Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 45 (2015)

Minimizing Locality of One-Way Functions via Semi-Private Randomized Encodings

Benny Applebaum* Yuval Ishai† Eyal Kushilevitz‡

August 29, 2016

Abstract

A one-way function is d-local if each of its outputs depends on at most d input bits. In [3] it was shown that, under relatively mild assumptions, there exist 4-local one-way functions (OWFs). This result is not far from optimal as it is not hard to show that there are no 2-local OWFs. The gap was partially closed in [3] by showing that the existence of 3-local OWFs is implied by the intractability of decoding a random linear code (or equivalently the hardness of learning parity with noise).

In this note we provide further evidence for the existence of 3-local OWFs. We construct a 3-local OWF based on the assumption that a random function of (arbitrarily large) constant locality is one-way. (A closely related assumption was previously made by Goldreich [15].) Our proof consists of two steps: (1) We show that, under the above assumption, random local functions remain hard to invert even when some information on the preimage x is leaked; and (2) Such "robust" local one-way functions can be converted to 3-local one-way functions via a new construction of *semi-private randomized encoding*. We believe that these results may be of independent interest.

1 Introduction

How simple can a one-way function (OWF) be? In this paper we measure simplicity in terms of *output locality*. We say that a function $f : \{0,1\}^* \to \{0,1\}^*$ is d-local if each output bit of f depends on at most d input bits. When d is fixed (and does not grow with the input length), d-local functions can be computed by constant-depth circuits with bounded fan-in gates as captured by the complexity class $\mathbf{NC^0}$. The existence of one-way functions in $\mathbf{NC^0}$ was established in [3] based on various standard cryptographic assumptions, or more generally, based on the existence of log-space computable one-way functions. In this paper, we ask:

What is the minimal locality d for which d-local one-way functions exist?

*School of Electrical Engineering, Tel-Aviv University, bennyap@post.tau.ac.il. Supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, ISF grant 1155/11, and the Check Point Institute for Information Security.

†Department of Computer Science, Technion, yuvali@cs.technion.ac.il. Supported by the European Research Council as part of the ERC project CaC (grant 259426), ISF grant 1709/14 and BSF grant 2012378.

‡Department of Computer Science, Technion, eyalk@cs.technion.ac.il. Supported by ISF grant 1709/14 and BSF grant 2012378.

ISSN 1433-8092

While it is known that 2-local functions cannot be one-way [15], the results of [3] yield 4-local one-way functions. This small gap between 2 and 4 is partially resolved by [3, 4] who show that, assuming the intractability of decoding a random linear code (or equivalently the intractability of *learning parity with noise*), there exist one-way functions with locality 3. In this paper, we provide further evidence for the existence of 3-local one-way functions by constructing such functions based on the one-wayness of *random functions with large (expected) constant locality.*

Assumption 1.1 (Random Local Functions are OWFs - Informal)**.** *For a locality parameter $d \in \mathbb{N}$, sample a Boolean circuit $f : \{0,1\}^n \to \{0,1\}^n$ by connecting each output gate y_i to each input variable x_j independently at random with probability d/n, and by placing at each output-gate a random predicate P_i chosen uniformly from all the predicates of appropriate arity. Then, the resulting function is weakly one-way.*

The assumption asserts that any efficient adversary that gets the description of f fails to invert it (on a randomly chosen input) with some noticeable probability. Formally, this means that the corresponding collection is weakly one-way, cf. [16, Definition 2.4.3]. Our collection of functions, denoted by $\mathcal{F}_{n,m,d/n}$, is closely related to Goldreich's candidate one-way function [15], whose one-wayness was extensively studied in the last few years. (See [2] for a survey). Further details about the exact formulation of our assumption appear in Section 5.

We prove the following theorem:

Informal Theorem 1.2 (main)**.** *Assume that, for some large constant d, Assumption 1.1 holds. Then, one-way functions with locality 3 exist.*

2 Techniques

Background. Before introducing our techniques it is instructive to review the concept of *randomized encoding* [20, 21] and its role in the original construction of [3]. Roughly speaking, a function $\hat{f}(x,r)$ is a randomized encoding of a function $f(x)$ if:

1. (Correctness) For every fixed input x and a uniformly random choice of r, the output distribution $\hat{f}(x,r)$ forms a "randomized encoding" of $f(x)$, from which $f(x)$ can be decoded;
2. (Privacy) The distribution of this randomized encoding depends only on the encoded value $f(x)$ and does not further depend on x.

In [3] it is shown that randomized encoding preserves the security of many cryptographic primitives. Concretely, if $\hat{f}(x,r)$ is an encoding of a one-way function $f(x)$, then the function $\hat{f}(x,r)$ is also one-way. This gives rise to the following general template for constructing cryptography with low complexity: First, show how to encode functions f in a relatively complicated complexity class **Strong** by functions $\hat{f}$ in some low complexity class **Weak**, and then conclude that the existence of OWFs in **Strong** implies the existence of OWFs in **Weak**.

This framework was instantiated in [3] by letting **Strong** be the class of log-space computable functions, and taking **Weak** to be the class of 4-local functions. The underlying encoding was based on: (1) a result of [20, 21] which shows that log-space computable functions can be encoded by functions that each of their outputs is computed by a degree 3 polynomial over GF(2); and (2) a locality reduction lemma [3] that transforms degree-d encodings into encodings of locality $d+1$.

Based on this template, it is natural to try and reduce the locality to 3 by constructing a degree 2 encoding for some OWF. Unfortunately, we do not know whether such an encoding exists. In fact, the results of [20] provide some evidence against the prospects of this general approach, ruling out the existence of degree-2 *perfectly private* encodings for most nontrivial functions. Instead, we will rely on a new *weaker* variant of *semi-private* randomized encoding that admits degree-2 instantiation. To compensate the weaker privacy property of the encoding, we will have to rely on a stronger form of one-wayness. Details follow.

2.1 Semi-private Randomized Encoding and Robust OWF

Let $\hat{f}$ be a randomized encoding of a Boolean function $f : \{0,1\}^n \to \{0,1\}$. Recall that, according to the privacy property, the output distribution of $\hat{f}(x,r)$ (induced by a uniform choice of r) should hide all the information about x except for the value $f(x)$. Semi-privacy relaxes this requirement by insisting that the input x remain hidden by $\hat{f}(x,r)$ only in the case that $f(x)$ takes some specific value, say 0. (If $f(x)$ is different from this value, $\hat{f}(x,r)$ fully reveals x.) As it turns out, this relaxed privacy requirement is sufficiently liberal to allow a degree-2 encoding of general boolean functions.

Given any OWF $f : \{0,1\}^n \to \{0,1\}^m$, one could attempt to apply a semi-private encoding as described above to every output bit of f, obtaining a degree-2 function $\hat{f}$. However, $\hat{f}$ will typically not be one-way: every output bit of f that evaluates to 1 might reveal the entire input. This motivates the following notion of a *robust* OWF. Loosely speaking, a OWF f is said to be robust if it remains (weakly) hard to invert even if a random subset of its output bits are "exposed", in the sense that all input bits leading to these outputs are revealed.

Formally, consider the following inversion game. First, we choose a random input $x \in \{0,1\}^n$, compute $y = f(x)$ and send it to the adversary. Then, for each output bit of f we toss a coin b_i. If $b_i = 1$, we allow the adversary to see the bits of x that influence the i-th output bit. That is, we send $(x_{K(i)}, i, b_i)$ to the adversary, where $x_{K(i)}$ is the restriction of x to the set $K(i) \subseteq [n]$ of inputs that affects the i-th output bit. If $b_i = 0$, we reveal nothing regarding x and send (i, b_i) to the adversary. The adversary wins the game if she finds a preimage x' which is consistent with the information given to her, i.e., $f(x') = f(x)$, and $x'_{K(i)} = x_{K(i)}$ whenever $b_i = 1$. The function is *robust one-way* if, for some polynomial $p(\cdot)$, any efficient adversary fails to find a consistent preimage with probability at least $1/p(n)$. (See Section 3 for a formal definition and Section 6 for a discussion on a leakage-resilience interpretation of this notion.)

Intuitively, the purpose of the robustness requirement is to guarantee that the information leaked by the semi-private encoding leaves enough uncertainty about the input to make inversion difficult. Indeed, we show that when semi-private randomized encoding (SPRE) is applied to a (slightly modified) robust OWF the resulting function is distributionally one-way. Hence, a construction of degree-2 SPRE can be used to convert a robust OWF to a distributionally OWF with degree 2. Furthermore, it turns out that it is possible to convert the latter to a standard OWF with similar degree.

