Using Indistinguishability Obfuscation via UCEs

Christina Brzuska[1] and Arno Mittelbach[2]

[1] Tel Aviv University, Israel
[2] Darmstadt University of Technology, Germany
brzuska@post.tau.ac.il arno.mittelbach@cased.de

Abstract. We provide the first standard model construction for a powerful class of Universal Computational Extractors (UCEs; Bellare et al. Crypto 2013) based on indistinguishability obfuscation. Our construction suffices to instantiate q-query correlation-secure hash functions and to extract polynomially many hardcore bits from any one-way function. For many cryptographic primitives and in particular for correlation-secure hash functions all known constructions are in the random-oracle model. Indeed, recent negative results by Wichs (ITCS 2013) rule out a large class of techniques to prove the security of correlation-secure hash functions in the standard model. Our construction is based on puncturable PRFs (Sahai und Waters; STOC 2014) and indistinguishability obfuscation. However, our proof also relies on point obfuscation under auxiliary inputs (AIPO). This is crucial in light of Wichs' impossibility result. Namely, Wichs proves that it is often hard to reduce two-stage games (such as UCEs) to a "one-stage assumption" such as DDH. In contrast, AIPOs and their underlying assumptions are inherently two-stage and, thus, allow us to circumvent Wichs' impossibility result.
Our positive result is also noteworthy insofar as Brzuska, Farshim and Mittelbach (Crypto 2014) have shown recently, that iO and some variants of UCEs are mutually exclusive. Our results, hence, validate some of the new UCE notions that emerged as a response to the iO-attack.

Keywords Correlation-secure hash functions, hardcore functions, indistinguishability obfuscation, differing-inputs obfuscation, point-function obfuscation, auxiliary-input obfuscation, universal computational extractors (UCEs)

1 Introduction

For many cryptographic primitives, it is easy to construct a secure scheme in the random oracle model, but it is hard to give a construction in the standard model. For example, correlated-input hash functions (CIH) which were introduced by Goyal, O'Neill, and Rao [31], are easy to construct in the random oracle model, because the random oracle itself is secure under correlated inputs. However, up to now, no standard-model construction is known, and indeed, a recent black-box separation by Wichs [40] explains why it is so hard to construct them. Namely, the security definition of a CIH involves a pair of adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ and is thus

a two-stage game (i.e., the adversary is not a single algorithm but consists of two separate algorithms). The first adversary samples correlated inputs $(x_1, ..., x_t)$. Then a hash key hk is generated and the second adversary with access to hk needs to distinguish between getting a tuple of random strings and getting the tuple $(H(\mathsf{hk}, x_1), ..., H(\mathsf{hk}, x_t))$. Now, Wichs employs a meta reduction to show that it is unlikely to have a black-box reduction $\mathcal{R}$ from CIH to a (one-stage) cryptographic assumption such as the decisional Diffie–Hellman assumption (DDH). Namely, he shows that if such a reduction to DDH exists, then the DDH assumption is wrong. In his proof, he substantially exploits that the CIH game is a two-stage game. For a black-box reduction $\mathcal{R}$ it must hold that if the reduction $\mathcal{R}$ gets access to a pair of oracles $(\mathcal{A}_1, \mathcal{A}_2)$ that break CIH, then $\mathcal{R}^{\mathcal{A}_1,\mathcal{A}_2}$ must also break DDH. Wichs constructs a pair of inefficient adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ which, however, can be efficiently emulated using a stateful simulator Sim. That is, the simulator simulates both adversaries together while sharing state between them. As the reduction cannot distinguish between the two settings $\mathcal{R}^{\mathcal{A}_1,\mathcal{A}_2}$ and $\mathcal{R}^{\mathsf{Sim}}$ this breaks DDH, and hence, if we believe that DDH is a hard problem, then such an $\mathcal{R}$ cannot exist. Note that Wichs' proof is not specific to DDH, but rather applies to any one-stage assumption and presents a substantial barrier to prove security. Moreover, Wichs' impossibility result extends to a range of security notions that are specified by two-stage games.

In this paper, we use cryptographic obfuscation techniques to circumvent Wichs' impossibility result and achieve security notions that are based on two-stage assumptions. Towards this goal, the key idea is to combine point-function obfuscation and indistinguishability obfuscation.

Point and indistinguishability obfuscation. A point function p_x is a function that returns 1 on input x and $\perp$ on all other values. A point function obfuscator under auxiliary input AIPO returns a point function $p \leftarrow_{\$} \mathsf{AIPO}(x)$ that hides the point x even in case the adversary receives some side-channel information z about x. More formally, the security of AIPO is defined as security for all computationally unpredictable distributions $\mathcal{D}$, that is, $\mathcal{D}$ outputs a pair (x, z), where x is a point and z is some leakage that hides x computationally. AIPO is secure, if for all computationally unpredictable $\mathcal{D}$, $(\mathsf{AIPO}(x), z)$ is indistinguishable from $(\mathsf{AIPO}(u), z)$, where $(x, z) \leftarrow \mathcal{D}$ and u is a uniformly random point. Such AIPO schemes have been constructed in [20, 11].

While point function obfuscators are obfuscation schemes for a very specific class of functionalities (namely point functions) Garg et al. [26] have recently revived the study of general obfuscation schemes with their candidate construction of indistinguishability obfuscation. The notion of indistinguishability obfuscation is weaker than VBB-obfuscation—thereby circumventing the impossibility results of Barak et al. [3, 2]—and says intuitively that, for any two circuits that compute the same function, their obfuscations are indistinguishable. The publication of the candidate for indistinguishability obfuscation by Garg et al. inspired simultaneous breakthroughs for hard problems in various sub-areas of cryptography [39, 15, 1, 25, 33, 14, 9, 30] including functional and deniable encryption, two-round secure

multi-party computation, full-domain hash, poly-many hardcore bits from any one-way function, multi-input functional encryption and more.

Correlated-input hash-functions. In this paper, we give the first standard-model construction for q-query CIHs. Our CIH is not only one-way under correlated inputs, but also outputs elements that are indistinguishable from random. We will compare our notion of q-query CIH with other notions of CIHs shortly.

On a high-level, our construction is a de facto instantiation of a random oracle. As the behavior of a PRF is similar to that of a random function, we instantiate the random oracle by securely delegating a PRF, that is, we obfuscate a PRF with a hard coded key. Indeed, our hash-function construction only consists of a (puncturable) PRF that is obfuscated via an indistinguishability obfuscator (iO):

Hash Construction: $\mathsf{iO}(\mathsf{PRF}(k,.))$.

Bellare, Stepanovs, and Tessaro (BST; [9]) already used this natural construction in the direct construction of hardcore functions for injective one-way functions from indistinguishability obfuscation. We will discuss BST and the relation to our our work shortly.

