Leakage Resilient One-Way Functions: The Auxiliary-Input Setting

Ilan Komargodski(✉)

Weizmann Institute of Science, Rehovot, Israel
ilan.komargodski@weizmann.ac.il

Abstract. Most cryptographic schemes are designed in a model where perfect secrecy of the secret key is assumed. In most physical implementations, however, some form of information leakage is inherent and unavoidable. To deal with this, a flurry of works showed how to construct basic cryptographic primitives that are resilient to various forms of leakage.

Dodis et al. (FOCS '10) formalized and constructed leakage resilient one-way functions. These are one-way functions f such that given a random image $f(x)$ and leakage $g(x)$ it is still hard to invert $f(x)$. Based on any one-way function, Dodis et al. constructed such a one-way function that is leakage resilient assuming that an attacker can leak any lossy function g of the input.

In this work we consider the problem of constructing leakage resilient one-way functions that are secure with respect to *arbitrary computationally hiding* leakage (a.k.a auxiliary-input). We consider both types of leakage — selective and adaptive — and prove various possibility and impossibility results.

On the negative side, we show that if the leakage is an adaptively-chosen arbitrary one-way function, then it is *impossible* to construct leakage resilient one-way functions. The latter is proved both in the random oracle model (without any further assumptions) and in the standard model based on a strong vector-variant of DDH. On the positive side, we observe that when the leakage is chosen ahead of time, there are leakage resilient one-way functions based on a variety of assumption.

1 Introduction

The holy grail of cryptography is designing systems that remain secure in the presence of adversarial behavior. For this, one has to specify (1) a cryptographic primitive of interest (e.g. an encryption scheme or a signature scheme), and (2) a model that captures the power of a potential adversary and what it means for it to break the system.

I. Komargodski—Supported in part by a Levzion fellowship, by grants from the Israel Science Foundation grant no. 1255/12, BSF and from the I-CORE Program of the Planning and Budgeting Committee and the Israel Science Foundation (grant no. 4/11).

M. Hirt and A. Smith (Eds.): TCC 2016-B, Part I, LNCS 9985, pp. 139–158, 2016.
DOI: 10.1007/978-3-662-53641-4_6

One of the most common assumptions is that secret keys are perfectly secret and are completely unknown to an adversary. However, in many physical implementations some information does leak due to various side-channel attacks, reuse of randomness, and more.

This deficiency raised the necessity to build a theory of security against classes of side-channel attacks. Starting with the works of [14, 27, 30], a flurry of works in which different classes of side channel attacks have been defined and different cryptographic primitives have been designed to provably withstand these attacks (see, for example, [1, 2, 6–9, 14–16, 18, 20–24, 27, 30, 31, 34]).

We consider the problem of constructing the most basic cryptographic primitive, a one-way function, in a setting where an adversary obtains side-channel information (this notion was first formalized by [3, 17]). A one-way function f is an efficiently computable function such that given $f(x)$ for a random input x, any efficient adversary cannot find an x' such that $f(x) = f(x')$. A leakage resilient one-way function f is a one-way function such that given $f(x)$ as above and $g(x)$, where g is adversarially chosen, it is still hard to invert f and recover such an x'.

To obtain some sort of security, one clearly has to restrict the adversary to choose g from some collection of functions that do not trivially reveal x by themselves. Indeed, if g is the identity function, no leakage resilient function f exists. Thus, several assumptions on the power of the adversary have been considered. Already in the work of Canetti et al. [14], the authors showed how to obtain a leakage-resilient one-way function assuming that the attacker can leak an arbitrary but sufficiently small subset of the bits of the input. However, this may be overly restrictive as it provides no guarantees if the attacker can learn the XOR of all the input bits. This issue was addressed in several works (see, for example, [3, 15, 17]) showing that there exists a leakage-resilient one-way function assuming that the attacker can leak any lossy function of the input, namely, any function whose image size is significantly smaller than the domain size. The leakage-resilience in both settings is proven based on the existence of any one-way function which is the weakest assumption possible. For completeness, we provide a proof of the following theorem in Appendix A.

Theorem 1 ([3, 17], Informal). *Assuming that one-way functions exist, there exists a one-way function f, such that for any adversarially-chosen lossy function g, given $f(x)$ and $g(x)$ for a random x, it is computationally hard to invert f.*

Motivated by the positive results for a wide class of leakage functions, we study the question of designing leakage-resilient one-way functions that are secure with respect to *arbitrary computationally hiding* leakage function. We model this by allowing the leakage to be an arbitrary one-way function, even such that fully determine the input.[1] We consider both an *adaptive* notion of security in which the leakage function is adversarially chosen (from a restricted

[1] This setting is sometimes referred to as the *auxiliary-input* setting (see, for example, [16, 18, 26]).

pre-defined collection) after f is fixed, and a selective notion in which the leakage is chosen ahead of time, before f is.

1.1 Our Contributions

Adaptively-chosen leakage. We show that if the leakage can be an arbitrary one-way function, then there cannot be a leakage resilient one-way function f. More precisely, we show that for every one-way function f, there exists a one-way function g (that depends on f) such that when one gets both $f(x)$ and $g(x)$, it is easy to invert f.

We prove this result in two ways: in the random oracle model and in the standard model based on a strong vector-variant of DDH. Specifically, we first show that if the leakage function has access to a random oracle O, then we can construct an oracle-aided function g^{O} which is one-way and $g^{\mathsf{O}}(x)$ together with $f(x)$ allow to recover x. For the result in the standard model, we rely on multi-bit point obfuscators that exist based on a strong vector-variant of the DDH assumption [5, 13]; see Sect. 2.3 and Theorem 8.

Theorem 2 (Informal). *Let O be a random oracle. For every one-way function f, there is a one-way function g^{O} such that for every x given $f(x)$ and $g(x)$ it is easy to recover x.*

Theorem 3 (Informal). *Assuming multi-bit point obfuscators, for every one-way function f, there is a one-way function g such that for every x given $f(x)$ and $g(x)$ it is easy to recover x.*

Moreover, such multi-bit point obfuscators can be constructed from a strong vector-variant of the DDH assumption.

Selectively-chosen leakage. We show that if the leakage function g is fixed ahead of time, then there exists a leakage resilient one-way function f for g from various assumptions. To this end, we observe that one-wayness with respect to selectively-chosen leakage is tightly related to extracting polynomially-many hard-core bits.

Theorem 4 (Informal). *For every leakage one-way function g, a hardcore function for g that outputs polynomially-many hard-core bits is a leakage-resilient one-way function for g.*

If g is a sub-exponentially hard one-way function, then extracting polynomially-many hard-core bits is possible due to Goldreich and Levin [25] (and any pseudorandom generator). Bellare, Stepanovs, and Tessaro [4] (see also the follow-up work of Brzuska and Mittelbach [11]) were the first to show how to extract *any* polynomial number of hard-core bits from *any* one-way function. Their construction is based on obfuscation. More recently, Zhandry [36] obtained the same result based on exponentially-hard DDH.

Thus, instantiating Theorem 4 with the variety of known methods for extracting polynomially-many hard-core bits from g, we obtain a leakage-resilient one-way function for g, whose security is based either on one-way functions, on obfuscation, on exponential hardness of DDH, and more.