2.2 Constructing Robust OWF

We construct a robust OWF under the assumption that the collection $\mathcal{F}_{n,m,d/n}$ of random function $f : \{0,1\}^n \to \{0,1\}^m$ of expected locality d is weakly one-way. Recall that the circuit that computes

f is sampled by connecting each output gate y_i to each input variable x_j independently at random with probability d/n, and by placing at each output-gate a random predicate P_i chosen uniformly from all the predicates of appropriate arity. (More generally, our results apply to any efficiently samplable distribution ensemble over predicates which is invariant under permutation of the input variables and under partial fixing; see Section 5 for details.)

We show that, if $\mathcal{F}_{n,n,d/n}$ is one-way for some constant d, then, for some related parameters $n' = e^d n, m' = 2n, d' = e^d d$, a randomly chosen function $h \stackrel{R}{\leftarrow} \mathcal{F}_{n',m',d'/n'}$ is robust one-way. Roughly speaking, the circuit of h embeds many small copies of f, and so inverting h in the presence of random exposure, is as hard as inverting f. To get some intuition, assume that exactly half of the outputs of $h \stackrel{R}{\leftarrow} \mathcal{F}_{n',m',d'/n'}$ are exposed. Then there are exactly n exposed outputs and n unexposed outputs. In this case, an input node is not exposed (i.e., is *not* connected to an exposed output) with probability exactly $(1-d'/n')^n = (1-d/n)^n \approx e^{-d}$. Hence, the expected number of unexposed inputs is n, and so the circuit of h contains a "non-exposed sub-circuit" f with n inputs and n outputs. It turns out that the function $f \stackrel{R}{\leftarrow} \mathcal{F}_{n,n,d/n}$ can be embedded in this sub-circuit. (See Section 5 for further details.)

Organization

Section 3 provides some preliminaries including a generalization of statistical randomized encoding. Semi-private randomized encodings are defined, constructed and analyzed in Section 4. In Section 5 we define the notion of robust one-way functions, and present a construction based on the hardness of inverting a random local function. We conclude in Section 6 with a discussion and some open questions.

3 Preliminaries

Probability notation. Let $\mathcal{U}_n$ denote a random variable that is uniformly distributed over $\{0,1\}^n$. Different occurrences of $\mathcal{U}_n$ in the same statement refer to the same random variable (rather than independent ones). If X is a probability distribution, we write $x \stackrel{R}{\leftarrow} X$ to indicate that x is a sample taken from X. If S is a set, we write $x \stackrel{R}{\leftarrow} S$ to indicate that x is uniformly selected from S. The *statistical distance* between discrete probability distributions X and Y is defined as $\|X - Y\| = \frac{1}{2}\sum_z |\Pr[X = z] - \Pr[Y = z]|$. Equivalently, the statistical distance between X and Y may be defined as the maximum, over all boolean functions T, of the *distinguishing advantage* $|\Pr[T(X) = 1] - \Pr[T(Y) = 1]|$. A function $\varepsilon(\cdot)$ is said to be *negligible* if $\varepsilon(n) < n^{-c}$ for any $c > 0$ and sufficiently large n. For two distribution ensembles $X = \{X_n\}$ and $Y = \{Y_n\}$, we write $X \equiv Y$ if X_n and Y_n are identically distributed, and $X \stackrel{s}{\equiv} Y$ if the two ensembles are *statistically indistinguishable*; namely, $\|X_n - Y_n\|$ is negligible in n.

We make use of the following standard fact about the binomial distribution.

Fact 3.1. *Let $X = X_1 + \ldots + X_n$ where $X_i, i \in [n]$ are identically distributed in $[0,1]$ with mean p.*

1. *If $p \in (0,1)$ is a constant then $\Pr[X = \lfloor p(n+1) \rfloor] \geq \Omega(1/\sqrt{n})$.*
2. *If $p = O(1/n)$ then $\Pr[X > \log n] < (O(1/\log n))^{\log n} = n^{-\omega(1)}$.*

Proof. (1) By a standard additive Chernoff bound (see, e.g., [11, Theorem 1.1]) X falls in the interval $pn \pm \sqrt{n}$ with probability of $1 - 2e^{-2} > 2/3$. Since the mode of the binomial distribution is $\lfloor p(n+1) \rfloor$, it follows that $\Pr[X = \lfloor p(n+1) \rfloor] > \frac{2/3}{2\sqrt{n}} = \Omega(1/\sqrt{n})$.

(2) By a union bound,

$$\Pr[X > \log n] \leq \sum_{S \subset [n], |S| = \log n} \Pr\left[\bigwedge_{i \in S} X_i = 1\right] = \binom{n}{\log n} p^{\log n},$$

the theorem follows by plugging in $p = O(1/n)$ and by using the standard upper-bound $\binom{n}{k} \leq (\frac{en}{k})^k$. □

Locality and Degree. Let $f : \{0,1\}^n \to \{0,1\}^l$ be a function. We say that the i-th output variable y_i *depends* on the j-th input variable x_j (or equivalently, x_j *affects* the output y_i) if there exists a pair of input strings which differ only on the j-th location whose images differ on the i-th location. The locality of an output variable is the number of inputs on which it depends. We say that an output variable has degree d if it can be expressed as a multivariate polynomial of degree d in the input variables over the binary field $\mathbb{F}_2$. The locality of an output variable trivially upper bounds its degree.

Collection of Functions. We model cryptographic primitives as collections of functions $\mathcal{F} = \left\{f_k : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{k \in \{0,1\}^{s(n)}}$ equipped with a pair of efficient algorithms: (1) an evaluation algorithm which given $(k \in \{0,1\}^s, x \in \{0,1\}^n)$ outputs $f_k(x)$; and (2) a key-sampling algorithm $\mathcal{K}$ which given 1^n samples a index $k \in \{0,1\}^{s(n)}$. We will sometimes keep the key-sampler implicit and write $f \xleftarrow{R} \mathcal{F}$ to denote the experiment where $k \xleftarrow{R} \mathcal{K}(1^n)$ and $f = f_k$. A collection of functions has *constant locality* if there exists a constant d which does not grow with n such that for every fixed k each output of the function f_k has locality of at most d. Similarly, the collection has *constant algebraic degree* of d if for every fixed k each output of the function f_k has degree of at most d. (Intuitively, we distinguish between the complexity of sampling a key – which is a one-time operation that can be preprocessed – and the complexity of computing the function given a preprocessed key.) When $\mathcal{F}$ is used as a cryptographic primitive we will always assume that the adversary that tries to break it gets the collection index as a public parameter. Moreover, our constructions are all in the "public-coin" setting, and so they remain secure even if the adversary gets the coins used to sample the index of the collection.

3.1 One-Way Functions and Collections

We review several variants of collections of one-way functions (OWFs).

Definition 3.2 (One-way function)**.** *Let $\mathcal{F} = \left\{f_k : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{k \in \{0,1\}^{s(n)}}$ be a collection of functions equipped with key-sampling algorithm $\mathcal{K}$. Then,*

- **Strongly hard to invert.** *The collection $\mathcal{F}$ is* strongly hard to invert *if for every (non-uniform) polynomial-time algorithm, A, the probability $\Pr_{k \xleftarrow{R} \mathcal{K}(1^n)}[A(1^n, k, f_k(\mathcal{U}_n)) \in f_k^{-1}(f_k(\mathcal{U}_n))]$ is negligible in n.*

- **Weakly hard to invert.** *The collection $\mathcal{F}$ is* weakly hard to invert *if there exists a polynomial $p(\cdot)$, such that for every (non-uniform) polynomial-time algorithm, A, and all sufficiently large n's* $\Pr_{k \xleftarrow{R} \mathcal{K}(1^n)}[A(1^n, k, f_k(\mathcal{U}_n)) \notin f_k^{-1}(f_k(\mathcal{U}_n))] > \frac{1}{p(n)}$.

- **Distributionally hard to invert.** *The collection $\mathcal{F}$ is* distributionally hard to invert *if there exists a positive polynomial $p(\cdot)$ such that for every (non-uniform) polynomial-time algorithm, A, and all sufficiently large n's, $\|(A(1^n, k, f_k(\mathcal{U}_n)), k, f_k(\mathcal{U}_n)) - (\mathcal{U}_n, k, f_k(\mathcal{U}_n))\| > \frac{1}{p(n)}$, where $k \xleftarrow{R} \mathcal{K}(1^n)$.*

If $\mathcal{F}$ can be computed and sampled in polynomial-time and it is also hard to invert (resp. weakly hard to invert, distributionally hard to invert) then it is called strongly one-way *(resp.* weakly one-way*,* distributionally one-way*). By default, a* one-way function *is strongly one-way.*

Note that the first variant defined above is the standard notion of OWF (adopted to the case of collections).