Note that before obfuscating the PRF we need to pad the circuit to a specific length. This is needed when using indistinguishability obfuscation to move from one circuit to another one in the security proof and thus the construction must be padded to the length of the biggest circuit needed within the security proof. Jumping ahead, we note that although our construction and that of BST look identical on the outside the padding is different. For BST, the construction needs to be padded differently depending on the size of the one-way function. In turn, our padding is universal and thus we yield a universal hardcore function that works for any one-way function.

Circumventing Wichs' impossibility result. Although the construction is natural, proving its security is non-trivial, as the security guarantees of iO do not even allow us to show easily that it is hard to extract the PRF key. Towards proving the security of our construction, we build on the puncturable PRF technique by Waters and Sahai [39] and combine it with point function obfuscators secure under auxiliary input (AIPO).

Using AIPOs is crucial to circumvent the impossibility result by Wichs [40], because the security of AIPOs is defined via a two-stage security game. The first AIPO adversary samples a point, and the second adversary tries to break the obfuscation of the point function. In a sense, the impossibility result of Wichs tells us that using a two-stage assumption such as AIPO in the proof is, indeed, necessary. In particular, iO and PRFs are both one-stage assumptions. Note that, as AIPOs are only used in the proof and not in the construction, it might be possible that the same construction can be proven secure without making use of AIPOs possibly through some other two-stage assumption.

Universal hardcore functions for any one-way function. Bellare, Stepanovs, and Tessaro (BST; [9]) recently established that the same construction (with a different

amount of padding) also yields a hardcore function for any injective one-way function, assuming a puncturable PRG and iO.

For general one-way functions, BST gave a second, different construction of a hardcore function and proved it based on so-called differing-inputs obfuscation. Differing-inputs obfuscation is a stronger assumption than iO and has been shown conditionally impossible by Garg et al. [27] assuming special-purpose obfuscators. Therefore, in the current version of their paper, Bellare et al. [9] use a weaker variant of diO that is not affected by the results of Garg et al. [27].

In an updated version of their paper, Garg et al. [28] show that, assuming a special-purpose obfuscator and indistinguishability obfuscation for Turing Machines, there are one-way functions for which the second construction of BST cannot be a secure hardcore function, because their hardcore function has "output-only dependence". This means that hardcore bits $h(x)$ are completely determined by $f(x)$, or in other words, for any inputs x and x' such that $f(x) = f(x')$ it holds that $h(x) = h(x')$. We note that the only candidate for iO for Turing machines is currently based on full diO.

The conditional negative result for output-only dependent hardcore functions does not apply to the construction $\mathsf{iO}(\mathsf{PRF}(k, .))$ which is the construction that we use throughout this paper and which BST—with a different amount of padding—prove to be a hardcore function for injective one-way functions. In turn, assuming AIPO in addition to iO allows us to prove this construction secure for all one-way functions, even those that have many pre-images. Another difference with the BST result is that we yield a universal hardcore function for any one-way function while their padding depends on the one-way function.

Our proof builds on ideas by BST, and we will come back to their result in the context of presenting our proof techniques. We note that for our security proof, we assume AIPO in addition to iO and thereby are able to avoid diO variants altogether. The assumption of point obfuscators is currently incomparable to the assumption of differing-inputs obfuscation as well as to more restricted versions that were used by BST. It is an interesting question to explore the relationship between these assumptions.

Modularizing proofs via UCEs. We could prove the security of our construction directly, but instead, we split our proof into two parts. First, we show that our construction enjoys some useful, abstract properties. Then we use results by Bellare et al. [6] that show that these abstract properties suffice for the application at hand. This way, we provide a means of using iO in a black-box way. Our abstraction is a version of UCE security [6] that we discuss next.

The UCE Framework by Bellare, Hoang, and Keelveedhi (BHK; [6]) introduces assumptions that allow us to instantiate random oracles in a wide range of applications. Loosely speaking, UCEs are PRF-like assumptions that split the distinguisher into two parts: a first adversary S that gets access to a keyed hash function or a random oracle (and which is called the *source*), and a second adversary D that gets the hash key hk (and which is called the *distinguisher*).

The two algorithms together try to guess whether the source was given access to a keyed hash function (under a randomly chosen key) or to a random oracle.

Concretely, the UCE notions are defined via a two-stage UCE game (we depict the communication flow in Figure 1 and the pseudocode in Figure 2). First, the source S is run with oracle access to HASH (which either implements a random oracle or the hash function with a randomly chosen key hk) to output some leakage L. Subsequently, distinguisher D is run on the leakage L and hash key hk but without access to oracle HASH. Distinguisher D outputs a single bit b indicating whether oracle HASH implements a random oracle or hash function H with key hk.

Without any restrictions, (S, D) can easily win the UCE game. For example, say, source S makes a random query x to receive $y \leftarrow \text{HASH}(x)$ and outputs (x, y) as leakage. As distinguisher D knows the hash key hk as well as the leakage (x, y), it can recompute the hash value and check whether $y = \mathsf{H}(\mathsf{hk}, x)$.

BHK present several possible restrictions on the source which give rise to various UCE notions. It turns out to be particularly useful to restrict sources to be computationally unpredictable, that is, the leakage created by the source S—when interacting with a random oracle—should not reveal (computationally) any of the source's queries to HASH. This notion is denoted by UCE$[\mathcal{S}^{\text{cup}}]$, where $\mathcal{S}^{\text{cup}}$ denotes the class of computationally unpredictable sources [7]. BHK show that UCE$[\mathcal{S}^{\text{cup}}]$-secure hash functions can safely replace a random oracle in a large number of interesting applications such as hardcore functions or deterministic public-key encryption [6]. In a recent work Brzuska, Farshim and Mittelbach (BFM; [17]) show that UCE security with respect to computational unpredictability cannot be achieved in the standard model assuming indistinguishability obfuscation exists. Several refinements have been proposed since, including a statistical notion of unpredictability denoted by $\mathcal{S}^{\text{sup}}$ as well as source classes containing sources that are structurally required to produce output in a special way as well as sources which are restricted to only a fixed number of queries [7, 17, 36].

Our notion of UCE security strengthens the notion of unpredictability to what we call strong unpredictability and we denote the corresponding class of sources by $\mathcal{S}^{\text{s-cup}}$ for the computational variant and by $\mathcal{S}^{\text{s-sup}}$ for its statistical version. Namely, we demand that the leakage be computationally/statistically unpredictable even if the predictor additionally gets the answers to the queries that the source received from the oracle. We give the pseudo-code for strong unpredictability in Figure 3.

It turns out that UCEs for strongly computationally unpredictable sources that can only make a single query (denoted by UCE$[\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{\text{1-query}}]$) already imply hardcore functions for any one-way function. Furthermore, UCEs for strongly statistically unpredictable sources that can only make q queries (denoted UCE$[\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}]$) imply q-query correlation-secure hash functions. We note that strongly unpredictable sources can be regarded as a generalization of so-called split sources [7] which were introduced by BHK after the BFM impossibility results. We will discuss the exact relationship later.