1.2 Overview of Our Techniques

In Theorem 2 the underlying idea is very simple. We assume a random oracle O and assume that there exists a leakage resilient one-way function f, where the leakage is any one-way function. We define a leakage function $g(x) = \mathsf{O}(f(x)) \oplus x$. Recovering x given $f(x)$ and $g(x)$ is easy by first applying O to $f(x)$ and then XORing the result with $g(x)$. The non-trivial part is showing that this function g is also one-way.

Roughly speaking, our analysis uses the fact that any adversary trying to invert $g(x)$ will have to query the oracle at the point $f(x)$. Otherwise, all it sees are uniform strings from which it cannot infer anything about a possible pre-image. It is left to show that $f(x)$ is sufficiently random so that it cannot be guessed by any polynomial-time adversary with non-negligible probability. Indeed, since f by itself is a one-way function, its image distribution has super-logarithmic min-entropy which satisfied our requirement.

For Theorem 3, our construction is based on multi-bit point obfuscators MBPO and can be seen as an instantiation of the above idea in the standard model. The leakage function, on input x, will output a multi-bit point obfuscation of the multi-bit point function that maps $f(x)$ to x, denoted by $g(x) = \mathsf{MBPO}(I_{f(x)\to x})$. One obstacle is that an obfuscator is a *probabilistic* procedure, and thus cannot be used directly in our setting. Hence, we use *public-coin* multi-bit point obfuscators, which are obfuscators that output their internal random coins. This allows us to define a leakage function which has hard-wired random coins for the use of the point obfuscator. Specifically, we hardwire into g random coins r and define $g_r(x) = \mathsf{MBPO}(I_{f(x)\to x}; r)$. We show that g_r, with very high probability, is a one-way function using the security of the obfuscator.[2]

We observe that such a multi-bit point obfuscator exists based on the strong vector-variant of DDH of Bitansky and Canetti [5] given in Sect. 2.3.[3]

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For a randomized function f and an input $x \in \mathcal{X}$, we denote by $y \leftarrow f(x)$ the process of sampling a value y from the distribution $f(x)$. A function $\mathsf{neg} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(\lambda) < \lambda^{-c}$ for all $\lambda > N_c$. For two strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ we denote by $x||y$ the string concatenation of x and y.

[2] Theorem 2 can also be proved by first showing how to use a random oracle to construct a multi-bit point obfuscator. We thank a reviewer for pointing this out.

[3] We emphasize we do not require security with respect to auxiliary-input, which was shown to be a problematic assumption [10].

Two sequences of random variables $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *computationally indistinguishable* if for any probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that $|\mathbf{Pr}[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \mathbf{Pr}[\mathcal{A}(1^\lambda, Y_\lambda) = 1]| \leq \mathsf{neg}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

2.1 Min-Entropy

The $\mathsf{min\text{-}entropy}$ of a distribution X over $\{0,1\}^n$ is defined by

$$\mathsf{H}_\infty(X) = - \min_{x \in \{0,1\}^n} \log_2 \mathbf{Pr}[X = x].$$

2.2 One-Way Functions

Definition 1 (One-way functions). *A function $f \colon \{0,1\}^* \to \{0,1\}^*$ is said to be* $\mathsf{one\text{-}way}$ *if the following two conditions hold:*

1. *There exists a polynomial-time algorithm A such that $A(x) = f(x)$ for every $x \in \{0,1\}^*$.*
2. *For every probabilistic polynomial-time algorithm B there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\mathsf{ADV}^{\mathsf{OWF}}_{f,B} = \mathbf{Pr}[B(1^n, f(x)) \in f^{-1}(f(x))] \leq \mathsf{neg}(n),$$

where the probability is taken uniformly over all possible $x \in \{0,1\}^n$ and the internal randomness of B.

The following claim will be useful.

Claim 5. *Let $f \colon \{0,1\}^n \to \{0,1\}^m$ be a one-way function where $m = m(n)$ is a polynomial. It holds that $\mathsf{H}_\infty(f(X)) \geq \omega(\log n)$.*

Proof. Since f is a one-way function, it must be that on a random $x \in \{0,1\}^n$, it is hard to a preimage for $f(x)$. Assume, towards contradiction, that $\mathsf{H}_\infty(f(X)) = O(\log n)$. That is,

$$\min_{y \in \{0,1\}^m} \log_2 \frac{1}{\mathbf{Pr}_{x \in \{0,1\}^n}[f(x) = y]} = O(\log n).$$

Thus, there exists a $y^* \in \{0,1\}^m$ for which $\mathbf{Pr}_{x \in \{0,1\}^n}[f(x) = y^*] \geq 1/p(n)$ for some polynomial $p(\cdot)$.

Define an adversary $\mathcal{A}$ that given a random image $y = f(x)$ outputs a uniformly random x'. This adversary wins if both $f(x) = y^*$ and $f(x') = y^*$. Since x and x' are chosen independently and uniformly at random, we have that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{x' \in \{0,1\}^n}[\mathcal{A}(f(x')) \in f^{-1}(y)] &\geq \mathop{\mathbf{Pr}}_{x,x' \in \{0,1\}^n}[f(x) = y^* \text{ and } f(x') = y^*] \\ &= (\mathop{\mathbf{Pr}}_{x \in \{0,1\}^n}[f(x) = y^*])^2 \geq 1/(p(n))^2.\end{aligned}$$

That is, $\mathcal{A}$ will successfully invert y with non-negligible probability, contradiction the one-wayness of f.

We extend the definition of a one-way function to *oracle-aided* one-way functions. Roughly speaking, an oracle-aided function f^O is an oracle-aided one-way function if there is an oracle-aided efficient algorithm that computes f^O on every point, and given an image of f^O on a random preimage, any efficient algorithm (that has oracle access to O) cannot find the preimage.

Definition 2 (Oracle aided one-way function). *Let O be an oracle. A function f^O that has oracle access to O is said to be oracle aided one-way if the following two conditions hold:*

1. *There exists an oracle-aided polynomial-time algorithm A^O such that $A^O(x) = f^O(x)$ for every $x \in \{0,1\}^*$.*
2. *For every oracle-aided probabilistic polynomial-time algorithm B^O and $n \in \mathbb{N}$,*

$$\mathsf{ADV}^{\mathsf{OOWF}}_{f,B} = \mathbf{Pr}[B^O(1^n, f^O(x)) \in (f^O)^{-1}(f^O(x))] < \mathsf{neg}(n),$$

where the probability is taken uniformly over all possible $x \in \{0,1\}^n$ and the internal randomness of B.

2.3 Point Obfuscations

A point function $I_x \colon \{0,1\}^n \to \{0,1\}$ returns 1 on input $x \in \{0,1\}^n$ and 0 on all other inputs. A point obfuscator is an obfuscator that gets a point function I_x as input (in some canonical form in which x is explicit) and outputs a circuit with the same functionality but where x is computationally hidden.