Remark 3.3 (Collection vs. Single Functions)**.** *The (more standard) setting of a single function $f : \{0,1\}^* \to \{0,1\}^*$ is derived from the above definitions by considering the special case in which the ensemble $\mathcal{F}$ contains, for each input length, a single function f_n which corresponds to f restricted to n-bit inputs. It is also useful to observe that the notion of one-way collections (for all the above variants of one-wayness) can be derived from the (standard) single-function setting, by collapsing the collection $\mathcal{F}$ and the key-sampling algorithm $\mathcal{K}$ to a single mapping F which, given an input x and key-sampling randomness ρ, outputs value $f_{\mathcal{K}(1^n;\rho)}(x)$ together with the randomness ρ. It is not hard to see that F is one-way (resp., weak one-way, distributional one-way) if and only if the collection $\mathcal{F}$ is one-way (resp., weak one-way, distributional one-way). This view of collections allows us to easily translate results from the single function setting to the collection setting. To simplify the presentation, we will mostly state our claims for the single function setting with the understanding that they generalize to collections.*

The following lemma from [3, Lemma 8.2] (building on [19, 28]) shows how to transform a degree-2 distributionally one-way collection into a strongly one-way collection with degree 2 and locality 3.

Lemma 3.4. *A degree-2 distributional OWF collection implies a degree-2 OWF collection with locality 3.*

3.2 Randomized Encoding

We will need the following form of randomized encoding [20, 3].

Definition 3.5 (Randomized encoding (RE))**.** *Let $h : \{0,1\}^n \to \{0,1\}^{l(n)}$ be an efficiently computable function. We say that the function $f : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}$ is a δ-correct, ε-private* randomized encoding *of h, if it satisfies the following:*

- **δ-correctness.** *There exists a deterministic*[1] *algorithm B, called a* decoder*, such that for every input $x \in \{0,1\}^n$, $\Pr[B(1^n, f(x, \mathcal{U}_{m(n)})) \neq h(x)] \leq \delta(n)$.*

[1] We restrict the decoder to be deterministic for simplicity. This restriction does not compromise generality, in the sense that one can transform a randomized decoder to a deterministic one by incorporating the coins of the former in the encoding itself.

- **ε-privacy.** *There exists an efficient randomized algorithm S, called a* simulator*, such that for every $x \in \{0,1\}^n$, $\|S(h(x)) - f(x,\mathcal{U}_{m(n)})\| \leq \varepsilon(n)$.*

By default, we think of $\varepsilon(n)$ and $\delta(n)$ as negligible functions, and, in this case, refer to f as a statistical encoding of h.

We will further use a generalized form of statistical randomized encoding defined as follows.

Definition 3.6 (Generalized statistical randomized encoding)**.** *Let $h : \{0,1\}^n \to \{0,1\}^{l(n)}$ be an efficiently computable function. We say that $g : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}$ is a* generalized statistical randomized encoding *(GSRE) of h if there exists a function $f : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}$ such that:*

- *f is a statistical randomized encoding of h.*
- *f is* isomorphic *to g in the sense that there exists permutation $\pi : \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ such that for every $x \in \{0,1\}^{n+m}$, $f(x) = g(\pi(x))$; and π is efficiently computable and efficiently invertible (in time* $\mathrm{poly}(n)$*).*[2]

It was proven in [3] that RE preserves cryptographic hardness. Specifically, it was shown that if a function h is weakly one-way then its statistical encoding g is distributionally one-way [3, Lemma 5.5]. Below we show that a similar result holds for the case of generalized statistical encoding.

Lemma 3.7. *If h is a weak-OWF and g is a GSRE of h, then g is a distributional OWF.*

Proof. Let f be a (standard) statistical encoding of h which is isomorphic to g via the isomorphism π. By [3, Lemma 5.5], f is a distributional one-way function. We will show that g is a distributional one-way function as well. To simplify notation, we view f and g as deterministic functions that take a single input $x \in \{0,1\}^n$ and map it into an $l(n)$-bit string.

Let A be an efficient adversary which distributionally inverts g with error $\varepsilon(n)$, i.e.,

$$\varepsilon(n) = \|(A(1^n, g(y)), g(y)) - (y, g(y))\|,$$

where y is uniformly chosen from $\{0,1\}^n$. Consider the adversary $\hat{A}$ which given $(1^n, z)$ invokes A on the same input, and translates the result y' to $x' = \pi^{-1}(y')$. It is not hard to see that $\hat{A}$ breaks f with the same advantage as A. Formally, we should show that

$$\|(\hat{A}(1^n, f(x)), f(x)) - (x, f(x))\| \leq \varepsilon(n), \tag{1}$$

where x is uniformly chosen from $\{0,1\}^n$. Let $z = f(x)$ and $y = \pi(x)$. Consider the mapping T which given a pair of strings $a \in \{0,1\}^n, b \in \{0,1\}^{l(n)}$ outputs $(\pi^{-1}(a), b)$. By our assumption on A (and by noting that applying a deterministic function cannot increase the statistical distance between distributions), we have that

$$\|T(A(1^n, g(y)), g(y)) - T(y, g(y))\| \leq \varepsilon(n).$$

However, by definition, $T(A(1^n, g(y)), g(y)) = (\hat{A}(1^n, f(x)), f(x))$ and, similarly, $T(y, g(y)) = (x, f(x))$, and so Eq. 1 follows. □

[2] In fact, for our proofs we only need the ability to invert π efficiently. Still, we find it natural to require efficient evaluation in the forward direction as well.

4 Semi-private Randomized Encoding

In this section we define the notion of semi-private randomized encoding (SPRE), describe a construction of degree-2 SPRE for functions with low (logarithmic) locality, and analyze the effect of encoding a function with SPRE.

4.1 Definition and Construction

The following definition relaxes the notion of randomized encoding.

Definition 4.1 (Semi-private randomized encoding (SPRE))**.** *Let $g : \{0,1\}^n \to \{0,1\}$ be a boolean function. We say that a function $\hat{g} : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}$ is a* semi-private randomized encoding *(SPRE) of g with error ε if the following conditions hold:*

- **ε-correctness.** *There exists a polynomial-time decoder B, such that for every $x \in \{0,1\}^n$ it holds that $\Pr[B(1^n, \hat{g}(x, \mathcal{U}_{m(n)})) \neq g(x)] < \varepsilon(n)$.*
- **One-sided privacy.** *There exists a probabilistic polynomial-time simulator S_0, such that for every $x \in \{0,1\}^n$ such that $g(x) = 0$ it holds that $S_0(1^n) \equiv \hat{g}(x, \mathcal{U}_{m(n)})$.*
- **One-sided exposure.** *There exists a polynomial-time exposure algorithm E_1, such that for every $x \in \{0,1\}^n$ such that $g(x) = 0$ it holds that $\Pr[E_1(1^n, \hat{g}(x, \mathcal{U}_{m(n)})) \neq x] < \varepsilon(n)$.*

By default, ε is a negligible function in n.

Remark 4.2 (The role of one-sided exposure)**.** *The exposure requirement asserts that the encoding should reveal the input x when g evaluates to zero. This property will be used later in our reductions to argue that inverting an SPRE of g is similar to inverting an exposed version of g. Departing from the current context, the exposure property allows us to treat SPRE as a variant of* conditional disclosure of secrets *(CDS) [14] where x should be revealed only if the condition $g(x) = 0$ holds and, otherwise, it should be kept secret.*[3] *This is a natural functionality which may turn to be useful in future contexts.*

We turn to the question of constructing degree-2 SPRE. Since we will only need to encode functions that depend on a small number of inputs, it will be convenient to construct such an encoding based on the DNF representation of the function.

Construction 4.3 (SPRE for canonic DNF)**.** *Let $g : \{0,1\}^d \to \{0,1\}$ be a boolean function. Let $\bigvee_{i=1}^{k} T_i$ be its unique canonic DNF representation. That is, for each $\alpha \in \{0,1\}^d$ such that $g(\alpha) = 1$ there exists a corresponding term $T_i(x)$ which evaluates to 1 if and only if $x = \alpha$. We encode such T_i by the degree-2 function $\hat{T}_i(x, r) = \langle x - \alpha, r \rangle$, where $\langle \cdot, \cdot \rangle$ denotes inner product over* GF(2)*. Let t be some integer (later used as a security parameter). Then, the degree-2 function $\hat{g}$ is defined by concatenating t copies of $\hat{T}_i$ (each copy with independent random inputs $r_{i,j}$) for each of the k terms. Namely,*

$$\hat{g}(x, (r_{i,j})_{i \in [k], j \in [t]}) \stackrel{def}{=} \left((\hat{T}_1(x, r_{1,j}))_{j=1}^{t}, \ldots, (\hat{T}_k(x, r_{k,j}))_{j=1}^{t} \right),$$

[3] In CDS (or the related notion of "partial garbling" [24, 13]) the input contains a message s and a label v. The encoding always reveals the label v, and releases the message s only if v satisfies the underlying predicate. The difference between CDS and SPRE parallels the difference between *attribute based encryption* [26, 17] and *predicate encryption* [7, 27, 25, 18].

where $r_{i,j} \in \{0,1\}^d$.