So far UCEs have only been constructed in idealized models. BHK showed that a random oracle is UCE-secure in the strongest proposed settings and conjectured that HMAC is UCE-secure if the underlying compression function is modeled as an ideal function. This conjecture has recently been confirmed by Mittelbach [37] who shows that HMAC and various Merkle-Damgård variants are UCE-secure in the ideal compression function model. We note that so far, no standard model instantiation of any (non-trivial) UCE variant has been proposed and, hence, we present the first standard model construction of UCEs.[1]

Techniques. Our construction is based on indistinguishability obfuscation and similar to many other recent constructions from iO [39, 9, 33, 14] our construction also makes use of puncturable PRFs [39] which admit the generation of keys that allow to evaluate the PRF on all points except for points in a small target set (often containing just a single point). Our security reduction, however, differs from existing techniques. That is, we make use of point function obfuscations which allows us to hide the punctured points within our constructed circuits. Hiding the punctured points was also the key problem in the earlier discussed work by Bellare, Stepanovs and Tessaro [9] who construct hardcore-functions for one-way functions. They solve the problem elegantly by using the one-way function from the security game to blind the punctured point by embedding the image under the one-way function. However, when testing whether a given point is equivalent to the punctured point this test is ambiguous which is why they need to assume differing-inputs obfuscators for one-way functions that map more than polynomially many points to the same image value. This is where point function obfuscation comes into the picture which allows us to bypass any assumptions related to differing-input obfuscation variants. Yet, of course, point obfuscators are as far as is currently known an assumption incomparable to differing-inputs obfuscation.

Point obfuscation and iO. In a recent and independent work, Hofheinz uses point obfuscation in a similar way to construct fully secure constrained pseudorandom functions [32] in the random oracle model. A constrained PRF is a generalized form of a puncturable PRF which allows for the generation of keys that enable the holder to evaluate the PRF on a set of points but not on all points. In contrast to previous constructions [13, 16, 34] Hofheinz uses point obfuscation and an extension he introduces as *extensible testers* in conjunction with indistinguishability obfuscation to hide which points a given key allows to honestly evaluate. This allows him to achieve full security without relying on complexity leveraging which was used in previous constructions entailing a superpolynomial loss of security in the adaptive setting. We note that unlike this work Hofheinz relies on the simpler assumption of plain point obfuscation (that is, obfuscation without auxiliary inputs) and he shows how to build extensible testers based on the DDH-based point obfuscator by Canetti [20].

[1] The UCE Framework is very flexible and it is, for example, possible to define a UCE restriction that corresponds to PRF security.

Brzuska and Mittelbach study the connection between point obfuscation with multi-bit output secure in the presence of auxiliary inputs and indistinguishability obfuscation [18]. They show that indistinguishability obfuscation and a strong form of multi-bit point obfuscation are mutually exclusive. Their results do not carry over to the setting of statistically hard-to-invert auxiliary information (which we rely on for CIHs) and it is not clear if their results can be extended to cover plain AIPO, that is point functions with single-bit outputs.

Our results. We next discuss the specific UCE assumptions that our construction will meet and the relation to the specific point obfuscation schemes used. In Section 3 we will show that our construction is UCE[$\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{\text{1-query}}$]-secure assuming iO, puncturable PRFs and the existence of AIPO. That is, we consider UCE-secure for computationally strongly unpredictable sources that make a single query. In Section 3.3, we prove that our construction is also UCE[$\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}$]-secure, that is, secure against statistically unpredictable sources that make at most q queries.

As explained, we base the security of our construction on the existence of a different (incomparable) notion of point obfuscation. We consider a notion of AIPO which only needs to be secure against statistically unpredictable distributions but, in turn, we require it to be q-composable [21, 10]. Intuitively, q-composability says that an obfuscation remains secure even if an adversary sees q many (possibly related) obfuscations. The reason that we need q-composable AIPO is that now, the source is a allowed to make q queries and hence, we need to hide q points in the proof. q-composable AIPO implies multi-bit point function obfuscation [21] and thus does not exist, if iO exists [18].

However, as we here only consider sources in $\mathcal{S}^{\text{s-sup}}$, that is, sources which are only statistically strongly unpredictable, it suffices that our AIPO-notion is secure against statistically unpredictable samplers which weakens the notion of AIPO. Matsuda and Hanoka [35] have recently shown that q-composable AIPO secure against statistically unpredictable samplers is implied by composable VGB-AI point obfuscators, a notion that Bitansky and Canetti constructed under the q-Strong Vector Decision Diffie Hellman assumption [10]. Note that, for the proof to work, we need to let the circuit size of our construction grow, artificially, with the number of queries q. Towards this goal, we use some padding that does not have any functionality.

In summary we get the following results:

1. Our construction is UCE[$\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{\text{1-query}}$]-secure assuming indistinguishability obfuscation for all circuits in $\mathcal{P}$/poly and AIPO secure with respect to *computationally* hard-to-invert auxiliary information exist.
2. Our construction is UCE[$\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}$]-secure assuming indistinguishability obfuscation for all circuits in $\mathcal{P}$/poly and q-composable AIPO with respect to *statistically* hard-to-invert auxiliary information exist.

On the feasibility of our AIPO *assumptions.* Standard AIPO secure against computationally unpredictable samplers has been constructed by Canetti in [20] under

(non-standard) variants of the DDH assumption and by Bitansky and Paneth in [11] under (non-standard) assumptions on pseudorandom permutations. We discuss the constructions and the underlying assumptions in the full version of this work [19]. One might hope that AIPO is naturally composable. However, Canetti et al. show that this is generally not the case [21, 10]. On the other hand, Bitansky and Canetti [10] show that under the *t-Strong Vector Decision Diffie Hellman assumption* the original point obfuscation scheme of Canetti [20] composes in the so-called virtual grey-box (VGB) setting. The VGB setting was introduced by Bitansky and Canetti [10] and is a relaxation of the strongest obfuscation setting the virtual black-box (VBB) setting [3, 2]. Similarly to VBB obfuscation, VGB obfuscation is in general not achievable, yet for the class of point functions it seems in reach [10]. The VGB setting is particularly interesting because "plain" VGB and VGB with auxiliary information are equivalent [10]. This result stands in contrast to the VBB setting where allowing auxiliary information results in a stronger notion. Furthermore, we currently have no candidate constructions for composable point obfuscation schemes in this stronger setting. We note that for our purpose composable obfuscation in the VGB setting is sufficient for our purpose as Matsuda and Hanaoka [35] show that this setting already implies q-composable AIPO with respect to statistically unpredictable samplers which form the basis for our q-query correlation-secure hash functions.

In a very recent work Brzuska and Mittelbach (BM) investigate the connection between indistinguishability obfuscation and multi-bit output point obfuscation secure in the presence of auxiliary input (MB-AIPO) [18]. A multi-bit point function $p_{x,m}$ is zero everywhere except on x where it outputs m. BM show that various strong notions of MB-AIPO and indistinguishability obfuscation are mutually exclusive. However, their results do not seem to carry over to plain AIPO, that is to AIPO for plain point functions as needed in our constructions. We refer to [18] for a discussion on MB-AIPO and discuss the implications of an extension of the results of BM to plain AIPO shortly when talking about the feasibility of our UCE notions.