Definition 3 (Point obfuscator). *A point obfuscator $\mathsf{PO}(\cdot)$ is a probabilistic polynomial-time algorithm that gets as input a point function I_x, where $x \in \{0,1\}^n$, and outputs a circuit C such that*

1. *For all x, the circuit $C \leftarrow \mathsf{PO}(I_x)$ is functionally equivalent to I_x.*
2. *For any probabilistic polynomial-time algorithm $\mathcal{A}$, there is an probabilistic polynomial-time simulator S and a negligible function $\mathsf{neg}(\cdot)$, such that for all $x \in \{0,1\}^n$ and $n \in \mathbb{N}$,*

$$\mathsf{ADV}^{\mathsf{PO}}_{\mathcal{A},\mathcal{D}} = |\Pr_{\mathcal{A},\mathsf{PO}}[\mathcal{A}(\mathsf{PO}(I_x)) = 1] - \Pr_{S}[S^{I_x}(1^n)) = 1]| \le \mathsf{neg}(n).$$

Moreover, a point obfuscator is called public coin if it publishes all internal coin tosses as part of its output.

In [12], Canetti provided a construction that satisfies Definition 3 assuming a strong variant of the DDH assumption. The construction of Canetti is given next.

Construction 6 ([12]'s point obfuscator). *Let $\mathcal{G} = \{\mathbb{G}_n\}_{n\in\mathbb{N}}$ be a group ensemble with uniform and efficient representation and operations, where each $\mathbb{G}_n$ is a group of prime order $p_n \in (2^{n-1}, 2^n)$. The* public coin *point obfuscator PO for points in the domain $\mathbb{Z}_{p_n}$ is defined as follows: $\mathsf{PO}(I_x)$ samples a random generator $r \leftarrow \mathbb{G}_n^*$ of $\mathbb{G}_n$ and outputs r, r^x. Evaluation of the obfuscation at point z is done by checking whether $r^x = r^z$.*

A multi-bit point function $I_{x \to y} \colon \{0,1\}^n \to \{0,1\}^m$ is a function that returns $y \in \{0,1\}^m$ on input $x \in \{0,1\}^n$ and $\perp$ on all other inputs. A multi-bit point function obfuscator, given a multi-bit point function in some canonical form in which x and y are explicit, outputs a circuit with the same functionality but where x and y are computationally hidden.

Definition 4 (Multi-bit point obfuscator). *A multi-bit point obfuscator* MBPO *is a probabilistic polynomial-time algorithm that gets as input a multi-bit point function $I_{x \to y}$, where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, and outputs a circuit C such that*

1. *For all $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, the circuit $C \leftarrow \mathsf{MBPO}(I_{x \to y})$ is functionally equivalent to the function $I_{x \to y}$.*
2. *For any probabilistic polynomial-time algorithm $\mathcal{A}$, there is a probabilistic polynomial-time simulator S and a negligible function $\mathsf{neg}(\cdot)$,*[4] *such that for all $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $y \in \{0,1\}^m$ and*

$$\begin{aligned}\mathsf{ADV}^{\mathsf{MBPO}}_{\mathcal{A},\mathcal{D}} &= |\Pr_{\mathcal{A},\mathsf{MBPO}}[\mathcal{A}(\mathsf{MBPO}(I_{x \to y})) = 1] - \Pr_{S}[S^{I_{x \to y}}(1^{n+m}) = 1]| \\ &\leq \mathsf{neg}(n).\end{aligned}$$

Moreover, a multi-bit point obfuscator is called public coin *if it publishes all internal coin tosses as part of its output.*

One way to obtain a multi-bit point obfuscator was suggested by Canetti and Dakdouk [13]. Specifically, they showed that a *composable* point obfuscator gives rise to a multi-bit point obfuscator.

Definition 5 (Composable point obfuscator). *A point obfuscator $\mathsf{PO}(\cdot)$ is said to be t-*composable *if for any probabilistic polynomial-time algorithm $\mathcal{A}$, there is a probabilistic polynomial-time simulator S and a negligible function $\mathsf{neg}(\cdot)$ such that for any $x_1, \ldots, x_t$ it holds that*

$$\begin{aligned}\mathsf{ADV}^{t\text{-}\mathsf{PO}}_{\mathcal{A},\mathcal{D}} &= |\Pr_{\mathcal{A},\mathsf{PO}}[\mathcal{A}(\mathsf{PO}(I_{x_1}), \ldots, \mathsf{PO}(I_{x_t})) = 1] - \Pr_{S}[S^{I_{x_1}, \ldots, I_{x_t}}(1^{t \cdot n})) = 1]| \\ &\leq \mathsf{neg}(n).\end{aligned}$$

Canetti and Dakdouk [13] showed how to use an m-composable point function obfuscator PO to obtain a multi-bit point function that supports outputs (i.e. y values) of length m. Specifically, they suggested the following construction.

[4] We note that for our application of the multi-bit point obfuscator, it is enough to consider the seemingly relaxed notion of virtual grey-box (VGB) multi-bit point obfuscators, where the simulator has a polynomial bound on the number of queries to its oracle, but is otherwise unlimited. We use the stronger definition which is implied by the weaker one [5, Proposition 7.3].

Construction 7 ([13]'s multi-bit point obfuscator). *Let* PO *be a point obfuscator for the domain* $\{0,1\}^n$. *Given a point* $x \in \{0,1\}^n$ *and value* $y = y_1 \dots y_m \in \{0,1\}^m$, *sample* $s \leftarrow \{0,1\}^n$ *uniformly at random and let*

$$a_i = \begin{cases} x & \text{if } i = 0 \text{ or } y_i = 1, \\ s & \text{otherwise.} \end{cases}$$

Now, the obfuscation of $I_{x,y}$ *is*

$$\mathsf{MBPO}(I_{x \to y}) = \mathsf{PO}(I_{a_0}), \dots, \mathsf{PO}(I_{a_m}), \tag{1}$$

and in order to evaluate $\mathsf{MBPO}(I_{x \to y})$ *on input* z *one first checks if* $z = a_0 = x$ *(by evaluating the first obfuscated circuit). If not (namely,* $z \neq a_0$*), then it outputs* $\perp$. *Otherwise (namely, if* $z = a_0$*), it evaluated all other point obfuscations to find all coordinates in which* $z = a_i = x$ *and outputs* $y_1 \dots y_m$, *where* $y_i = 1$ *if* $a_1 = z = x$ *(and* 0 *otherwise). Notice that if* PO *is public coin then so is* MBPO.

Bitansky and Canetti [5] showed that under the $(m+1)$-strong vector DDH assumption (defined next), the point obfuscator of Canetti from Theorem 6 is $(m+1)$-composable and thus can be used to get a multi-bit point function. We further observe that since Canetti's point obfuscator is public coin (see Theorem 6), it follows that Canetti and Dakdouk's multi-bit point obfuscator is public coin. We begin with the assumption and then state the theorem.

Definition 6 (Well spread distribution). *A distribution* $\mathcal{X}_n$ *over* $\{0,1\}^n$ *is* well-spread *if it is efficiently and uniformly samplable, and it has super-logarithmic min-entropy. Namely,* $\mathsf{H}_\infty(\mathcal{X}_n) \geq \omega(\log n)$.