Lemma 4.4. *The encoding* $\hat{g}$ *(defined in Construction 4.3) is an SPRE of* g *with error* $k \cdot 2^{-t}$. *The time complexity of the simulator, decoder, and exposing algorithms is* $\mathrm{poly}(tkd)$.

Proof. Let $\bigvee_{i=1}^{k} T_i$ be the canonic DNF representation of g. We view the variables $r_{i,j}$ of $\hat{T}$ as the random input of the encoding. Observe that if $T_i(x) = 1$ then $\hat{T}_i(x, r) = 0$ for every r. On the other hand, if $T_i(x) = 0$ then $\hat{T}_i(x, \mathcal{U}_d)$ is distributed uniformly over GF(2) (since $\hat{T}_i$ is an inner product of a random vector with a non-zero vector). Therefore, when $g(x) = 0$, the output of all the copies of each $\hat{T}_i$ are distributed uniformly and independently over GF(2). Hence, we can perfectly simulate $\hat{g}(x; r)$ in time $O(tk)$.

If $g(x) = 1$ then there exists a single term T_i that equals to one and the other terms equal to zero (since this is a canonic DNF); thus all the copies of $\hat{T}_i$ equal to zero while the other $\hat{T}_j$'s are distributed uniformly and independently over GF(2). The extraction algorithm will locate the first i for which all the copies of $\hat{T}_i$ equal to zero, and output the unique x which satisfies the term T_i. This algorithm errs only if there exists another i for which $T_i(x)$ is not satisfied but all the copies of $\hat{T}_i$ equal to zero. This event happens with probability at most $k2^{-t}$.

Similarly, the decoder outputs 1 if and only if there exists an i for which all the copies of $\hat{T}_i$ equal to zero. Therefore the decoder never errs when $g(x) = 1$ and errs with probability at most $k2^{-t}$ when $g(x) = 0$. □

Construction 4.3 yields an efficient degree-2 SPRE for functions whose canonic-DNF representation is efficiently computable. Specifically, we will be interested in functions $f : \{0,1\}^n \to \{0,1\}^l$ with logarithmic locality. In this case, we can efficiently encode each output with a degree-2 SPRE via the above construction.

4.2 Encoding a Function via SPRE

We move on to study the effect of applying an SPRE to a function. To this aim, we define a new operation on functions called *random exposure*.

Definition 4.5 (Random exposure of boolean function)**.** *Let* $f : \{0,1\}^n \to \{0,1\}$ *be a boolean function. The* random exposure *of* f *is the function* $f_{\exp}(x, b)$ *which maps* $x \in \{0,1\}^n$ *and* $b \in \{0,1\}$ *to the triple* $(f(x), b, z)$ *where*

$$z = \begin{cases} x & \text{if } b = 1, \\ 0^n & \text{if } b = 0. \end{cases}$$

Observe that SPRE exposes the input when the output evaluates to one, while $f_{\exp}$ tosses a random coin b that determines whether to reveal the input or not. Still, it is not hard to move from 1-exposure to random exposure by padding the output with a random bit and revealing this bit.

Formally, fix some efficiently computable boolean function $f : \{0,1\}^n \to \{0,1\}$. Let $g(x, y) = f(x) \oplus y$ and let $\hat{g}(x, y, r)$ be an SPRE of g. We will show that the function $h(x, y, r) = (\hat{g}(x, y, r), y)$ statistically encodes $f_{\exp}$ under the generalized notion of randomized encoding.

Lemma 4.6. *Assume that* $\hat{g}$ *is an SPRE of* g. *Then, the function* $h(x, y, r)$ *is a generalized statistical randomized encoding of the function* $f_{\exp}(x, b)$.

Proof. Let $h'(x,b,r) = h(x, f(x) \oplus b, r)$. It is not hard to see that h is isomorphic to h' via the mapping $(x,b,r) \mapsto (x, f(x) \oplus b, r)$. Furthermore, this mapping is efficiently computable and efficiently invertible since f is efficiently computable. It is left to show that h' is a (standard) randomized encoding of the function $f_{\exp}(x,b)$ with statistical correctness and perfect privacy.

We begin with correctness. Given $h'(x,b,r) = (\hat{g}(x,y,r), y)$ for $y = f(x) \oplus b$ and a uniformly chosen r, we apply the decoder of $\hat{g}$ to the first entry and recover, with all but negligible probability, the value $g(x,y) = f(x) \oplus y = b$. By XOR-ing this with y we recover $f(x)$. Knowing both $f(x)$ and b, it is left to recover x in case $b = 1$. Indeed, when $b = 1$ the function $g(x,y) = b$ also evaluates to 1, and so we can apply the extraction algorithm to $\hat{g}(x,y,r)$ and recover x with all but negligible probability, as required. In both cases the efficiency of the decoder follows from the efficiency of the original decoder and extraction algorithm.

We move on to privacy. Given an output $(f(x), b, z)$ of $f_{\exp}$, we compute $y = f(x) \oplus b$ and sample $\hat{g}(x,y,r)$ as follows: (1) if $g(x,y) = f(x) \oplus y = 1$ (i.e., $b = 1$) the input x is available (via z) and so we simply compute $\hat{g}(x,y,r)$ for a uniformly chosen r; (2) if $g(x,y) = f(x) \oplus y = 0$ (i.e., $b = 0$), we sample $\hat{g}(x,y,r)$ via the perfect one-sided simulator S_0 of $\hat{g}$. In any case, the simulation is perfect and efficient. □

We move on to the case of multi-output functions. In the following let $f : \{0,1\}^n \to \{0,1\}^l$ be a function where f_i denotes the boolean function computing the i-th bit of f and $K(i)$ is the set of inputs that influence the i-th output. For a string x we let $x_{K(i)}$ denote the restriction of x to the indices in $K(i)$. The random exposure $f_{\exp}$ of a non-boolean function f is defined by concatenating the random exposures of each of the outputs. Formally,

Definition 4.7 (Random exposure of general function)**.** *The* random exposure $f_{\exp}$ *of a function* $f : \{0,1\}^n \to \{0,1\}^l$ *maps* $x \in \{0,1\}^n$ *and* $b_1, \ldots, b_l \in \{0,1\}$ *to* $(f_{i,\exp}(x_{K(i)}, b_i))_{i \in [l]}$*, where* $f_{i,\exp}$ *is the random exposure of the boolean function* f_i*.*

By simple concatenation we extend Lemma 4.6 to the multi-output case.

Construction 4.8 (From SPRE to RE of $f_{\exp}$)**.** *For a function* $f : \{0,1\}^n \to \{0,1\}^l$ *define the following functions for every* $i \in [l]$*:*

$$\begin{aligned} g_i(x_{K(i)}, y_i) &\stackrel{def}{=} f_i(x_{K(i)}) \oplus y_i, \text{ where } f_i \text{ computes the } i\text{-th output of } f \\ h_i(x_{K(i)}, y_i, r_i) &\stackrel{def}{=} (\hat{g}_i(x_{K(i)}, y_i, r_i), y_i), \text{ where } \hat{g}_i \text{ is an SPRE of } g_i, \\ h(x, (y_i, r_i)_{i \in [l]}) &\stackrel{def}{=} (h_i(x_{K(i)}, y_i, r_i))_{i \in [l]}. \end{aligned}$$

Lemma 4.9. *The function* $h(x,y,r)$ *is a GSRE of the function* $f_{\exp}(x,b)$*.*

Proof. For every $i \in [l]$, let $h'_i(x_{K(i)}, b_i, r_i) = h_i(x_{K(i)}, f_i(x_{K(i)}) \oplus b_i, r_i)$. The proof of Lemma 4.6, shows that h'_i is a standard randomized encoding of $f_{i,\exp}(x_{K(i)}, b_i)$ with perfect privacy and statistical correctness. The concatenation lemma from [3, Lemma 4.9] shows that an encoding of a nonboolean function can be obtained by concatenating encodings of its output bits, using an independent random input for each bit. Hence, the function

$$h'(x, (b_i, r_i)_{i \in [l]}) = (h'_i(x_{K(i)}, b_i, r_i))_{i \in [l]},$$

is a standard statistical randomized encoding of $f_{\exp}(x,b)$. Finally, the isomorphism between h and h' is identical to the one described in the proof of Lemma 4.6. □

We can now derive the main result of this section.

Theorem 4.10. *Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be an efficiently computable function with output locality $O(\log n)$. Then its random exposure $f_{\exp}$ can be encoded via degree-2 generalized statistical randomized encoding h.*

Proof. Apply Construction 4.8 to f where each SPRE $\hat{g}_i$ is defined according to Construction 4.3. The theorem now follows from Lemmas 4.4 and 4.9. □

5 Robust One-Way Function

In this section we define the notion of robust one-way functions (ROWF), we use the results of the previous section to argue that a local ROWF gives rise to a degree-2 one-way function, and, finally we present a construction of local ROWFs based on the one-wayness of random local functions.