On the feasibility of our UCE notions. In a recent work, Brzuska, Farshim, and Mittelbach (BFM; [17]) show that, assuming indistinguishability obfuscation exists, no standard model hash construction can be UCE-secure with respect to computationally unpredictable sources. Our construction achieves a weaker yet related notion of security, namely UCE-security with respect to strongly computationally unpredictable sources which raises the question whether the BFM result can be extended to this setting.

The BFM result crucially hinges on the possibility of extending the output-length of the studied hash construction such that it is significantly larger than the key size. For example, this can be achieved by using multiple queries to the hash construction or via extending the output size by applying a pseudo-random generator [17, 8]. Both approaches fail with our construction: the size of our hash key grows with the number of allowed queries and since we consider strong unpredictability it seems implausible to prove the construction $\mathsf{PRG}(\mathsf{H}(\cdot,\cdot))$-secure under the assumption that H is UCE-secure with respect to strongly

computationally unpredictable sources. Thus, we think that extending the BFM attack is implausible. Furthermore, if it can be extended this would immediately imply that indistinguishability obfuscation implies the non-existence of AIPO, which would be a surprising result. We discuss the BFM result in greater detail in the full version [19] and note that, even if an extension of the BFM result were to break AIPOs with computational unpredictability, then the second construction would not be affected, as it only considers AIPOs secure with respect to statistically hard-to-invert auxiliary information.

Notions of Correlation-Secure Hash-Functions. We now compare our notion of q-query CIHs to different notions of correlated-input security. Note that q-query CIH means that the size of the hash-key can depend on the number of inputs q. However, and that is a crucial difference to previous works, each input value is hashed using the same hash-key. In turn, Freeman et al. [23] as well as Rosen and Segev [38] use a fresh hash-key for every input. Notably, the correlation-secure functions that they construct also have a trapdoor. Note that the correlated-input variant[2] of the IND security game for deterministic public-key encryption [5, 4, 12] and the CIH game are almost identical if it is required that the CIH has a trapdoor. We can then view the computation of the CIH as an encryption operation and the CIH game becomes a slightly stronger version of the IND security game (that is, a real-or-random rather than a left-or-right game). Hence, a CIH function which has a trapdoor is also a deterministic public-key encryption scheme.

As in the schemes of [23, 38] a new key needs to be generated for every new message, the constructions are not a deterministic public-key encryption scheme. In turn, if our q-query CIH were a trapdoor function, then by definition, it would also be a q-query deterministic public-key encryption scheme. Unfortunately, our construction of a q-query CIH does not come with a trapdoor, and we do not know whether this is possible.

Another related notion of CIH are *statistically* secure q-query CIHs. Here, as for our notion of q-query CIH, the key size may grow with the number of queries and one uses the same hash key for each query. In contrast to our security notion one here requires that the output is statistically close to random given the hash key. As we are concerned with statistical security, this notion is only achievable for distributions that come with a notable amount of entropy, that is, the q pre-images need to have entropy that is at least q times the output length. In turn, for the notion of entropy that we consider, the entropy of the pre-images does not need to grow with q and can also be less than the length of the output.

Hence, this notion of statistically secure CIH only applies to a substantially smaller class of distributions. In turn, while our construction relies on the strong assumption of indistinguishability obfuscation, statistically secure CIH can be achieved without any assumptions. That is, if the pre-images carry enough (true) entropy, then one can extract q uniformly random image values by using a q-wise independent hash-functions [24].

[2] Here, we refer to the variant where each message needs to have high entropy on its own, but might have low entropy conditioned on the other messages.

Finally, Goyal, O'Neill, and Rao [31] construct CIHs that are secure under polynomially related inputs and introduce a hierarchy of CIH notions: *One-wayness* under correlated inputs, *unpredictability* under correlated inputs and *pseudorandomness* under correlated inputs. These notions describe a hierarchy of security notions when we consider CIHs with superlogarithmic output length. We note that we achieve the strongest of these notions, namely *pseudorandomness* under correlated inputs.

Full version. Due to space restrictions, this version should be regarded as an extended abstract as we defer many details and all proofs to the full version [19]. In the remainder of this extended abstract we present our main results and give some intuition for the underlying proofs.

2 Preliminaries

2.1 Obfuscation

Indistinguishability obfuscation. While the strongest obfuscation notion, that is, virtual black-box obfuscation provably does not exist in general for all circuits [3], weaker notions such as *indistinguishability obfuscation* may well exist. VBB requires the existence of a simulator. On the other hand, an indistinguishability obfuscation (iO) scheme only ensures that the obfuscations of any two functionally equivalent circuits are computationally indistinguishable. Indistinguishability obfuscation was originally proposed by Barak et al. [3] as a potential weakening of virtual-black-box obfuscation. We recall the definition from [26].

Definition 1. *A* PPT *algorithm* iO *is called an* indistinguishability obfuscator *for a circuit ensemble* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *if the following conditions are satisfied:*

- **Correctness.** *For all security parameters* $\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, and for all inputs* x *we have that* $\Pr\left[C'(x) = C(x) : C' \leftarrow_{\$} \mathsf{iO}(1^\lambda, C)\right] = 1$.
- **Security.** *For any* PPT *distinguisher* $\mathcal{D}$*, for all pairs of circuits* $C_0, C_1 \in \mathcal{C}_\lambda$ *such that* $C_0(x) = C_1(x)$ *on all inputs* x *the following distinguishing advantage is negligible:*

$$\left|\Pr\left[\mathcal{D}(1^\lambda, \mathsf{iO}(1^\lambda, C_1)) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, \mathsf{iO}(1^\lambda, C_0)) = 1\right]\right| \leq \mathsf{negl}(\lambda)\,.$$

Closely related to indistinguishability obfuscation is the notion of *differing-inputs obfuscation* (diO) which also goes back to the seminal paper of Barak et al. [3]. Building on a theorem by Boyle, Chung and Pass [15], we are able to avoid diO as an assumption and only use it as an intermediary concept in our proof. We defer the details to the full version [19].

Point obfuscation. While indistinguishability, as well as differing-inputs, obfuscation are obfuscation schemes for general circuits one can also study obfuscation schemes for particular function classes such as point functions. A point function p_x for some value $x \in \{0,1\}^*$ maps every input to $\perp$ except for x which is mapped

to 1. We consider a variant of point function obfuscators under auxiliary input which was first formalized by Canetti [20], although in a slightly different context. We here give the definition from [11] presented in a game based formulation. The first definition formalizes unpredictable distributions which are in turn used to define obfuscators for point functions.