Let $m = m(n)$ *be a polynomial. An ensemble of distributions* $\mathcal{X}_n^{(1)}, \dots, \mathcal{X}_n^{(m)}$ *(each over* $\{0,1\}^n$*) is* coordinate-wise well-spread *if for each* $i \in [m]$, $\mathcal{X}_n^{(i)}$ *is well-spread.*

Assumption 8 (m-strong vector DDH [5]). *Let* $m = \mathsf{poly}(n)$. *There exists a group ensemble* $\mathcal{G} = \{\mathbb{G}_n\}_{n \in \mathbb{N}}$, , *where each* $\mathbb{G}_n$ *is a group of prime order* p_n *with uniform and efficient representation and operations, such that for any coordinate-wise well-spread distribution ensemble* $\mathcal{X} = \{\mathcal{X}_n = (\mathcal{X}_n^{(1)}, \dots, \mathcal{X}_n^{(m)})\}_{n \in \mathbb{N}}$ *over vectors in* $\mathbb{Z}_{p_n}^m$ *the following two ensembles are computationally indistinguishable:*[5]

$$((g_1, g_1^{a_1}), \dots, (g_m, g_m^{a_m})), \text{ where } g_1, \dots, g_m \leftarrow \mathbb{G}_n^* \text{ and } (a_1, \dots, a_m) \leftarrow \mathcal{X}_n$$

and

$$((g_1, g_1^{a_1}), \dots, (g_m, g_m^{a_m})), \text{ where } g_1, \dots, g_m \leftarrow \mathbb{G}_n^* \text{ and } (a_1, \dots, a_m) \leftarrow \mathbb{Z}_{p_n}^m.$$

[5] There is a variant for this definition which bears more similarities to DDH, generalizes the assumption of Canetti [12], and it is equivalent to the definition we presented as long as $m \geq 2$. See [5] for more information.

Now we are ready to state the resulting theorem of [5] from Theorem 7 with the underlying Theorem 8.[6]

Theorem 9. *Assume the $(m + 1)$-strong vector DDH assumption. Then, the construction from Eq. 1 is a* public coin *multi-bit point obfuscator for multi-bit point functions that output m bits.*

3 Definition of Leakage Resilient One-Way Functions

Here we define leakage resilient one-way functions. Intuitively, a one-way function f is leakage resilient for leakage function g if given $f(x)$ and $g(x)$ it is hard to recover an x' such that $f(x') = f(x)$, where x is chosen uniformly at random. Our actual definition is a relaxation and a generalization of the above informal description: (1) we allow f to be sampled from a collection of functions, and (2) we let g come from an a-priori fixed collection of leakage functions.

More precisely, a leakage resilient one-way function collection $\mathcal{F} = \{f\colon \{0,1\}^n \to \{0,1\}^*\}$ is defined with respect to a collection of leakage functions $\mathcal{L} = \{g\colon \{0,1\}^n \to \{0,1\}^*\}$. $\mathcal{F}$ is said to be leakage resilient one-way if given $f \leftarrow \mathcal{F}$ it is hard to invert $f(x)$ on a random image even given f and $g(x)$ for any adaptively chosen $g \in \mathcal{L}$ (namely, the choice of g can depend on f).

Definition 7 (Leakage resilient one-way function). *Let $\mathcal{F} = \{f\colon \{0,1\}^n \to \{0,1\}^*\}$ be a collection of functions associated with an efficient probabilistic sampler $\mathsf{Gen}_{\mathcal{F}}(1^n)$ that outputs a function $f \in \mathcal{F}$ together with an efficient (deterministic) algorithm for evaluating f.*

The function collection $\mathcal{F}$ is a leakage resilient one-way function collection *for a collection of functions $\mathcal{L} = \{g\colon \{0,1\}^n \to \{0,1\}^*\}$ if for every probabilistic polynomial-time algorithms $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, there exists a negligible function $\mathsf{neg}(\cdot)$ such that for every $n \in \mathbb{N}$ it holds that*

$$\mathsf{ADV}^{\mathsf{lrOWF}}_{\mathcal{A},\mathcal{F},\mathcal{L}} = \mathbf{Pr}[\mathsf{EXP}_{\mathcal{A},\mathcal{F},\mathcal{L}}(n) = 1] \leq \mathsf{neg}(n),$$

where the random variable $\mathsf{EXP}_{\mathcal{A},\mathcal{F},\mathcal{L}}(n)$ is defined via the following experiment:

1. *$f \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^n)$.*
2. *$(g, \mathsf{state}) \leftarrow \mathcal{A}_0(1^n, f)$, where $g \in \mathcal{L}$.*
3. *$x^* \leftarrow \{0,1\}^n$ (chosen uniformly at random and independently of f and g).*
4. *$x \leftarrow \mathcal{A}_1(f, f(x^*), g(x^*), \mathsf{state})$.*
5. *If $f(x) = f(x^*)$, then output 1, and otherwise output 0.*

[6] It may seem odd that Definitions 3 and 4 are stated in a "worst-case" language, while Theorem 8 is stated in an "average-case" language. However, notice that the former are definitions that are given in a simulation-based language while the latter is an indistinguishability-based one. It is known that for (multi-bit) point functions all of these variants are equivalent (see [5, Theorem 5.1 and Proposition 7.3] for a proof).

If $\mathcal{L}$ consists of one fixed leakage function g,[7] *then we say that f is a* selective *leakage resilient one-way function for $\mathcal{L}$. Otherwise, it is called an* adaptive *leakage resilient one-way function.*

One vs. a collection of leakage resilient functions. One may also be interested in a single one-way function $f\colon \{0,1\}^n \to \{0,1\}^*$ which is leakage resilient. In this case, Item 1 in the definition of the experiment $\mathsf{EXP}_{\mathcal{A},\mathcal{F},\mathcal{L}}(n)$ can be ignored. We chose to present and work with a definition which allows f to be chosen from a family as it is more general and since some of our results actually require having f be chosen from a collection.

Adaptive vs. selective security. Our definition captures both adaptive and selective (i.e. non-adaptive) choice of the leakage. Indeed, if the collection $\mathcal{L}$ consists of a single function g, then we can choose the leakage resilient collection $\mathcal{F}$ *knowing* the leakage g ahead of time (we think of this as the *selective* setting). On the other hand, if the collection $\mathcal{L}$ contains more functions, we view the security requirement as an adaptive one, since one has to design the collection $\mathcal{F}$ without knowing in advance which $g \in \mathcal{L}$ will be chosen by an adversary. To exemplify an extreme case of the last point, consider the case in which $\mathcal{L}$ is the set of *all* one-way functions. Then, when designing $\mathcal{F}$, one has very little information about the leakage.

What kind of leakage makes sense? It does not make sense to allow $g \in \mathcal{L}$ to output x, as in this case there is no leakage resilient one-way function family $\mathcal{L}$. This means that every $g \in \mathcal{L}$ has to introduce some hardness for inverting x from $g(x)$ (when x is a uniform input). (This is a standard and necessary assumption.) There are several interesting settings for the leakage collection $\mathcal{L}$, for example:

1. All one-way functions.
2. All sub-exponentially hard one-way functions.
3. All functions whose image size is significantly smaller than the domain size.
4. An arbitrary single one-way function.