Definition 5.1 (Robust one-way function)**.** *Let $\mathcal{F} = \left\{f_k : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{k\in\{0,1\}^{s(n)}}$ be a collection of functions and let $\mathcal{G} = \{g_k : g_k$ is the random exposure of $f_k \in \mathcal{F}\}$ be the collection of random exposures of $\mathcal{F}$. We say that $\mathcal{F}$ is a* collection of robust OWFs *if $\mathcal{G}$ is a collection of weak one-way functions.*

Theorem 5.2. *If there exists a robust OWF f with locality $O(\log n)$ then there exists a degree-2 one-way function h with (optimal) output locality 3. Furthermore, if $\mathcal{F}$ is a collection of robust OWFs with locality $O(\log n)$, then there exists a collection $\mathcal{H}$ of degree-2 one-way functions with (optimal) output locality 3.*

Proof. Let f be the ROWF and let $f_{\exp}$ be its random exposure which is, by definition, a weak one-way function. Since f has logarithmic locality we can apply Theorem 4.10 and get a degree-2 generalized statistical randomized encoding h of $f_{\exp}$. By Lemma 3.7, h is a distributional one-way function. Finally, by Lemma 3.4, a degree-2 distributional OWF can be transformed into a degree-2 OWF with output locality 3.

We extend the proof for the case of collections along the argument of Remark 3.3. Let $\mathcal{F} = \{f_k\}$ be a collection of robust OWFs of locality $O(\log n)$ with key sampling algorithm $\mathcal{K}$. For every $f_k \in \mathcal{F}$, let g_k be the random exposure of f_k, and let h_k denote the degree-2 generalized statistical randomized encoding of g_k whose existence is promised by Theorem 4.10. We define the collections $\mathcal{G} = \{g_k\}$ and $\mathcal{H} = \{h_k\}$ which are both equipped with the same key-sampling algorithm $\mathcal{K}$. We will show that $\mathcal{H} = \{h_k\}$ is a collection of distributional one-way functions.

Consider the single-function representation of $\mathcal{F}$, denoted by F, which, given an input x and key-sampling randomness ρ, outputs value $f_{\mathcal{K}(1^n;\rho)}(x)$ together with the randomness ρ. Similarly, let G (resp., H) be the single-function representation of $\mathcal{G}$ and $\mathcal{H}$. By the definition of robustness, the collection $\mathcal{G}$ is weakly one-way and so, by Remark 3.3, the function G is weakly one-way. It is not hard to see that H is a GSRE of G since every h_k is a GSRE of g_k. It follows, by Lemma 3.7, that the function H is distributional one-way, which means, by definition, that $\mathcal{H}$ is a collection of distributional one-way functions. We complete the proof, by using Lemma 3.4 to transform $\mathcal{H}$ into a collection of degree-2 OWFs with output locality 3. □

5.1 Random Local Functions

Notation. Let m and n be integers corresponding to input length and output length. Let $M \in \{0,1\}^{m \times n}$ be a binary matrix with d_i ones in the i-th row M_i. Through this section we abuse notation and view M_i both as a row vector and as the set $\{j | M_{i,j} = 1\}$. Correspondingly, we let $|M_i|$ denote the size d_i of the set M_i. For a real number $p \in (0,1)$, denote by $\mathcal{M}_{m,n,p}$ the probability distribution over matrices $M \in \{0,1\}^{m \times n}$ where each entry is chosen to be one with probability p independently of the other entries.

The RLF collection. For a matrix M and a list of m predicates $P = (P_i)_{i=1}^m$ where $P_i : \{0,1\}^{|M_i|} \to \{0,1\}$, we define the function $f_{M,P} : \{0,1\}^n \to \{0,1\}^m$ via the mapping

$$x \mapsto (P_1(x_{M_1}), \ldots, P_m(x_{M_m})).$$

Namely, the i-th output is computed by applying P_i to the string x restricted to the set M_i. For integer-valued function $m(n) : \mathbb{N} \to \mathbb{N}$, real-valued function $q(n) : \mathbb{N} \to (0,1)$, and a sequence $\mathcal{P} = (\mathcal{P}^{(i)})_{i \in \mathbb{N}}$ of probability distributions $\mathcal{P}^{(i)}$ over i-ary predicates, we define the *Random Local Function* (RLF) collection, denoted by $\mathcal{F}_{n,m,q,\mathcal{P}}$, to be the collection of functions $f_{M,P}$ where M is chosen from $\mathcal{M}_{m,n,q}$ and the i-th predicate $P_i : \{0,1\}^{|M_i|} \to \{0,1\}$ is a random predicate chosen from $\mathcal{P}^{(|M_i|)}$.

Predicate distributions. We will consider two concrete choices for predicate distributions. We let $\mathcal{R}$ denote the *uniform distribution* over predicates, i.e., $\mathcal{R}^{(i)}$ uniformly samples a truth table from $\{0,1\}^{2^i}$. We also consider the distribution $\mathcal{R}_\oplus = \left\{\mathcal{R}_\oplus^{(i)}\right\}$ where $\mathcal{R}_\oplus^{(i)}$ chooses a predicate $P(z_1, \ldots, z_i)$ as follows. Randomly partition the input variables into two sets, S and its complement $\bar{S}$, where each input variable is inserted to S with probability $\frac{1}{2}$; Choose a uniform predicate P' over the S variables and XOR it with the parity of all variables outside S, i.e., $P(z) = P'(z_S) \oplus \bigoplus_{i \notin S}(z_i)$. We refer to $\mathcal{R}_\oplus$ as the XOR-uniform distribution.

More generally, our results apply to any predicate distribution $\mathcal{P} = (\mathcal{P}^{(i)})_{i \in \mathbb{N}}$ which satisfies the following conditions.

Definition 5.3. *A predicate distribution $\mathcal{P} = (\mathcal{P}^{(i)})_{i \in \mathbb{N}}$ is* admissible *if the following hold:*

- *For every i, the distribution $\mathcal{P}^{(i)}$ is invariant under permutation of the order of the input variables. Namely, for any permutation $\pi : [i] \to [i]$, the predicate $Q : \{0,1\}^i \to \{0,1\}$ obtained by sampling $P \xleftarrow{R} \mathcal{P}^{(i)}$ and letting $Q(z_1, \ldots, z_i) = P(z_{\pi(1)}, \ldots, z_{\pi(i)})$, is distributed identically to $\mathcal{P}^{(i)}$.*
- *$\mathcal{P}$ is closed under partial fixing. Namely, the following holds for every i. For every $s < i$, every s-size subset $S \subset [i]$ and partial assignment $\rho \in \{0,1\}^S$, the predicate $Q : \{0,1\}^{i-s} \to \{0,1\}$ obtained by sampling $P \xleftarrow{R} \mathcal{P}^{(i)}$ and fixing the variables in S to the assignment ρ, is distributed identically to $\mathcal{P}^{(i-s)}$.*
- *Efficiency: One can sample the truth table of a predicate $P \xleftarrow{R} \mathcal{P}^{(i)})$ in time* $\mathrm{poly}(2^i)$.

It is not hard to verify that the aforementioned predicate distributions, $\mathcal{R}$ and $\mathcal{R}_\oplus$ (uniform and XOR-uniform) are admissible.

Intractability assumption. We assume that the collection $\mathcal{F}_{n,m,d/n,\mathcal{P}}$ is weakly hard to invert for some constant d, output length $m = n$ and some admissible predicate distribution $\mathcal{P}$. Observe that in this case the expected output locality is d, and, by Fact 3.1 item 2, with all but negligible probability, all outputs have locality at most $\log n$ and so P_i can be described by a polynomial-size string and the function can be evaluated in polynomial time.

Assumption 5.4 (Random local function)**.** *There exists a constant d, an admissible predicate distribution $\mathcal{P} = \{\mathcal{P}^{(i)}\}$ and a polynomial $p(\cdot)$, such that for every (non-uniform) polynomial-time algorithm, A, and all sufficiently large n's*

$$\Pr[A(1^n, M, P, f_{M,P}(x)) \notin f_{M,P}^{-1}(f_{M,P}(x))] > \frac{1}{p(n)},$$

where $x \xleftarrow{R} \mathcal{U}_n, M \xleftarrow{R} \mathcal{M}_{n,n,d/n}$ and $P = (P_i)_{i=1}^n$ and each $P_i : \{0,1\}^{|M_i|} \to \{0,1\}$ is sampled independently from $\mathcal{P}^{(|M_i|)}$.