Definition 2 (Unpredictable distribution). *A distribution ensemble* $\mathcal{D} = \{D_\lambda = (Z_\lambda, X_\lambda)\}_{\lambda\in\mathbb{N}}$*, on pairs of strings is unpredictable if no poly-size (non-uniform) circuit can predict* X_λ *from* Z_λ*. That is, for every poly-size circuit sequence* $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ *and for all large enough* λ*:*

$$\Pr_{(z,x)\leftarrow\$\ D_n}[C_\lambda(z) = x] \leq \mathsf{negl}(\lambda)$$

Definition 3 (Auxiliary input point obfuscation for unpredictable distributions (AIPO)). *A* PPT *algorithm* AIPO *is a* point obfuscator for unpredictable distributions *if it satisfies the functionality and polynomial slowdown requirements as VBB obfuscation [3, 2], and the following secrecy property: for any (efficiently sampleable) unpredictable distribution* $\mathcal{B}_1$ *over* $\{0,1\}^{\mathsf{poly}(\lambda)} \times \{0,1\}^\lambda$ *it holds for any* PPT *algorithm* $\mathcal{B}_2$ *that the probability that the following experiment outputs true for* $(\mathcal{B}_1, \mathcal{B}_2)$ *is negligibly close to* $\frac{1}{2}$*:*

$$\begin{aligned}
&b \leftarrow\$ \{0,1\}\\
&(z, x_0) \leftarrow\$ \mathcal{B}_1(1^\lambda)\\
&x_1 \leftarrow\$ \{0,1\}^\lambda\\
&p \leftarrow\$ \mathsf{AIPO}(x_b)\\
&b' \leftarrow\$ \mathcal{B}_2(1^\lambda, p, z)\\
&\textbf{\textit{return}}\ b = b'
\end{aligned}$$

The probability is over the coins of adversary $(\mathcal{B}_1, \mathcal{B}_2)$*, the coins of* AIPO *and the choices of* x_1 *and* b*.*

2.2 Universal Computational Extractors (UCE)

The UCE Framework by Bellare, Hoang, and Keelveedhi (BHK; [6]) introduces assumptions that allow us to instantiate random oracles in a wide range of applications and which are not succeptible to the impossibility result by Canetti, Goldreich and Halevi [22]. Loosely speaking, UCEs are PRF-like assumptions that split the distinguisher into two parts: a first adversary S that gets access to a keyed hash function or a random oracle (and which is called the *source*), and a second adversary D that gets the hash key hk (and which is called the *distinguisher*). The two algorithms together try to guess whether the source was given access to a keyed hash function or to a random oracle.

Concretely, the UCE notions are defined via a two-stage UCE game (we depict the communication flow in Figure 1 and the pseudocode in Figure 2). First, the source S is run with oracle access to HASH to output some leakage L. Subsequently, distinguisher D is run on the leakage L and hash key hk but without

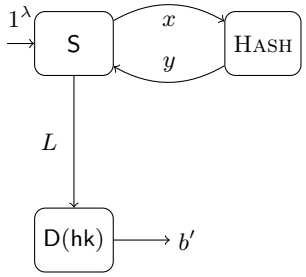

Fig. 1: Schematic of the UCE game.

MAIN $\mathrm{UCE}^{\mathsf{S},\mathsf{D}}_{\mathsf{H}}(\lambda)$

$b \leftarrow\$ \{0,1\}$
$\mathsf{hk} \leftarrow\$ \mathsf{H.KGen}(1^\lambda)$
$L \leftarrow\$ \mathsf{S}^{\text{HASH}}(1^\lambda)$
$b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
return $(b = b')$

HASH(x)

if $T[x] = \bot$ **then**
  if $b = 1$ **then**
    $T[x] \leftarrow \mathsf{H.Eval}(\mathsf{hk}, x)$
  else $T[x] \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
return $T[x]$

MAIN $\mathsf{Pred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)$

$\mathsf{done} \leftarrow \bot;\ Q \leftarrow \emptyset$
$L \leftarrow\$ \mathsf{S}^{\text{HASH}}(1^\lambda)$
$\mathsf{done} \leftarrow \top$
$Q' \leftarrow\$ \mathsf{P}^{\text{HASH}}(1^\lambda, L)$
return $(Q \cap Q' \neq \emptyset)$

HASH(x)

if $\mathsf{done} = \bot$ **then**
  $Q \leftarrow Q \cup \{x\}$
if $T[x] = \bot$ **then**
  $T[x] \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
return $T[x]$

Fig. 2: The UCE game together with the unpredictability game. In the UCE game source S has access to HASH, which returns real or ideal hash values, and leaks L to distinguisher D. The latter additionally gets the hash key and outputs a bit b'. On the right we give the unpredictability game.

access to oracle HASH. Distinguisher D outputs a single bit b indicating whether oracle HASH implements a random oracle or hash function H with key hk.

Without any restrictions, (S, D) can easily win the UCE game. For example, say, source S makes a random query x to receive $y \leftarrow \text{HASH}(x)$ and outputs (x, y) as leakage. As distinguisher D knows the hash key hk as well as the leakage (x, y), it can recompute the hash value and check whether $y = \mathsf{H}(\mathsf{hk}, x)$. BHK present several possible restrictions on the source which give rise to various UCE notions.

Formal UCE Definition. In line with [9] we consider families of functions F consisting of algorithms $F.\mathsf{KGen}$, $F.\mathsf{kl}$, $F.\mathsf{Eval}$, $F.\mathsf{il}$ and $F.\mathsf{ol}$. Algorithm $F.\mathsf{KGen}$ is a PPT algorithm taking the security parameter 1^λ and outputting a key $k \in \{0,1\}^{F.\mathsf{kl}(\lambda)}$ where $F.\mathsf{kl} : \mathbb{N} \to \mathbb{N}$ denotes the key length. Functions $F.\mathsf{il} : \mathbb{N} \to \mathbb{N}$ and $F.\mathsf{ol} : \mathbb{N} \to \mathbb{N}$ denote the input and output length functions associated to F and for any $x \in \{0,1\}^{F.\mathsf{il}(\lambda)}$ and $k \leftarrow\$ F.\mathsf{KGen}(1^\lambda)$ we have that $F.\mathsf{Eval}(k, x) \in \{0,1\}^{F.\mathsf{ol}(\lambda)}$, where the PPT algorithm $F.\mathsf{Eval}$ denotes the "evaluation" function associated to F.

We denote hash functions by H. Let $\mathsf{H} = (\mathsf{H.KGen}, \mathsf{H.Eval}, \mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol})$ be a hash-function family and let (S, D) be a pair of PPT algorithms. We define the UCE advantage of a pair (S, D) against H through

$$\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathsf{S},\mathsf{D}}(\lambda) := 2 \cdot \Pr\left[\mathrm{UCE}^{\mathsf{S},\mathsf{D}}_{\mathsf{H}}(\lambda)\right] - 1\ ,$$

where game $\mathrm{UCE}^{\mathsf{S},\mathsf{D}}_{\mathsf{H}}(\lambda)$ is shown in Figure 2 on the left (in Figure 1 we give a schematic overview of the communication within the game).