The notion in Item 3 was studied earlier (see, for example, [3,17] and implicitly in [2,29]) and was proven to be achievable from any one-way function. For completeness we present the construction and proof in Appendix A. In the main body, we study all other notions.

4 Impossibility of Adaptive Leakage Resilient One-Way Functions

In this section we prove our negative results. We show that without non-trivial limitation on the leakage collection $\mathcal{L}$, there cannot be a leakage resilient one-way functions. Specifically, we show that if the leakage collection $\mathcal{L}$ consists of

[7] Recall that f and g receive the same input so defining g to be some sort of a universal circuit and thereby obtaining a huge family of functions is useless.

all one-way functions, there cannot be a leakage resilient one-way function for $\mathcal{L}$. In particular, the leakage can be chosen after the leakage resilient function is chosen and depend on it.

In Sect. 4.1 we prove this in the random oracle model, where functions have access to a random oracle (and without any further cryptographic assumptions). In Sect. 4.2 we provide a construction in the standard model whose security relies on any public-coin multi-bit point obfuscator.

4.1 Impossibility in the Random Oracle Model

The following theorem shows that there cannot be a leakage resilient one-way function family $\mathcal{F}$ if the leakage function can depend on the function f chosen from $\mathcal{F}$ and if it has oracle access to a random oracle.

Theorem 10. *Let* $\mathsf{O}\colon \{0,1\}^* \to \{0,1\}^n$ *be a random oracle. Let* $\mathcal{L}^{\mathsf{O}} = \{g\colon \{0,1\}^n \to \{0,1\}^*\}$ *be the collection of all oracle-aided one-way functions. There is no leakage-resilient one-way function family* $\mathcal{F} = \{f\colon \{0,1\}^n \to \{0,1\}^*\}$ *for the collection* $\mathcal{L}^{\mathsf{O}}$*.*

Proof. Assume towards contradiction that such a function $f\colon \{0,1\}^n \to \{0,1\}^*$ exists, where $f \in \mathcal{F}$. We shall define an oracle-aided one-way function $g \in \mathcal{L}^{\mathsf{O}}$ for which

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, f(x), g(x)) = x] = 1. \tag{2}$$

This will contradict the assumption that f is leakage-resilient one-way.

Let $g\colon \{0,1\}^n \to \{0,1\}^*$ be the following function:

$$g(x) = \mathsf{O}(f(x)) \oplus x.$$

We show that Eq. (2) holds and that g is indeed in $\mathcal{L}^{\mathsf{O}}$. Given $y = f(x)$ and $y' = g(x)$ on a uniform $x \in \{0,1\}^n$, $\mathcal{A}$ can recover x as follows. Apply the random oracle O on y to get $\mathsf{O}(y) = \mathsf{O}(f(x))$ and XOR the output with y'. By the definition of g, the output must be x.

We are left with showing that g is in $\mathcal{L}^{\mathsf{O}}$, that is, it is one-way. Fix $n \in \mathbb{N}$ and let $\mathcal{A}$ be any $q(n)$-query inverter. For $y \in \{0,1\}^*$ and $i \in [q(n)]$ let $Q_i(y)$ be the random variable corresponding to the i-th query made by $\mathcal{A}$ to O when $\mathcal{A}$ is given as input the string y. Let us denote by $\mathsf{Suc}_i(y)$ the event that the i-th query of $\mathcal{A}$ to the random oracle defines a preimage. Namely,

$$\mathsf{Suc}_i(y) = 1 \iff \exists x' \in \{0,1\}^n \colon Q_i(y) = f(x') \text{ and } \mathsf{O}(f(x')) \oplus x' = y$$

Therefore,

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{O}}(y) \in f^{-1}(y)] \leq & \Pr[\mathsf{Suc}_1(y) = 1] \\ & + \sum_{i=1}^{q(n)} \Pr[\mathsf{Suc}_{i+1}(y) = 1 \mid \mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0],\end{aligned}$$

where $y = g(x)$ and the probabilities are taken over the choice of O and over the choice of $x \in \{0,1\}^n$.

To bound the probability of the event $\mathsf{Suc}_1(y) = 1$, notice that

$$\mathbf{Pr}[\mathsf{Suc}_1(y) = 1] \leq \mathbf{Pr}[Q_1(y) = f(x)] + \mathbf{Pr}[\mathsf{Suc}_1(y) = 1 \mid Q_1(y) \neq f(x)]$$

Claim 11. *$\mathbf{Pr}[Q_1(y) = f(x)] \leq \mathsf{neg}(n)$.*

Proof. Recall that $Q_1(y)$ is the *first* query that $\mathcal{A}$ makes to O. Since x is random and O maps every input to a random output, in the view of $\mathcal{A}$, $f(x)$ is distributed uniformly in the distribution of images of f. Since $\mathsf{H}_\infty(f(X)) \geq \omega(\log n)$ (see Theorem 5), it holds that $\mathbf{Pr}[Q_1(y) = f(x)] \leq \mathsf{neg}(n)$.

Claim 12. *$\mathbf{Pr}[\mathsf{Suc}_1(y) = 1 \mid Q_1(y) \neq f(x)] = 1/2^n$.*

Proof. Note that

$$\begin{aligned}&\mathbf{Pr}[\mathsf{Suc}_1(y) = 1 \mid Q_1(y) \neq f(x)]\\ &\quad \leq \mathbf{Pr}[\mathsf{O}(Q_1(y)) = z \oplus \mathsf{O}(f(x)) \oplus x \text{ and } z \in f^{-1}(Q_1(y)) \mid Q_1(y) \neq f(x)].\end{aligned}$$

Since $Q_1(y) \neq f(x)$, then the value $\mathsf{O}(Q_1(y))$ is completely uniform over $\{0,1\}^n$ and independent of $\mathsf{O}(f(x))$. Therefore, the probability that indeed $\mathsf{O}(Q_1(y)) \oplus z = \mathsf{O}(f(x)) \oplus x$, where $z \in f^{-1}(Q_1(y))$, is $1/2^n$.