About the assumption. The collection $\mathcal{F}_{n,n,d/n,\mathcal{P}}$ used in Assumption 5.4 is a variant of a candidate OWF proposed by Goldreich [15] that was extensively studied in the last decade. The main difference between our variant and Goldreich's variant is that we use a different random predicate for each output and Goldreich suggests to use the same (possibly random) predicate everywhere. Despite this difference, known evidence that support Goldreich's variant also apply to our variant as well.

In particular, Cook et al. [8] prove that a large class of algorithms (including ones that capture DPLL-based heuristics) fail to invert a local functions $f_{M,P} : \{0,1\}^n \to \{0,1\}^n$ in polynomial time as long as the matrix M enjoys some expansion properties and most of the predicates $P_1, \ldots, P_n$ satisfy some notion of hardness (captured by a concrete combinatorial criteria).[4] Furthermore, it is shown that predicates chosen from either $\mathcal{R}$ or $\mathcal{R}_\oplus$ fail to be "hard" with probability that tends to 0 when the arity d grows. Similarly, a random matrix $M \xleftarrow{R} \mathcal{M}_{n,n,d/n}$ fails to satisfy the required expansion property with probability that tends to 0 when the arity d grows, and so the lower-bounds of [8] apply to our setting as well.

We further mention that things change in the long-output regime. Specifically, the results of [6] allow to invert $f_{M,P}$, for most matrices M, as long as there are enough $(\text{poly}(d)n)$ "easy" predicates $P_i(z_1, \ldots, z_d)$ which are correlated with a single input variable z_j (i.e., $\Pr_z[P_i(z) = z_j] \neq \frac{1}{2}$) or with a pair of input variables $z_j \oplus z_k$ (i.e., $\Pr_z[P_i(z) = z_j \oplus z_k] \neq \frac{1}{2}$). Since a uniform predicate $P \xleftarrow{R} \mathcal{R}$ is likely to be easy, the collection $\mathcal{F}_{n,m,d/n,\mathcal{R}}$ is efficiently invertible for $m \geq \text{poly}(d)n$. In contrast, a predicate P chosen from the XOR-uniform distribution $\mathcal{R}_\oplus$ is easy with only small probability of $\exp(-\Omega(d))$, and so this attack fails for $\mathcal{F}_{n,cn,d/n,\mathcal{R}_\oplus}$ as long as $c \ll \exp(d)n$.

Overall, for $m = n$, it seems likely that Assumption 5.4 holds both for uniform predicates $\mathcal{R}$ and the XOR-uniform predicates $\mathcal{R}_\oplus$. In fact, it seems likely that both $\mathcal{F}_{n,n,d/n,\mathcal{R}}$ and $\mathcal{F}_{n,n,d/n,\mathcal{R}_\oplus}$ are *strongly one-way.*

[4] The results are originally stated for a fixed good predicate but it can be easily generalized to a mixture of predicates and to the case that all but a small fraction of the predicates are "hard".

5.2 Robust One-Way Functions from Random Local Functions

Based on the one-wayness of $\mathcal{F}_{n,n,d/n}$ we show that the related collection $\mathcal{F}_{n',m',d'/n'}$ is robust one-way.

Theorem 5.5. *Let $\mathcal{P}$ be some admissible predicate distribution and let $d \in \mathbb{N}$ be a constant. If the collection $\mathcal{F}_{n,n,d/n,\mathcal{P}}$ is weakly one-way then $\mathcal{F}_{n',m',q',\mathcal{P}}$ is robust one-way for $n' = \lceil e^d n \rceil$, $m' = 2n$ and $q' = d/(\lfloor n'/e^d \rfloor)$.*

The proof follows the outline sketched in the introduction and is deferred to Section 5.3.

Remark 5.6. *Recall that the notion of robust one-way collection only requires showing that the random exposure of $\mathcal{F}_{n',m',q',\mathcal{P}}$ is weakly one-way. Still, one could hope to prove that, if $\mathcal{F}_{n,n,d/n,\mathcal{P}}$ is one-way (as opposed to weakly one-way), then so is the random exposure of $\mathcal{F}_{n',m',q',\mathcal{P}}$. Our proof of Theorem 5.5 falls short of providing such a reduction.*

We conclude with the following corollary.

Corollary 5.7. *Under Assumption 5.4, there exists a collection of degree-2 OWF with locality 3.*

Proof. Under Assumption 5.4, Theorem 5.5 implies that the collection $\mathcal{F} := \mathcal{F}_{n',m',q',\mathcal{P}}$ is robust one-way. Let us modify this collection by rejecting all functions whose locality is larger than $\log n$. As already observed, by Fact 3.1 item 2, this affects only a negligible fraction of the functions, and so the resulting collection is still robust one-way. Our new collection $\mathcal{F}'$ now has logarithmic locality, and the description of each function $h_{M,P}$ in this collection includes an explicit description of the truth tables $(P_1, \ldots, P_{m'})$. Thus, we can apply Theorem 5.2 and construct a collection of degree-2 OWFs with locality 3 based on Assumption 5.4. □

5.3 Proof of Theorem 5.5

Let $d \in \mathbb{N}$ be a constant. Let $\mathcal{P}$ be an admissible predicate distribution, which will be omitted, from now on, from the notation $\mathcal{F}$, i.e., $\mathcal{F}_{n,m,d/n} := \mathcal{F}_{n,m,d/n,\mathcal{P}}$. We will show that if $\mathcal{F}_{n,m,d/n}$ is weakly one-way, for $m = n$, then $\mathcal{F}_{n',m',q'}$ is robust one-way, for $n' = \lceil e^d n \rceil$, $m' = 2m$ and $q' = d/(\lfloor n'/e^d \rfloor) = d/n$. The overall strategy is to embed an output of a function $f_{M,P} \xleftarrow{R} \mathcal{F}_{n,m,d/n}$ in an output of a random exposure h_{exp} of $h_{M',P'} \xleftarrow{R} \mathcal{F}_{n',m',q'}$ and then show that an inverter $\hat{A}$ for h_{exp} can be used to invert $f_{M,P}$.

Formally, let $p(n)$ be the polynomial guaranteed by the assumption that $\mathcal{F}_{n,m,d/n}$ is weakly one-way, and assume, towards a contradiction, that the random exposure of $h \xleftarrow{R} \mathcal{F}_{n',m',q'}$ is not weakly one-way. Specifically, let $\varepsilon(n')$ be a function which is smaller than $\frac{1}{n^2 p(n)}$ for infinitely many n's. Assume that there exists an efficient algorithm $\hat{A}$ that with probability $1 - \varepsilon(n')$ inverts a random output

$$\phi = (M_i, P_i, y_i, b_i, z_i)_{i=1}^{m'}$$

chosen from the *uniform distribution* $\Phi_{n'}$ defined by letting:

$$M \xleftarrow{R} \mathcal{M}_{n',m',q'}, P_i \xleftarrow{R} \mathcal{P}^{(|M_i|)}, x \xleftarrow{R} \{0,1\}^{n'}, b = (b_1, \ldots, b_{m'}) \xleftarrow{R} \{0,1\}^{m'}, y_i = P_i(x_{M_i})$$

and

$$z_i = \begin{cases} x_{M_i} & \text{if } b_i = 1 \\ 0^{|M_i|} & \text{otherwise.} \end{cases}$$

We would like to analyze the success probability of $\hat{A}$ on a restricted class of "typical" instances. Given a tuple ϕ, we say that an output i is *exposed* if $b_i = 1$ and say that an input j is *exposed* if it participates in an exposed output, i.e., if $M_{i,j} = 1$ for some exposed output i. An input or output which is not exposed is simply called *unexposed*. We say that ϕ is *typical* if the number of unexposed outputs is exactly $m'/2 = m$, the number of unexposed inputs is exactly n (out of n'), and all the outputs are connected to at most $\log n$ inputs. We say that a typical tuple ϕ is in *canonical* form if the first m outputs and the first n inputs are exactly the ones which are unexposed. Let $T_{n'}$ (respectively, $C_{n'}$) denote the uniform distribution $\Phi_{n'}$ conditioned on selecting a typical (respectively, canonical) instance.

It turns out that the algorithm $\hat{A}$, which is guaranteed to invert random instances $\phi \xleftarrow{R} \Phi_{n'}$ with probability $1 - \varepsilon$, inverts typical instances with probability $1 - \Omega(n'\varepsilon)$. Furthermore, $\hat{A}$ can be easily modified into an inverter for canonical instances with similar success probability.

Lemma 5.8. *The inverter $\hat{A}$ inverts a typical instance $\phi \xleftarrow{R} T_{n'}$ with probability at least $1-\Omega(n'\varepsilon)$. Furthermore, there exists an efficient algorithm B that inverts a random canonical instance $\phi \xleftarrow{R} C_{n'}$ with probability at least $1 - \Omega(n'\varepsilon)$.*

The first part is proven by showing that a random instance is typical with probability $\Omega(1/n')$, and the second part is proven by noting that a canonical instance can be randomized into a typical instance by randomly permuting the inputs and the outputs. See Section 5.3.1 for a full proof.