Unpredictability. Without any further restrictions there are PPT pairs (S, D) that achieve an advantage in the $\mathrm{UCE}^{\mathsf{S},\mathsf{D}}_{\mathsf{H}}(\lambda)$ game close to 1. BHK define several

possible restrictions for sources yielding various flavors of UCE assumptions [6]. Here, we are interested in a strengthened version of the original *computational* unpredictability [6] restriction. A source S is called *computationally unpredictable* if the advantage of any PPT predictor P, defined by

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

is negligible, where game $\mathsf{Pred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)$ is shown in Figure 2 on the right. In line with [7], we call the class of all computationally unpredictable sources $\mathcal{S}^{\mathrm{cup}}$, where $\mathcal{S}^{\mathrm{cup}}$ denotes the class (set) of all computationally unpredictable sources. Similarly, we define the class of statistically unpredictable sources where the predictor in game $\mathsf{Pred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)$ can run in unbounded time but is still restricted to only polynomially many oracle queries. The class of statistically unpredictable sources is denoted by $\mathcal{S}^{\mathrm{sup}}$.

UCE *Security.* We say a hash function H is UCE secure for sources $\mathsf{S} \in \mathcal{S}$ denoted by UCE[$\mathcal{S}$], if for all PPT sources $\mathsf{S} \in \mathcal{S}$ and all PPT distinguishers D the advantage $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathsf{S},\mathsf{D}}(\lambda)$ is negligible. In that way we get the UCE assumptions UCE[$\mathcal{S}^{\mathrm{cup}}$] and UCE[$\mathcal{S}^{\mathrm{sup}}$], that is, UCE with respect to computationally (resp. statistically) unpredictable sources.[3]

2.3 Puncturable PRFs

Besides point function obfuscation schemes, our main ingredient in the upcoming proofs are so-called puncturable pseudorandom functions (PRF) [39]. A family of puncturable PRFs $G :=(G.\mathsf{KGen}, G.\mathsf{Puncture}, G.\mathsf{kl}, G.\mathsf{Eval}, G.\mathsf{il}, G.\mathsf{ol})$ consists of functions that specify input length, output length and key length as well as a key generation algorithm $k \leftarrow G.\mathsf{KGen}$, a deterministic evaluation algorithm $G.\mathsf{Eval}(k, x)$ that takes a key k, an input x of length $G.\mathsf{il}(1^\lambda)$ and outputs a value y of length $G.\mathsf{ol}(1^\lambda)$. Additionally, there is a PPT puncturing algorithm $G.\mathsf{Puncture}$ which on input a polynomial-size set $S \subseteq \{0,1\}^{G.\mathsf{il}(\lambda)}$, outputs a special key k_S. A family of functions is called puncturable PRF if the following two properties are observed

- **Functionality preserved under puncturing.** For every PPT adversary $\mathcal{A}$ such that $\mathcal{A}(1^\lambda)$ outputs a polynomial-size set $S \subseteq \{0,1\}^{G.\mathsf{il}(\lambda)}$, it holds for all $x \in \{0,1\}^{G.\mathsf{il}(\lambda)}$ where $x \notin S$ that:

  $$\Pr\left[G.\mathsf{Eval}(k,x) = G.\mathsf{Eval}(k_S,x) : k \leftarrow\!\!\$\ G.\mathsf{KGen}(1^\lambda), k_S \leftarrow\!\!\$\ G.\mathsf{Puncture}(k,S)\right] = 1$$

- **Pseudorandom at punctured points.** For every PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathcal{A}_1(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{G.\mathsf{il}(\lambda)}$ and state st, consider an experiment where $k \leftarrow G.\mathsf{KGen}(1^\lambda)$ and $k_S = G.\mathsf{Puncture}(k, S)$. Then we have

  $$\left|\Pr\left[\mathcal{A}_2(\mathsf{st}, k_S, S, G.\mathsf{Eval}(k,S)) = 1\right] - \Pr\left[\mathcal{A}_2(\mathsf{st}, k_S, S, U_{G.\mathsf{ol}(\lambda)\cdot|S|}) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

[3] The notion UCE[$\mathcal{S}^{\mathrm{cup}}$] was originally named UCE1 and later changed to UCE[$\mathcal{S}^{\mathrm{cup}}$] [6, 7]. The notion of statistical unpredictability was introduced in [17, 7].

where $\mathsf{Eval}(k, S)$ denotes the concatenation of $\mathsf{Eval}(k, x_1), \ldots, \mathsf{Eval}(k, x_k)$ where $S = \{x_1, \ldots, x_k\}$ is the enumeration of the elements of S in lexicographic order, negl is a negligible function, and U_ℓ denotes the uniform distribution over $\{0,1\}^\ell$.

As observed by [13, 16, 34] puncturable PRFs can, for example, be constructed from pseudorandom generators via the GGM tree-based construction [29]. As AIPO implies one-way functions [19] AIPO also implies puncturable PRFs.

3 UCEs from iO and Point Obfuscation

In this section we present our constructions of UCEs from iO and AIPO. We first define the precise UCE notions that our constructions achieve and introduce the UCE restriction of *strong unpredictability*. We will then in Section 3.2 present a construction of a UCE-secure function with respect to sources which are strongly computationally-unpredictable and which make exactly one oracle query. In Section 3.3 we will show how to extend the construction to allow for an a-priory fixed number of queries by switching to a statistical version of strong unpredictability.

Interestingly, our two constructions are basically the same modulo circuit padding. That is, our constructions depend on an obfuscation of a circuit, which in both cases is the same but padded to a different length. A larger but functionally equivalent circuit seems to be necessary to allow for multiple source queries.

We discuss applications of our constructions in the full version of this work [19]. Due to space limitations we also defer to the full version [19] for a discussion on why our construction does not (seem to) fall pray to the BFM attacks on computationally unpredictable sources [17].

3.1 Strongly Unpredictable and q-Query Sources

We now introduce the precise source restrictions for our upcoming UCE constructions. We define a new restriction that we call *strong unpredictability* and which can be seen as either a stronger form of unpredictability or a relaxed version of split sources. Secondly, we consider sources that make only a bounded number of oracle queries.

Strong unpredictability. We consider sources which are strongly unpredictable both in the computational and in the statistical sense. We denote by $\mathcal{S}^{\text{s-cup}}$ the class of sources which are strongly, computationally unpredictable and by $\mathcal{S}^{\text{s-sup}}$ the class of strongly, statistically unpredictable sources. Strong unpredictability is a stronger requirement than unpredictability and we require that the leakage hides queries to HASH even if the predictor is given the query results. We say that a source S is called *strongly computationally unpredictable* if the advantage of any PPT predictor P, defined by

$$\mathsf{Adv}^{\mathsf{stpred}}_{\mathsf{S},\mathsf{P}}(\lambda) := \Pr\left[\mathsf{stPred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)\right],$$

MAIN $\mathsf{stPred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)$

$X^*, Y^* \leftarrow \emptyset$
$b \leftarrow\$ \{0,1\}$
$L \leftarrow\$ \mathsf{S}^{\text{HASH}}(1^\lambda)$
$x' \leftarrow\$ \mathsf{P}^{\text{HASH}}(1^\lambda, L, Y^*)$
return $(x' \in X^*)$

