return b'
$\underline{\textsc{Garble}(f)}$	
$(F, (X_1^0, X_1^1, \ldots, X_n^0, X_n^1), d) \leftarrow\!\!\$\ \mathsf{Ga}[\textsc{Hash}].\mathsf{Gb}(1^\lambda, f)$	$\underline{\textsc{Garble}(f)}$
Return F	Return F
$\underline{\textsc{Input}(i, c)}$	$\underline{\textsc{Input}(i, c)}$
$\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i \leftarrow X_i^c$; $V_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$	If $V_i = \perp$ then $V_i \leftarrow 0^{\mathsf{H.kl}(\lambda)}$; $V_i \leftarrow V_1 \oplus \cdots \oplus V_n \oplus hk$
If $\mathsf{cnt} < n$ then return (X_i, V_i)	Return (X_i, V_i)
$V_i \leftarrow \perp$; $X \leftarrow (X_1, \ldots, X_n)$; $V \leftarrow (V_1, \ldots, V_n)$	
$L \leftarrow (F, r, X, V)$; Return $\perp$	

Let a be the challenge bit of game $\mathrm{UCE}^{S,D}_{\mathsf{H}}(\lambda)$, and b be the challenge bit of game $\mathrm{Obv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\mathrm{topo}},\mathrm{Sim}}(\lambda)$.

$\underline{\text{Sim}(1^\lambda, \phi, 0)}$
$(n, m, q, A, B) \leftarrow \phi$
For $g \leftarrow n+1$ to $n+q$, $\mathsf{a} \leftarrow 0$ to 1, $\mathsf{b} \leftarrow 0$ to 1 do
  $T[g, \mathsf{a}, \mathsf{b}] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$, $F \leftarrow (n, m, q, A, B, T)$
Return (F, ε)

$\underline{\text{Sim}(1^\lambda, i, \tau, \mathsf{cnt})}$
$X_i \leftarrow\!\!\$\ \{0,1\}^\lambda$; $K \leftarrow (0^m, hk)$; $V_i \leftarrow 0^{|K|}$
If $\mathsf{cnt} < n$ then $V_i \leftarrow\!\!\$\ \{0,1\}^{|K|}$
Else
  $X \leftarrow (X_1, \ldots, X_n)$
  $Y \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, 1^\lambda)].\mathsf{Ev}(F, X)$
  $(Y_1, \ldots, Y_m) \leftarrow Y$; $y_1 \cdots y_m \leftarrow \tau$
  For $j \leftarrow 1$ to m do $u_j \leftarrow y_j \oplus \text{lsb}(Y_j)$
  $U \leftarrow u_1 \cdots u_m$; $K \leftarrow (U, hk)$
  $V_i \leftarrow V_1 \oplus \cdots \oplus V_n \oplus K$
Return (X_i, V_i)

Figure 26: **Constructed simulator for part (3) of Theorem 5.12.**

Then

$$\Pr[\,\text{Obv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\text{topo}},\text{Sim}}(\cdot) \,|\, b = 1\,] \;=\; \Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a = 1\,], \text{ and}$$
$$\Pr[\,\text{Obv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\text{topo}},\text{Sim}}(\cdot) \,|\, b = 0\,] \;=\; \Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a = 0\,]\ .$$

Summing up yields Equation (22). What's left is to show that S is statistically reset-secure. Indeed, whatever this source leaks is also leaked by the source in part (1). Hence from Lemma B.1, for any adversary R and any polynomial p that bounds the number of R's queries to HASH, it follows that $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\lambda) \le 2p(\lambda)/2^\lambda$ for every $\lambda \in \mathbb{N}$.

For part (3), let Sim be the simulator constructed in Fig. 26. We'll construct a PT statistically reset-secure source S and a PT distinguisher D such that

$$\mathsf{Adv}^{\text{prv2},\,\Phi_{\text{topo}},\,\text{Sim}}_{\mathsf{GaP[H]},\mathcal{A}}(\cdot) \le \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},S,D}(\cdot)\ . \tag{23}$$

The theorem then follows from the assumption that $\mathsf{H} \in \mathsf{UCE}[\mathcal{S}^{\text{srs}}]$. The constructions of S and D are shown below.