The next lemma, whose proof is deferred to Section 5.3.2, shows that a random output ψ of $\mathcal{F}_{n,m,d/n}$ can be embedded in a random canonical output of $\mathcal{F}_{n',m',q'}$.

Lemma 5.9. *There is an efficient transformation α which takes a random output ψ of $\mathcal{F}_{n,m,d/n}$, aborts with failure with negligible probability, and otherwise (conditioned on not aborting) outputs a random canonical output $\phi \xleftarrow{R} C_{n'}$ such that if $x = (x_1, \ldots, x_{n'})$ is a preimage of ϕ then the prefix $(x_1, \ldots, x_n)$ is a preimage of ψ.*

We can now complete the proof of Theorem 5.5. By Lemma 5.8, the existence of the inverter $\hat{A}$ which inverts $\Phi_{n'}$ with probability $1-\varepsilon$ implies the existence of an efficient algorithm B that inverts random canonical instances $\phi \xleftarrow{R} C_{n'}$ with probability $1 - \Omega(1/(np(n)))$. Hence, by Lemma 5.9, B can be used to invert $\mathcal{F}_{n,m,d/n}$ with similar probability as follows. Map an output of $\mathcal{F}_{n,m,d/n}$ to an output $\phi \xleftarrow{R} C_{n'}$ via α, use B to find a preimage x' and output the first n bits of x'. The success probability is $1 - \Omega(1/(np(n)))$, in contradiction to the assumption of the theorem.

5.3.1 Proof of Lemma 5.8

We will need the following claim:

Claim 5.10. *For some constant $a > 0$, $\Pr_{\phi \xleftarrow{R} \Phi_{n'}}[\phi \text{ is typical }] > \frac{a}{n'}$.*

Proof. Since each of the $m' = 2n$ output vertices is exposed with probability $\frac{1}{2}$, by Fact 3.1 item 1, the probability that exactly n outputs are exposed is $\binom{2n}{n} \cdot 2^{-2n} = \Omega(1/\sqrt{n})$. Conditioned on this event, we claim that the probability that there are exactly n unexposed inputs out of all $n' = \lceil e^d n \rceil$ inputs, is $\Omega(1/\sqrt{n}) = \Omega(1/\sqrt{n'})$.

Indeed, since each edge exists with probability $q' = \frac{d}{n}$, an input is unexposed with probability $(1-q')^n = (1-d/n)^n$. Therefore, the number of unexposed inputs is distributed according to the binomial distribution, and it suffices to show that $b(n; n', (1-d/n)^n) > \Omega(1/\sqrt{n})$, where $b(k; \ell, p)$ is the probability to have exactly k successes out of a sequence of ℓ independent Bernoulli trials each with a probability p of success. Indeed, letting $p = (1-d/n)^n$ and recalling that $n' = \lceil e^d n \rceil$, we can write:

$$b(n; n', p) = b(\lfloor p(n'+1) \rfloor + O(1); n', p) > \Omega(b(\lfloor p(n'+1) \rfloor; n', p)) > \Omega(1/\sqrt{n'}),$$

where the first equality follows from the estimate $(1-d/n)^n = e^{-d} - O(1/n)$ (derived from the Taylor expansion), the second inequality follows from the smoothness of $b(k; \ell, p)$ with respect to k, namely, the ratio $b(k; \ell, p)/b(k+1; \ell, p) = (1-p)k/(\ell - k)p$ is constant when $k = \Theta(\ell)$ and $p \in (0,1)$ is a constant, and the third inequality follows from Fact 3.1 item 1. Finally, by Fact 3.1 item 2, the probability that some output is connected to more than $\log n$ inputs is negligible $(m' e^{-\Omega(\log n' \log\log n')})$, and the claim follows. □

We can now prove Lemma 5.8. The first part follows from Bayes' law as $\Pr[\hat{A} \text{ does not invert } T_{n'}]$ equals to

$$\frac{\Pr[\hat{A} \text{ does not invert } \Phi_{n'} \bigwedge \Phi_{n'} \text{ is typical}]}{\Pr[\Phi_{n'} \text{ is typical}]} \leq \frac{\Pr[\hat{A} \text{ does not invert } \Phi_{n'}]}{\Pr[\Phi_{n'} \text{ is typical}]} < \frac{\varepsilon}{a/n'},$$

where the last inequality follows from Claim 5.10. Overall, we get:

$$\Pr[\hat{A} \text{ inverts } T_{n'}] > 1 - n'\varepsilon/a,$$

where a is a positive constant.

To prove the second item we note that one can easily map $T_{n'}$ to $C_{n'}$ by randomly permuting the order of the inputs and the order of the outputs. Formally, given a canonical instance $\phi = (M_i, P_i, y_i, b_i, z_i)_{i=1}^{m'}$ we define the inverter B as follows:

- Choose a random output permutation $\sigma : [m'] \to [m']$ and let $\phi' = (M'_i, P'_i, y'_i, b'_i, z'_i)_{i=1}^{m'}$ where

$$(M'_{\sigma(i)}, P'_{\sigma(i)}, y'_{\sigma(i)}, b'_{\sigma(i)}, z'_{\sigma(i)}) = (M_i, P_i, y_i, b_i, z_i).$$

- Choose a random input permutation $\pi : [n'] \to [n']$ and define

$$\phi'' = (M''_i, P''_i, y'_i, b'_i, z''_i)_{i=1}^{m'}$$

  as follows. (1) $M''_i = \{\pi(j) | j \in M_i\}$. (2) Let $\pi_i : [|M_i|] \to [|M_i|]$ be the permutation that maps j to k if the j-th largest element in M'_i is mapped by π to be the k-th largest element in M''_i. Then, the predicate P''_i is defined by permuting the input variables of P'_i under π_i. (3) The string z''_i is defined by permuting the entries of z'_i under π_i.

- Finally, invoke $\hat{A}$ on ϕ'', copy the output to x'', permute the entries of x'' under π^{-1}, and output the resulting string x.

It is not hard to see that a random canonical tuple ϕ is mapped by B into a random typical tuple ϕ''. (Here we make use of the fact that the predicate distribution $\mathcal{P}$ is invariant under permutations of the input variables.) Furthermore, if x'' is a preimage of ϕ'', then x is a preimage of ϕ. Hence, B succeeds with probability $1 - \Omega(n'\varepsilon)$.

5.3.2 Proof of Lemma 5.9

In the following we let $\mathcal{L}_{a,b,p}$ denote the distribution $\mathcal{M}_{a,b,p}$ conditioned on the event that none of the columns is an all-zero column, and none of the rows have Hamming weight larger than $\log n$. For our setting of parameters (i.e., $a = \Theta(n), b = \Theta(n)$ and $p = \Theta(1/n)$), such a distribution can be sampled efficiently with negligible failure probability.[5]

Given a tuple $\psi = (M_i, P_i, y_i)_{i=1}^{m}$ (which corresponds to an image of $\mathcal{F}_{n,m,d/n}$) we compute $\phi = (M'_i, P'_i, y'_i, b_i, z_i)_{i=1}^{m'}$ (from $C_{n'}$) as follows:

1. **Padding the matrix.** Sample a $m' \times n'$ matrix

$$M' \xleftarrow{R} \begin{pmatrix} M & \mathcal{M}_{m,n'-n,d/n} \\ \mathbf{0}_{(m'-m)\times n} & \mathcal{L}_{m'-m,n'-n,d/n}, \end{pmatrix}.$$

   Halt with failure if one of the rows of M' has Hamming weight larger than $\log n$.

2. **Exposed inputs.** Choose $n' - n$ random bits $x_{n+1} \dots x_{n'}$.

3. **Unexposed outputs.** For $i \in \{1, \dots, m\}$ let

$$y'_i = y_i, \qquad b_i = 0, \qquad z_i = 0^{|M'_i|}.$$

   Choose $P'_i : \{0,1\}^{|M'_i|} \to \{0,1\}$ at random from $\mathcal{P}^{(|M'_i|)}$ subject to the constraint

$$P'_i(\beta, x_{M'_i \setminus M_i}) = P_i(\beta) \text{ for every } \beta \in \{0,1\}^{|M_i|}.$$

   Namely, the restricted predicate $P'_i(\cdot, x_{M'_i \setminus M_i})$ is equal to P_i.

4. **Exposed outputs.** For each $i \in \{m+1, \dots, m'\}$ let

$$P'_i \xleftarrow{R} \mathcal{P}^{(|M'_i|)}, \qquad y'_i = P'_i(x_{M'_i}), \qquad b_i = 1, \qquad z_i = x_{M'_i}.$$

   Since M'_i contains only "exposed inputs" $n+1 \le j \le n'$ we can compute y'_i and z_i.)