We use a similar argument to bound the probability that $\mathsf{Suc}_{i+1}(y) = 1$ conditioned on $\mathsf{Suc}_1(y) \ldots, \mathsf{Suc}_i(y) = 0$. Specifically, we bound the expression

$$\begin{aligned}&\mathbf{Pr}[\mathsf{Suc}_{i+1}(y) = 1 \mid \mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0]\\ &\quad \leq \mathbf{Pr}[Q_{i+1}(y) = f(x) \mid \mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0]\\ &\quad + \mathbf{Pr}[\mathsf{Suc}_{i+1}(y) = 1 \mid \mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0 \text{ and } Q_{i+1}(y) \neq f(x)]\end{aligned}$$

Notice that $\mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0$ implies that $Q_1(y) \ldots, Q_i(y) \neq f(x)$. Thus, the view of $\mathcal{A}$ is that $f(x)$ is uniformly distributed in the distribution of images of f except the points $Q_1(y) \ldots, Q_i(y)$ (some of which may not even be valid images). Namely, for $\mathcal{A}$ the value $f(x)$ is uniformly distribution w.r.t the distribution in which one samples a random $x' \leftarrow \{0,1\}^n$, computes $f(x')$ and outputs $f(x')$ conditioned on $f(x') \notin \{Q_1(y) \ldots, Q_i(y)\}$ (otherwise, we sample x' again). This distribution has super-logarithmic min-entropy, namely,

$$\begin{aligned}\mathsf{H}_\infty(f(X) \mid f(X) \notin \{Q_1(y) \ldots, Q_i(y)\}) &\geq \mathsf{H}_\infty(f(X)) - \log i\\ &\geq \omega(\log n),\end{aligned}$$

where the last inequality follows from Theorem 5 and since $i \leq q(n)$ is a polynomial in n. Therefore, as in Theorem 11, we get that

$$\mathbf{Pr}[Q_{i+1}(y) = f(x) \mid \mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0] \leq \mathsf{neg}(n).$$

Given that $\mathsf{Suc}_1(y), \ldots, \mathsf{Suc}_i(y) = 0$ and $Q_{i+1}(y) \neq f(x)$, we have that $Q_{1+1}(y)$ is completely uniform over $\{0,1\}^n$ and independent of $\mathsf{O}(f(x))$ and all previous queries $\mathsf{O}(Q_1(y)),\ldots,\mathsf{O}(Q_i(y))$ (we assume, without loss of generality, that all queries to O are distinct). Therefore, the probability that $\mathsf{O}(Q_{i+1}(y)) \oplus z = \mathsf{O}(f(x)) \oplus x$, where $z \in f^{-1}(Q_{i+1}(y))$, is $1/2^n$. Thus, as in Theorem 12, we have that

$$\mathbf{Pr}[\mathsf{Suc}_{i+1}(y) = 1 \mid \mathsf{Suc}_1(y),\ldots,\mathsf{Suc}_i(y) = 0 \text{ and } Q_{i+1}(y) \neq f(x)] = 1/2^n.$$

In conclusion, since $q(n)$ is a polynomial, we get that

$$\begin{aligned}\mathbf{Pr}[\mathcal{A}^{\mathsf{O}}(y) \in f^{-1}(y)] &\leq \sum_{i=0}^{q(n)} \mathbf{Pr}[\mathsf{Suc}_{i+1}(y) = 1 \mid \mathsf{Suc}_1(y),\ldots,\mathsf{Suc}_i(y) = 0] \\ &\leq \sum_{i=0}^{q(n)} (\mathsf{neg}(n) + 1/2^n) \leq \mathsf{neg}(n).\end{aligned}$$

4.2 Impossibility in the Standard Model

The following theorem shows that there cannot be a leakage resilient one-way function family $\mathcal{F}$ if the leakage function can depend on the function f chosen from $\mathcal{F}$.

Theorem 13. *Let $\mathcal{L} = \{g \colon \{0,1\}^n \to \{0,1\}^*\}$ be the collection of all one-way functions. Assuming a public-coin multi-bit point obfuscator, there is no leakage resilient one-way function collection $\mathcal{F} = \{f \colon \{0,1\}^n \to \{0,1\}^m\}$ for the collection $\mathcal{L}$.*

Proof. Assume towards contradiction that such a function $f \colon \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{F}$ exists. We shall construct a function $g \in \mathcal{L}$ (depending on f) and show that for any $x \in \{0,1\}^n$, $f(x)$ together with $g(x)$ reveal x. Our building block is a public-coin multi-bit point obfuscator MBPO. Assume that MBPO takes as input a pair of strings $(x, y) \in \{0,1\}^m \times \{0,1\}^n$ and randomness of length λ. Let $r \leftarrow \{0,1\}^\lambda$ be a uniformly random string. We define $g_r \colon \{0,1\}^n \to \{0,1\}^m$ that outputs, on input x, a multi-bit point obfuscation of the function $I_{f(x)\to x}$. Namely,

$$g_r(x) = \mathsf{MBPO}(I_{f(x)\to x}; r) \tag{3}$$

For correctness, we argue that given $f(x)$ and $g_r(x)$ together it is easy to recover x. Indeed, one can just plug in $f(x)$ into the output of $g_r(x)$, namely into $\mathsf{MBPO}(I_{f(x)\to x}; r)$. By the correctness of the multi-bit point obfuscator it follows that the output of this operation has to be x.

For security we have to prove that $g_r(x)$ is a one-way function. Namely, given g_r and $g_r(x)$ on a uniformly random x, one cannot recover any x' such that $g_r(x') = g_r(x)$. First, we observe that by the (perfect) correctness of MBPO it holds that for every $x' \neq x$, it cannot be that $\mathsf{MBPO}(I_{f(x)\to x}; r) = \mathsf{MBPO}(I_{f(x')\to x'}; r)$. Thus, g_r is injective. It is left to show that given $g_r(x)$

any computationally bounded adversary cannot recover x with non-negligible probability.

We consider an even easier task for $\mathcal{A}$ of just outputting the first bit of x. By the security of MBPO, we have that for every such adversary $\mathcal{A}$, if there exists a polynomial p such that

$$\mathbf{Pr}[\mathcal{A}(\mathsf{MBPO}(I_{f(x)\to x};r)) = x_1] \geq 1/2 + 1/p(n),$$

then there is an efficient simulator S such that

$$\mathbf{Pr}[S^{I_{f(x)\to x}}(1^n) = x_1] \geq 1/2 + 1/p(n) - \mathsf{neg}(n).$$

However, since $I_{f(x)\to x}$ outputs $\perp$ on all inputs which are not $f(x)$, and since the distribution $f(x)$ has super-logarithmic min entropy (see Theorem 5), any efficient simulator will never query the oracle on $f(x)$ and thus will get no information about x. Hence, it is impossible for it to guess with non-negligible advantage the first bit of x.

5 Possibility of Selective Leakage Resilient One-Way Functions

In both impossibility results (Theorems 10 and 13) we used the fact that the leakage functions can be chosen adaptively and depend on f. In contrast, the following theorem shows that if we limit the choice of the leakage to be independent of f, a leakage resilient one-way function exists based on various assumptions.

The high level idea is that if the leakage g is fixed ahead of time, we can still extract from the input (for f and g) enough pseudorandom bits that will ensure one-wayness.

Theorem 14. *Let $g\colon \{0,1\}^n \to \{0,1\}^m$ be a fixed leakage one-way function. Then, there is a leakage-resilient one-way function $f : \{0,1\}^n \to \{0,1\}^*$ for $\mathcal{L} = \{g\}$ assuming that polynomially-many hardcore bits can be extracted from g.*

Instantiating the theorem with known results we obtain the following corollaries:

1. if g is sub-exponentially secure (with known hardness), then f can be based on any one-way function.
2. if g is a one-way function (with known hardness), then f can be based on any exponentially-secure one-way function.
3. if g is a injective one-way function, then f can be based on indistinguishability obfuscation [4].
4. if g is a one-way function, then f can be based on indistinguishability obfuscation and auxiliary-input point obfuscators [11].
5. if g is a one-way function, then f can be based on exponential hardness of DDH [36].