$\underline{S^{\text{HASH}}(1^\lambda)}$
$r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $\mathsf{cnt} \leftarrow 0$; $L \leftarrow \perp$; $K \leftarrow (0^m, 0^{\mathsf{H.kl}(\lambda)})$
$\mathcal{A}^{\text{GARBLE},\text{INPUT}}(1^\lambda; r)$; Return L

$\underline{\text{GARBLE}(f)}$
$(F, (X^0_1, X^1_1, \ldots, X^0_n, X^1_n), d) \leftarrow\!\!\$\ \mathsf{Ga}[\text{HASH}].\mathsf{Gb}(1^\lambda, f)$
Return (F, ε)

$\underline{\text{INPUT}(i, c)}$
$\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $X_i = X^c_i$; $V_i \leftarrow\!\!\$\ \{0,1\}^{|K|}$
If $\mathsf{cnt} < n$ then return (X_i, V_i)
$V_i \leftarrow \perp$; $X = (X_1, \ldots, X_n)$; $V = (V_1, \ldots, V_n)$
$L \leftarrow (F, r, X, V)$; Return $\perp$

$\underline{D(1^\lambda, hk, L)}$
$(F, r, X, V) \leftarrow L$; $(V_1, \ldots, V_n) \leftarrow V$
$(X_1, \ldots, X_n) \leftarrow X$
$b' \leftarrow \mathcal{A}^{\text{GARBLE},\text{INPUT}}(1^\lambda; r)$; Return b'

$\underline{\text{GARBLE}(f)}$
Return (F, ε)

$\underline{\text{INPUT}(i, c)}$
$x_i \leftarrow c$
If $V_i = \perp$ then
  $x \leftarrow x_1 \cdots x_n$; $y_1 \cdots y_m \leftarrow y \leftarrow \mathsf{cev}(f, x)$
  $Y \leftarrow \mathsf{Ga}[\mathsf{H.Ev}(1^\lambda, hk, \cdot, \cdot)].\mathsf{Ev}(F, X)$
  $(Y_1, \ldots, Y_m) \leftarrow Y$
  For $j = 1$ to m do $u_j \leftarrow \text{lsb}(Y_j) \oplus y_j$
  $U \leftarrow u_1 \cdots u_m$
  $V_i \leftarrow V_1 \oplus \cdots V_{i-1} \oplus V_i \oplus \cdots \oplus V_n \oplus (U, hk)$
Return (X_i, V_i)

Let a be the challenge bit of game $\text{UCE}^{S,D}_{\mathsf{H}}(\lambda)$, and b be the challenge bit of game $\text{Prv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\text{topo}},\text{Sim}}(\lambda)$. Then

$$\Pr[\,\text{Prv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\text{topo}},\text{Sim}}(\cdot) \,|\, b = 1\,] \;=\; \Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a = 1\,], \text{ and}$$
$$\Pr[\,\text{Prv2}^{\mathcal{A}}_{\mathsf{GaAO[H]},\Phi_{\text{topo}},\text{Sim}}(\cdot) \,|\, b = 0\,] \;=\; \Pr[\,\text{UCE}^{S,D}_{\mathsf{H}}(\cdot) \,|\, a = 0\,]\ .$$

Summing up yields Equation (23). What's left is to show that S is statistically reset-secure. Indeed, whatever this source leaks is also leaked by the source in part (1). Hence from Lemma B.1, for any adversary R and any polynomial p that bounds the number of R's queries to HASH, it follows that $\mathsf{Adv}^{\mathsf{reset}}_{S,R}(\lambda) \leq 2p(\lambda)/2^{\lambda}$ for every $\lambda \in \mathbb{N}$.