[5] Sample a matrix from $\mathcal{M}_{a,b,p}$ conditioned on the event that none of the columns is an all-zero column (a task which can be achieved efficiently by rejecting and resampling zero columns), and fail it if the resulting matrix has a row of weight larger than $\log n$. To see that rejection happens with negligible probability, observe that for a fixed row, each column contributes 1 with probability $p/(1-p)^a = \Theta(1/n)$, independently of the other columns, and therefore, by Fact 3.1 item 2, the weight exceeds $\log n$ with negligible probability.

Analysis. Assume that ψ is uniformly distributed, namely $M \xleftarrow{R} \mathcal{M}_{n,n,d/n}$, $x \xleftarrow{R} \{0,1\}^n$, and for $i \in [n]$, $P_i \xleftarrow{R} \mathcal{P}^{(|M'_i|)}$ and $y_i = P_i(x)$. Observe that the failure probability in the first step is negligible. Indeed, by Fact 3.1 item 2, with all but negligible probability none of the first m rows of M' has Hamming weight larger than $\log n$. (The last $m' - m$ rows of M' have Hamming weight smaller than $\log n$ due to the definition of $\mathcal{L}$.) We further claim that, conditioned on not failing, ϕ is distributed according to $C_{n'}$. Indeed, the following hold:

- The string $b = 1^m 0^{m'-m}$, the first m rows of M' are distributed according to $\mathcal{M}_{m,n',q'}$, and the last rows according to $\mathbf{0}_{m'-m\times n}\mathcal{L}_{m'-m,n'-n,q'}$. (Recall that $q' = d/n$.)
- Since the original predicates $(P_1, \ldots, P_m)$ are chosen from $\mathcal{P}$, so are the new predicates $(P'_1, \ldots, P_{m'})$. (Here we use the fact that $\mathcal{P}$ is invariant under partial fixing.)
- Since $y_i = P_i(x_{M_i})$ for $x \xleftarrow{R} \{0,1\}^n$, we have, by construction, that $y'_i = P'_i(x_{M'_i})$ where $x \xleftarrow{R} \{0,1\}^{n'}$.
- Finally, again, by construction, for $m+1 \le i \le m'$ we have that $z_i = x_{M'_i}$ and $z_i = 0^{|M'_i|}$ otherwise.

Now suppose that $x' = (x'_1, \ldots, x'_{n'})$ is a preimage of ψ. We show that, in this case, the prefix $(x'_1, \ldots, x'_n)$ is a preimage of ϕ. Since all the inputs $n+1 \le i \le n'$ are exposed, we have that $x'_{n+1,\ldots,n'} = x_{n+1,\ldots,n'}$. In addition, for $1 \le i \le m$ we have

$$y_i = P'_i(x'_{M_i}, x'_{R_i}) = P'_i(x'_{M_i}, x_{R_i}) = P_i(x'_{M_i}),$$

where R_i denotes the restriction of M'_i to the last $(n' - n)$ columns, and the last equality follows from the definition of P'_i. Therefore, $f_{M,P}(x'_1, \ldots, x'_n) = y$, as claimed. This completes the proof of Lemma 5.9.

6 Discussion and Open Questions

Leakage Resiliency of Random Local Functions. Our results essentially show that, for short output lengths $m = 2n/e^d$, the RLF collection achieves some form of *leakage resiliency* under *random exposures*. One may argue that this captures a realistic model of "probing attacks" where an adversary gets a physical access to the circuit that computes f, and for each output gate i, with probability $\frac{1}{2}$, gets to probe all the internal values of the sub-circuit that computes the output. We prove that one-wayness can still be achieved under this strong notion of leakage. Related models of physical leakage were considered in [23, 1, 12]. We also mention that Random Local Functions were used by [22] to obtain a pseudorandom generator which achieves a closely-related notion of leakage resiliency.

Interestingly, random local functions are known to be insecure (easy to invert) in a different model of *random leakage* where the adversary is allowed to see a constant fraction of the inputs which are selected *independently* of the function $f \xleftarrow{R} \mathcal{F}$. In fact, Bogdanov and Qiao [6] showed that, for long output length $m = n(1 + \mathrm{poly}(d))$, a random d-local function can be efficiently inverted even in the (more challenging) setting where the adversary is only given a string x' which is correlated with x without knowing which indices are correlated. Furthermore, for some predicates,

Dinur et al. [10] extended this attack to a setting where the adversary only gets to see the image y' of x'. In all these cases, the leakage (the choice of x') is crucially assumed to be statistically independent of the choice of f. In contrast, in our random-exposure model, the leakage strongly depends on the structure of the circuit that computes f. Hence, there seems to be a significant difference between circuit-dependent leakage and random leakage. It will be interesting to further study the relations between these different models of leakage.

Standard RE with Degree 2. In Section 4 it is shown that non-trivial functions can be encoded by degree-2 *semi-private* randomized encodings. This leaves open the possibility of achieving standard degree-2 randomized encoding with statistical privacy and statistical correctness. A positive result would lead to round-optimal secure computation protocols in the statistical setting of [5].

Locality 2 over larger alphabet. It is not hard to see that the task of inverting a 2-local binary function reduces to solving an instance of 2-SAT and can therefore be implemented in polynomial time [15]. However, this attack does not generalize to 2-local functions over a larger *non-binary* alphabet. Indeed, it is shown in [9] that, assuming the existence of one-way functions in log-space, there are one-way functions with locality 2 over a (large) constant-size alphabet.

Acknowledgement. We thank Irit Dinur and Hilary Finucane for sharing with us a copy of [9]. We also thank the anonymous reviewers for their useful feedbacks and for correcting an error in the proof of Lemma 5.8.

References

[1] M. Ajtai. Secure computation with information leaking to an adversary. In *Proc. of 43rd STOC*, pages 715–724, 2011.

[2] B. Applebaum. The cryptographic hardness of random local functions. *Computational Complexity*, 2015. DOI:10.1007/s00037-015-0121-8. Also available as ECCC TR15-027.

[3] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006. Preliminary version in Proc. 45th FOCS, 2004.

[4] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography by cellular automata or how fast can complexity emerge in nature? In *Proc. of 1st ICS*, pages 1–19, 2010.

[5] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. of 20th STOC*, pages 1–10, 1988.

[6] A. Bogdanov and Y. Qiao. On the security of Goldreich's one-way function. *Computational Complexity*, 21(1):83–127, 2012. Preliminary version in Proc. of 13th RANDOM, 2009.

[7] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proc. of 4th TCC*, pages 535–554, 2007.

[8] J. Cook, O. Etesami, R. Miller, and L. Trevisan. On the one-way function candidate proposed by goldreich. *TOCT*, 6(3):14, 2014. Preminary version in Proc. of 6th TCC.

[9] I. Dinur and H. Finucane. Cryptography with locality two. Unpublished Manuscript, 2011.

[10] I. Dinur, S. Goldwasser, and H. Lin. The computational benefit of correlated instances. In *Proc. of 6th ITCS*, pages 219–228, 2015.

[11] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[12] S. Faust, T. Rabin, L. Reyzin, E. Tromer, and V. Vaikuntanathan. Protecting circuits from computationally bounded and noisy leakage. *SIAM J. Comput.*, 43(5):1564–1614, 2014.

[13] T. K. Frederiksen, J. B. Nielsen, and C. Orlandi. Privacy-Free Garbled Circuits with Applications to Efficient Zero-Knowledge. In *Proc. of 34th EUROCRYPT*, 191-219, 2015.

[14] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.*, 60(3):592–629, 2000.

[15] O. Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(090), 2000.

[16] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of 13th CCS*, pages 89–98, 2006.

[18] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate Encryption for Circuits from LWE. In *Proc. of 35th CRYPTO*, pages 503–523, 2015.

[19] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proc. of the 30th FOCS*, pages 230–235, 1989.

[20] Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Proc. 41st FOCS*, pages 294–304, 2000.

[21] Y. Ishai and E. Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Proc. 29th ICALP*, pages 244–256, 2002.

[22] Y. Ishai, E. Kushilevitz, X. Li, R. Ostrovsky, M. Prabhakaran, A. Sahai, and D. Zuckerman. Robust pseudorandom generators. In *Proc. of 40th ICALP*, pages 576–588, 2013.

[23] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In *Proc. of 23rd CRYPTO*, pages 463–481, 2003.

[24] Y. Ishai, and H. Wee. Partial Garbling Schemes and Their Applications. In *Proc. of 41st ICALP*, pages 650–662, 2014.

[25] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Proc. of 27th EUROCRYPT*, pages 146–162, 2008.

[26] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proc. of 24th EUROCRYPT*, pages 457–473, 2005.

[27] E. Shi, J. Bethencourt, H. T. Chan, D. X. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pages 350–364, 2007.

[28] A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd FOCS*, pages 80–91, 1982.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de