HASH(x)

$X^* \leftarrow X^* \cup \{x\}$
$y \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$Y^* \leftarrow Y^* \cup \{y\}$
return y

Splt SOURCE $\mathsf{S}^{\text{HASH}}(1^\lambda)$

$(L_0, \mathbf{x}) \leftarrow\$ \mathsf{S}_0(1^\lambda)$
for $i = 1, \ldots, |\mathbf{x}|$ **do** $\mathbf{y}[i] \leftarrow\$$ HASH$(\mathbf{x}[i])$
$L_1 \leftarrow\$ \mathsf{S}_1(1^\lambda, \mathbf{y}); L \leftarrow (L_0, L_1)$
return L

Fig. 3: On the left: the strong unpredictability game where the predictor, in addition to the leakage is also given the result of the HASH queries. On the right: the definition of split sources [7]. A split source $\mathsf{S} = \mathsf{Splt}[S_0, S_1]$ consists of two parts S_0 and S_1 that jointly generate leakage L and neither part gets direct oracle access to HASH.

is negligible, where game $\mathsf{stPred}^{\mathsf{P}}_{\mathsf{S}}(\lambda)$ is shown in Figure 3 on the left. For the case of strongly statistically unpredictable sources ($\mathcal{S}^{\text{s-sup}}$) we allow the predictor to be unbounded in its running time, but restrict the number of oracle queries to be bounded polynomially.

In order to circumvent the BFM attacks on computationally unpredictable sources BHK introduce the notion of split sources [7, 17]. A source S is called split source, denoted by $\mathsf{S} \in \mathcal{S}^{\text{splt}}$ if it can be decomposed into two algorithms S_0 and S_1 such that neither part gets direct access to oracle HASH. We give the pseudocode of split sources in Figure 3 on the right. In a first step algorithm S_0 outputs a leakage string L_0 together with a vector $\mathbf{x}$. Then, each of the entries in $\mathbf{x}$ is queried to HASH and the results stored in vector $\mathbf{y}$. Finally, the second algorithm S_1 is run on vector $\mathbf{y}$ to produce the second part of the leakage L_1.

One can prove that split sources are a (strict) subclass of strongly unpredictable sources, that is, $\mathcal{S}^{\text{splt}} \cap \mathcal{S}^{\text{cup}} \subsetneq \mathcal{S}^{\text{s-cup}}$ (and similarly in the statistical case $\mathcal{S}^{\text{splt}} \cap \mathcal{S}^{\text{sup}} \subsetneq \mathcal{S}^{\text{s-sup}}$). For further information on the implications see the full version of this work [19].

q-query UCE. Our first construction only admits sources which make exactly one query. We call such sources single-query sources and denote the corresponding source class by $\mathcal{S}^{\text{1-query}}$. We also consider a relaxed notion to allow for polynomially bounded number of queries for some polynomial $q := q(\lambda)$. We call the corresponding sources q-query sources and denote their class by $\mathcal{S}^{q\text{-query}}$. We note that sources restricted to a constant number of queries are discussed in [7].

3.2 A UCE Construction Secure Against Sources in $\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{\text{1-query}}$

We will now present our construction which depending on different assumptions on the existence of point obfuscators will achieve UCE$[\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{\text{1-query}}]$-security or

UCE$[\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}]$-security. Note that depending on the number of supported queries the construction needs to pad the circuit before obfuscating it.

Construction 1. *Let $s : \mathbb{N} \to \mathbb{N}$, let G be a puncturable PRF and let iO be an indistinguishability obfuscator for all circuits in $\mathcal{P}/\text{poly}$. We define our hash function family H as*

$\underline{\mathsf{H}.\mathit{KGen}(1^\lambda)}$

$k \leftarrow\!\!\$\ G.\mathit{KGen}(1^\lambda)$
$\mathsf{hk} \leftarrow\!\!\$\ \mathsf{iO}(\mathsf{PAD}(s(\lambda), G.\mathsf{Eval}(k, \cdot)))$
return hk

$\underline{\mathsf{H}.\mathsf{Eval}(\mathsf{hk}, x)}$

$\overline{C} \leftarrow \mathsf{hk}$
return $\overline{C}(x)$

where $\mathsf{PAD} : \mathbb{N} \times \{0,1\}^* \longrightarrow \{0,1\}^*$ denotes a deterministic padding algorithm that takes as input an integer and a circuit and outputs a functionally equivalent circuit padded to length $s(\lambda)$.[4]

In other words, the key generation algorithm $\mathsf{H}.\mathsf{KGen}(1^\lambda)$ runs $k \leftarrow G.\mathsf{KGen}(1^\lambda)$ and returns $\mathsf{iO}(G.\mathsf{Eval}(k, \cdot))$, i.e., an obfuscation of the evaluation circuit of PRF G with key k hardwired into it. Function $\mathsf{H}.\mathsf{Eval}$ is basically a universal Turing machine which runs input x on the obfuscated circuit hk.

Theorem 2. *If G is a secure puncturable PRF, if iO is a secure indistinguishability obfuscator and if AIPO exists, then the hash function family H defined in Construction 1 is* UCE$[\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{1\text{-query}}]$*-secure.*

We prove the theorem via a sequence of 5 games (depicted in Figure 4 on page 20) where game Game_1 denotes the original UCE$[\mathcal{S}^{\text{s-cup}} \cap \mathcal{S}^{1\text{-query}}]$ game with hidden bit b fixed to 1. We present the proof in the full version of this work [19].

3.3 A UCE Construction Secure Against Sources in $\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}$

In this section we show that our construction is also UCE-secure with respect to sources which are strongly unpredictable in a statistical sense and which allow the source to make q-many queries for any polynomial $q := q(\lambda)$. That is, we consider sources in class $\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}$.

In case we allow the source to make q many queries, the first observation is that we need to choose the size of our obfuscated circuit such we can puncture at q many points. For each point, we will encode a random string into the circuit and thus, the circuit size grows with the number of points we need to puncture out. Besides this, the construction is identical to the one before with the exception that we need a different (incomparable) security property of our point function obfuscation scheme. That is, we require the point obfuscator to be a q-composable VGB obfuscator secure in the presence of statistically unpredictable auxiliary information which implies an AIPO obfuscator with statistically unpredictable auxiliary information. We refer to the full version for further details [19].

[4] Function s needs to be chosen in accordance with the puncturable PRF to allow for the required number of puncturings.

Theorem 3. *Let q be a polynomial. If G is a secure puncturable PRF, if* iO *is a secure indistinguishability obfuscator and if there exist a q-composable VGB point obfuscator for statistically unpredicatable auxiliary input, then the hash function family* H *defined in Construction 1 is* $\mathrm{UCE}[\mathcal{S}^{\text{s-sup}} \cap \mathcal{S}^{q\text{-query}}]$*-secure.*

The proof follows analogously to the proof of Theorem 2, except for puncturing at several points instead of a single point and therefore, we reduce to q-composable VGB point obfuscation. We defer the proof to the full version [19].