Proof of Theorem 14. Let g be the leakage function and let $\mathcal{H} = \{h\colon \{0,1\}^n \to \{0,1\}^{2n}\}$ be a family of hardcore function for any one-way function that output polynomially-many hard-core bits. Note that letting the range be $2n$ is without loss of generality since from any polynomial number of hardcore bits we can use a (standard) PRG and obtain the desired length. The leakage resilient one-way function f is defined as follows. We sample a random hard-core function from $\mathcal{H}$ and let

$$f_H(x) = H(x)$$

We argue that $f_H(x)$ is a one-way function even given $g(x)$, where g is a one-way function. For this we use the definition of a hard-core function which says that the distribution

$$(H, H(x), g(x))$$

is computationally indistinguishable from

$$(H, r, g(x)),$$

where $x \leftarrow \{0,1\}^n$, $H \leftarrow \mathcal{H}$, and $r \leftarrow \{0,1\}^{2n}$ are chosen independently uniformly at random. Now, since r is of length $2n$, with all but exponentially small probability, it holds that there is no preimage x' for f_H for which $f_H(x') = r$. Thus, since g is one-way as well, any polynomial-time adversary cannot find a preimage.

6 Future Directions

In this work we introduced and studied leakage resilient one-way functions with arbitrary computationally-hiding leakage. We showed that the natural adaptive definition is impossible to achieve in the random oracle model and in the standard model based on a (non-standard) computation assumption. We further observed that the non-adaptive variant is very related to hardcore functions and in some sense is dual to it.

It is interesting to base the impossibility result on other assumptions (any one-way function, DDH or even based on indistinguishability obfuscation). Also, extracting polynomially-many hardcore bits from any one-way function based on better assumptions is also an interesting problem.

Acknowledgements. We thank Zvika Brakerski, Moni Naor, Gil Segev, and Eylon Yogev for many fruitful discussions on the subject of this paper. We thank the reviewers for their useful comments.

A One-Way Functions Resilient for Bounded Leakage

In both impossibility results (Theorems 10 and 13) we used the fact that the leakage functions can output enough information to allow anyone to invert the original one-way function. In contrast, the theorem below shows that if we limit the image size of the functions in the leakage collection $\mathcal{L}$, a leakage resilient one-way function exists assuming one-way functions exist.

We start with a definition of a lossy function (as defined by [33]). This will capture our restriction on the amount of information the output of the leakage must "lose".

Definition 8 ((n,ℓ)-lossy function). *A function $f\colon \{0,1\}^n \to \{0,1\}^n$ is said to be (n,ℓ)-lossy if its image $\{f(x) \mid x \in \{0,1\}^n\}$ has size at most $2^{n-\ell}$ for every $x \in \{0,1\}^n$.*

Roughly speaking, the parameter ℓ captures the number of information bits f loses about a typical input x. We note that it is enough for us to relax the definition of a lossy function and only require that it has bounded image size on all but a negligible fraction of the x's. We use the stronger requirement for simplicity.

The construction will rely on universal one-way hash functions (UOWHFs) that were introduced by Naor and Yung [32]. The main feature of UOWHFs is that given an element x in the domain, it is computationally hard to find a *different* domain element $x' \neq x$ which collides with x. Naor and Yung showed how to use UOWHFs to construct digital signatures. Besides this application, they showed how to construct them using any injective one-way function. Later, Rompel [35] showed how to construct UOWHFs from any one-way function (see also [28]).

In the following definition we define a weak variant of UOWHFs in which the initial domain element is a uniform random input (rather than an adversarially chosen input).[8] The goal of the adversary is then to find a collision with that random input.

Definition 9 ((Weak) universal one-way hash functions). *Let $p(n) = n^{1/c}$ be a polynomial where $c \in \mathbb{N}$ is a constant. A collection of functions $\{F_h\colon \{0,1\}^n \to \{0,1\}^{p(n)}\}$ mapping strings of length n to strings of length $p(n)$ is a (collection of) universal one-way hash functions if it is described by a pair of efficient algorithms $\mathsf{UOWHF} = (\mathsf{Gen}, F)$ with the following properties.*

1. *Gen is a probabilistic algorithm that is given as input the unary value of n, and it outputs a function index h.*
2. *For every function index h in the image of Gen, F_h is given as input $x \in \{0,1\}^n$ and it outputs a string of length $p(n)$.*
3. *For every probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\mathsf{neg}(\cdot)$, such that*

$$\mathbf{Pr}[x' \leftarrow \mathcal{A}(h, x, F_h(x))\colon x \neq x' and\ F_h(x') = F_h(x)] \leq \mathsf{neg}(n),$$

where the probability is over the choice of $x \leftarrow \{0,1\}^n$, the choice of $h \leftarrow \mathsf{Gen}(1^n)$, and the internal randomness of $\mathcal{A}$.

[8] This is sometimes called a *second pre-image resistant function.*

Theorem 15 ([3,17]). *Let $k = n^{1/c}$ for a constant $c \in \mathbb{N}$ and let $\kappa = \omega(\log n)$. Let $\mathcal{F} = \{F_h \colon \{0,1\}^n \to \{0,1\}^{k-\kappa}\}$ be a family of universal one-way hash functions mapping strings of length n to strings of length $k-\kappa$ described by (Gen, F). Let $\mathcal{L} = \{g \colon \{0,1\}^n \to \{0,1\}^n\}$ be the collection of* all *(n,k)-lossy functions. Then, $\mathcal{F}$ is a leakage resilient one-way function collection for $\mathcal{L}$.*

The proof uses the notion of average min-entropy defined by Dodis et al. [19] which captures the remaining unpredictability of X conditioned on the value of Y. Roughly speaking, the average min-entropy of X given Y is the logarithm of the average probability of the most likely value of X given Y. That is,

$$\tilde{\mathsf{H}}_\infty(X \mid Y) = -\log\left(\mathop{\mathbf{E}}_{y \leftarrow Y}[2^{-\mathsf{H}_\infty(X|Y=y)}]\right).$$

The following property of average min-entropy was shown by Dodis et al. [19].

Lemma 1 ([19, Lemma 2.2]). *Let X and Y be two random variables. Then,*

1. *For any $\delta > 0$, it holds that*

$$\Pr_{y \leftarrow Y}[\mathsf{H}_\infty(X \mid Y = y) \geq \tilde{\mathsf{H}}_\infty(X \mid Y) - \log(1/\delta)] \geq 1 - \delta.$$

2. *If Y has at most 2^k possible values, then $\tilde{\mathsf{H}}_\infty(X \mid Y) \geq \mathsf{H}_\infty(X) - k$.*

Proof of Theorem 15. We assume towards contradiction that the statement is false. Namely, there exists a function $g \colon \{0,1\}^n \to \{0,1\}^{n-k}$ for which there exists an adversary $\mathcal{A}$ such that for $x^* \leftarrow \{0,1\}^n$ chosen uniformly at random given

$$h, F_h(x^*), g(x^*),$$

where $h \leftarrow \mathsf{Gen}(1^n)$, $\mathcal{A}$ is able to recover any x such that $F_h(x^*) = F_h(x)$ with non-negligible probability $1/p(n)$. We use this adversary $\mathcal{A}$ and construct an adversary $\mathcal{B}$ that breaks the security of the universal one-way hash function.