Acknowledgments

We thank the Asiacrypt 2014 reviewers for the many constructive comments. We especially thank Paul Baecher, Mihir Bellare, Pooya Farshim, Victoria Fehr, Giorgia Azzurra Marson, Adam O'Neill and Daniel Wichs for many helpful comments and discussions throughout the various stages of this work. Christina Brzuska was supported by the Israel Science Foundation (grant 1076/11 and 1155/11), the Israel Ministry of Science and Technology grant 3-9094), and the German-Israeli Foundation for Scientific Research and Development (grant 1152/2011). Arno Mittelbach was supported by CASED (www.cased.de) and the German Research Foundation (DFG) SPP 1736.

References

1. Ananth, P., Boneh, D., Garg, S., Sahai, A., Zhandry, M.: Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689 (2013), http://eprint.iacr.org/2013/689
2. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im)possibility of obfuscating programs. J. ACM 59(2), 6:1–6:48 (May 2012), http://doi.acm.org/10.1145/2160158.2160159
3. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer (Aug 2001)
4. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer (Aug 2007)
5. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer (Aug 2008)
6. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer (Aug 2013)
7. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424 (2013), http://eprint.iacr.org/2013/424
8. Bellare, M., Hoang, V.T., Keelveedhi, S.: Personal communication (Sep, 2013)
9. Bellare, M., Stepanovs, I., Tessaro, S.: Poly-many hardcore bits for any one-way function. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. pp. ??–?? LNCS, Springer, Berlin, Germany, Kaohsiung, Taiwan (Dec 7–11, 2014)

10. Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer (Aug 2010)
11. Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 190–208. Springer (Mar 2012)
12. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer (Aug 2008)
13. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 280–300. Springer (Dec 2013)
14. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 480–499. Springer (Aug 2014)
15. Boyle, E., Chung, K.M., Pass, R.: On extractability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 52–73. Springer (Feb 2014)
16. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 501–519. Springer (Mar 2014)
17. Brzuska, C., Farshim, P., Mittelbach, A.: Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 188–205. Springer (Aug 2014)
18. Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. pp. ??–?? LNCS, Springer, Berlin, Germany, Kaohsiung, Taiwan (Dec 7–11, 2014)
19. Brzuska, C., Mittelbach, A.: Using indistinguishability obfuscation via uces. Cryptology ePrint Archive, Report 2014/381 (2014), http://eprint.iacr.org/
20. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 455–469. Springer (Aug 1997)
21. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer (Apr 2008)
22. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC. pp. 209–218. ACM Press (May 1998)
23. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. Journal of Cryptology 26(1), 39–74 (Jan 2013)
24. Fuller, B., O'Neill, A., Reyzin, L.: A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 582–599. Springer (Mar 2012)
25. Garg, S., Gentry, C., Halevi, S., Raykova, M.: Two-round secure MPC from indistinguishability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 74–94. Springer (Feb 2014)
26. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS. pp. 40–49. IEEE Computer Society Press (Oct 2013)
27. Garg, S., Gentry, C., Halevi, S., Wichs, D.: On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 518–535. Springer (Aug 2014)

28. Garg, S., Gentry, C., Halevi, S., Wichs, D.: On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. Cryptology ePrint Archive, Report 2013/860 version from 13 Jun 2014 (Jun 2014), http://eprint.iacr.org/
29. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th FOCS. pp. 464–479. IEEE Computer Society Press (Oct 1984)
30. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 578–602. Springer (May 2014)
31. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer (Mar 2011)
32. Hofheinz, D.: Fully secure constrained pseudorandom functions using random oracles. Cryptology ePrint Archive, Report 2014/372 (2014), http://eprint.iacr.org/
33. Hohenberger, S., Sahai, A., Waters, B.: Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 201–220. Springer (May 2014)
34. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 13. pp. 669–684. ACM Press (Nov 2013)
35. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via point obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 95–120. Springer (Feb 2014)
36. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via UCE. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 56–76. Springer (Mar 2014)
37. Mittelbach, A.: Salvaging indifferentiability in a multi-stage setting. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 603–621. Springer (May 2014)
38. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. SIAM Journal on Computing 39(7), 3058–3088 (2010)
39. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC. pp. 475–484. ACM Press (May / Jun 2014)
40. Wichs, D.: Barriers in cryptography with weak, correlated and leaky sources. In: Kleinberg, R.D. (ed.) ITCS 2013. pp. 111–126. ACM (Jan 2013)

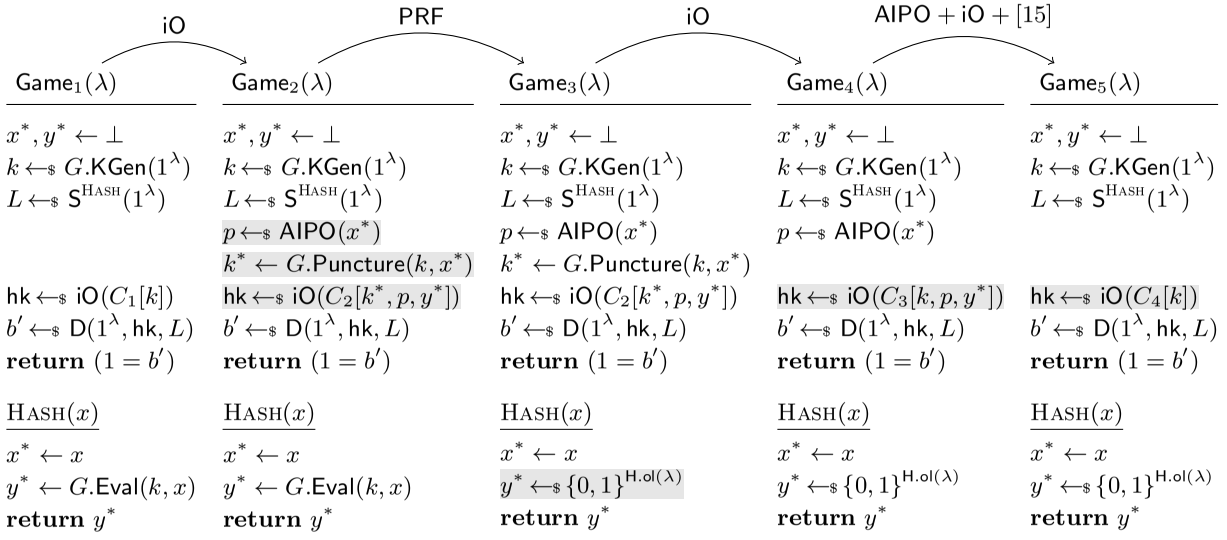

Fig. 4: The games used in the proof of Theorem 2 on the top and the used circuits on the bottom. To highlight the changes from game to game we have marked the changed lines with a light gray background color. By $C[k](x)$ we denote that circuit C depends on k (during construction time) and takes x as input. The arrows above the games indicate the security reduction to get from Game_i to Game_{i+1}.