Let $h, x^*, F_h(x^*)$ be a challenge for the universal one-way hash function, where $h \leftarrow \mathsf{Gen}(1^n)$ and $x^* \leftarrow \{0,1\}^n$ is chosen uniformly and independently. Our adversary $\mathcal{B}$ will first simulate the choice of g and compute $g(x^*)$. Then, it runs the inverter $\mathcal{A}$ on input $(h, F_h(x^*), g(x^*))$ and obtains a preimage x. Finally, $\mathcal{B}$ outputs x as its guess for the collision. We now argue that this adversary indeed breaks the security of the UOWHF. First, it is clear by the correctness of the adversary $\mathcal{A}$ that $F_h(x) = F_h(x^*)$. We are left to argue that $x \neq x^*$ with non-negligible probability.

Roughly speaking, the idea is that since x^* is chosen uniformly at random, given only $F_h(x^*)$ and $g(x^*)$, whose image size altogether $\ll 2^n$, there is not enough information regarding the real x^* that maps to $F_h(x^*)$ and $g(x^*)$. Namely, we will show that with high probability over the choice of x^*, there could be many consistent x's that map to the same output. The inverted cannot distinguish between them and thus will output the real x^* with very small probability. We formalize this intuition next.

Fix the function index $h \leftarrow \mathsf{Gen}(1^n)$ and leakage function g (that might depend h). Since $F_h(x^*)$ and $g(x^*)$ have together at most $2^{k-\kappa} \cdot 2^{n-k} = 2^{n-\kappa}$ possible outputs and $x^* \leftarrow \{0,1\}^n$ is uniform and independent of h, by item 2 of Lemma 1 we have that

$$\tilde{\mathsf{H}}_\infty(x^* \mid h, F_h(x^*), g(x^*)) \geq \mathsf{H}_\infty(x^* \mid h) - (n - \kappa) = \kappa.$$

By item 1 of Lemma 1, we get that for any $\delta > 0$, it holds that

$$\begin{aligned}\underset{x^* \leftarrow \{0,1\}^n}{\mathbf{Pr}}[\mathsf{H}_\infty(x^* \mid h, F_h(x^*), g(x^*)) \geq \tilde{\mathsf{H}}_\infty(x^* \mid h, F_h(x^*), g(x^*)) - \log(1/\delta)]\\ \geq 1 - \delta.\end{aligned}$$

Therefore,

$$\underset{x^* \leftarrow \{0,1\}^n}{\mathbf{Pr}}[\mathsf{H}_\infty(x^* \mid h, F_h(x^*), g(x^*)) \geq \kappa - \log(1/\delta)] \geq 1 - \delta. \tag{4}$$

Let $\delta = 1/2^{\kappa/2}$. Then, with all but a negligible probability over the choice of x^*, it holds that

$$\mathsf{H}_\infty(x^* \mid h, F_h(x^*), g(x^*)) \geq \kappa - \kappa/2 = \kappa/2.$$

Therefore, since $\kappa = \omega(\log n)$, by the definition of min-entropy $\mathbf{Pr}[x^* \leftarrow \mathcal{A}(h, F_h(x^*), g(x^*))] \leq \mathsf{neg}(\cdot)$. In conclusion, the adversary $\mathcal{B}$ is able to find a collision with non-negligible probability:

$$\begin{aligned}&\mathbf{Pr}[x \leftarrow \mathcal{B}(h, F_h(x^*), g(x^*))\colon x \neq x^* \text{ and } F_h(x^*) = F_h(x), g(x^*) = g(x)]\\ &\quad = \mathbf{Pr}[x \leftarrow \mathcal{A}(h, F_h(x^*), g(x^*))\colon F_h(x^*) = F_h(x), g(x^*) = g(x)]\\ &\quad - \mathbf{Pr}[x^* \leftarrow \mathcal{A}(h, F_h(x^*), g(x^*))] \geq\\ &\quad 1/p(n) - \mathsf{neg}(n) \geq 1/(2p(n)).\end{aligned}$$

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00457-5_28
2. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_3
3. Alwen, J., Dodis, Y., Wichs, D.: Survey: leakage resilience and the bounded retrieval model. In: Kurosawa, K. (ed.) ICITS 2009. LNCS, vol. 5973, pp. 1–18. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14496-7_1

4. Bellare, M., Stepanovs, I., Tessaro, S.: Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 102–121. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_6
5. Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. J. Cryptol. **27**(2), 317–357 (2014)
6. Boyle, E., Goldwasser, S., Jain, A., Kalai, Y.T.: Multiparty computation secure against continual memory leakage. In: Proceedings of the 44th Symposium on Theory of Computing Conference, STOC, pp. 1235–1254 (2012)
7. Boyle, E., Segev, G., Wichs, D.: Fully leakage-resilient signatures. J. Cryptol. **26**(3), 513–558 (2013)
8. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_1
9. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: public-key cryptography resilient to continual memory leakage. In: 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 501–510 (2010)
10. Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 142–161. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_8
11. Brzuska, C., Mittelbach, A.: Using indistinguishability obfuscation via UCEs. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 122–141. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_7
12. Canetti, R.: Towards realizing random oracles: hash functions that hide all partial information. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997). doi:10.1007/BFb0052255
13. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78967-3_28
14. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000). doi:10.1007/3-540-45539-6_33
15. Davì, F., Dziembowski, S., Venturi, D.: Leakage-resilient storage. In: Garay, J.A., Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 121–137. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15317-4_9
16. Dodis, Y., Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11799-2_22
17. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 511–520 (2010)
18. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC, pp. 621–630 (2009)
19. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)

20. Dodis, Y., Pietrzak, K.: Leakage-resilient pseudorandom functions and side-channel attacks on Feistel networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 21–40. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_2
21. Dodis, Y., Sahai, A., Smith, A.: On perfect and adaptive security in exposure-resilient cryptography. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 301–324. Springer, Heidelberg (2001). doi:10.1007/3-540-44987-6_19
22. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 293–302 (2008)
23. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-resilient signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11799-2_21
24. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from computationally bounded and noisy leakage. SIAM J. Comput. **43**(5), 1564–1614 (2014)
25. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, STOC, pp. 25–32 (1989)
26. Goldwasser, S., Kalai, Y.T.: On the impossibility of obfuscation with auxiliary input. In: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 553–562 (2005)
27. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45146-4_27
28. Katz, J., Koo, C.: On constructing universal one-way hash functions from arbitrary one-way functions. IACR Cryptology ePrint Archive, p. 328 (2005)
29. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10366-7_41
30. Micali, S., Reyzin, L.: Physically observable cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24638-1_16
31. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. SIAM J. Comput. **41**(4), 772–814 (2012)
32. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, STOC, pp. 33–43 (1989)
33. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)
34. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_27
35. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, STOC, pp. 387–394 (1990)
36. Zhandry, M.: The magic of ELFs. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 479–508. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53018-4_18