(Almost) Optimal Constructions of UOWHFs from 1-to-1, Regular One-way Functions and Beyond

Yu Yu* Dawu Gu† Xiangxue Li‡ Jian Weng§

Abstract

We revisit the problem of black-box constructions of universal one-way hash functions (UOWHFs) from several (from specific to more general) classes of one-way functions (OWFs), and give respective constructions that either improve or generalize the best previously known. In addition, the parameters we achieve are either optimal or almost optimal simultaneously up to small factors, e.g., arbitrarily small $\omega(1)$.

- For any 1-to-1 one-way function, we give an optimal construction of UOWHFs with key and output length $\Theta(n)$ by making a single call to the underlying OWF. This improves the constructions of Naor and Yung (STOC 1989) and De Santis and Yung (Eurocrypt 1990) that need key length $O(n \cdot \omega(\log n))$.
- For any known-(almost-)regular one-way function with known hardness, we give an optimal construction of UOWHFs with key and output length $\Theta(n)$ and a single call to the one-way function.
- For any known-(almost-)regular one-way function, we give a construction of UOWHFs with key and output length $O(n{\cdot}\omega(1))$ and by making $\omega(1)$ non-adaptive calls to the one-way function. This improves the construction of Barhum and Maurer (Latincrypt 2012) that requires key and output length $O(n{\cdot}\omega(\log n))$ and $\omega(\log n)$ calls.
- For any weakly-regular one-way function introduced by Yu et al. at TCC 2015 (i.e., the set of inputs with maximal number of siblings is of an n^{-c}-fraction for some constant c), we give a construction of UOWHFs with key length $O(n{\cdot}\log n)$ and output length $\Theta(n)$. This generalizes the construction of Ames et al. (Asiacrypt 2012) which requires an unknown-regular one-way function (i.e., $c = 0$).

Along the way, we use several techniques that might be of independent interest. We show that almost 1-to-1 (except for a negligible fraction) one-way functions and known (almost-)regular one-way functions are equivalent in the known-hardness (or non-uniform) setting, by giving an optimal construction of the former from the latter. In addition, we show how to transform any one-way function that is far from regular (but only weakly regular on a noticeable fraction of domain) into an almost-regular one-way function.

Keywords: Foundations, One-way Functions, Universal One-way Hash Functions, Target Collision Resistance.

*Shanghai Jiao Tong University. Email: yuyuathk@gmail.com.
†Shanghai Jiaotong University.
‡East China Normal University.
§Jinan University.

1 Introduction

Informally, a family of compressing hash functions, denoted by $\mathcal{G}$, is called *universal one-way*, if given a random function $g \in \mathcal{G}$ and a random (or equivalently, any pre-fixed) input x, it is infeasible for any efficient algorithm to find any $x' \neq x$ satisfying $g(x) = g(x')$. The seminal result that one-way functions (OWFs) imply universal one-way hash functions (UOWHFs) [19] constitutes one of the central pieces of modern cryptography. Applications of UOWHFs include basing digital signatures [9] on minimal assumptions (one-way functions), Cramer-Shoup encryption scheme [4], statistically hiding commitment scheme [12, 13], etc.

UOWHFs from any OWFs. The principle possibility result that UOWHFs can be based on any OWF was established by Rompel [19] (with some corrections given in [20, 16]). However, Rompel's construction was quite complicated and extremely unpractical. In particular, for any one-way function on n-bit inputs it requires key length $\tilde{O}(n^{12})$ and output length $\tilde{O}(n^8)$. Haitner et al. [11] improved the construction via the notion of inaccessible entropy [13], and reduced key and output length to $\tilde{O}(n^7)$. Therefore, even the best known generic UOWHF constructions (based on arbitrary OWFs) are mainly of theoretical interest and are too inefficient to be of any practical use.

UOWHFs from special OWFs. Another line of research focuses on more efficient (and nearly practical) constructions of UOWHFs from special structured OWFs. Naor and Yung gave an elegant "hash-then-truncate" construction of UOWHFs with key and output length $\Theta(n)$ which does a single call to any one-way permutation. More specifically, let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation, let h be a random permutation (over n bits) from a pairwise-independent hash permutation family $\mathcal{H}$, and let $\mathsf{trunc} : \{0,1\}^n \to \{0,1\}^{n-1}$ be a truncating function that outputs the first $n-1$ bits of input, then the following

$$\mathcal{G}_{owp} \stackrel{\mathsf{def}}{=} \{\ (\mathsf{trunc}\ \circ\ h\ \circ\ f) : \{0,1\}^n \to \{0,1\}^{n-1}\ ,\ h \in \mathcal{H}\ \}$$

is a family of UOWHFs with 1 bit of shrinkage (i.e., compress by 1 bit), where "$\circ$" denotes function composition. However, for a slightly weaker primitive, namely, 1-to-1 one-way functions, the authors of [17] only gave a rather complicated construction. De Santis and Yung [21] gave an improved construction from any 1-to-1 OWF $f : \{0,1\}^n \to \{0,1\}^l$ as below:

$$\mathcal{G}_{\text{1-1}} \stackrel{\mathsf{def}}{=} \{\ (h_{n-1}^n\ \circ\ \ldots\ \circ\ h_{l-2}^{l-1}\ \circ\ h_{l-1}^l\ \circ\ f) : \{0,1\}^n \to \{0,1\}^{n-1}\ ,\ h_{n-1}^n \in \mathcal{H}_{n-1}^n, \ldots, h_{l-1}^l \in \mathcal{H}_{l-1}^l\ \}\ ,$$

where each $\mathcal{H}_{i-1}^i$ denotes a family of pairwise-independent hash functions that compress i-bit strings into $(i-1)$ bits. Although $\mathcal{G}_{\text{1-1}}$ enjoys linear output length and a single function call, it requires[1] key length $O(\omega(\log n){\cdot}n)$. In addition, the work of [21] also introduced a construction from any known-regular[2] one-way function with key and output length $O(\omega(\log^2 n)\cdot n)$ and $O(\omega(1)\cdot\log n)$ adaptive calls, which was recently improved by Barhum and Maurer [3] to key and output length $O(\omega(\log n)\cdot n)$ and $O(\omega(1)\cdot\log n)$ non-adaptive calls. Based on unknown-regular one-way functions, Ames et al. [1] presented a more general construction with output length $\Theta(n)$, key length $O(\log n\cdot n)$ and $\tilde{O}(n)$ adaptive calls. We refer to Table 1 for a summary of previous constructions and a comparison to our work.

[1] A straightforward calculation suggests that $\mathcal{G}_{\text{1-1}}$ needs key length $O(l{\cdot}(l-n))$, and we know (see Fact 1) that every 1-to-1 one-way function implies another one-way function $f' : \{0,1\}^{n' \in \Theta(n)} \to \{0,1\}^{n'+\omega(\log n)}$ that is 1-to-1 except on a negligible fraction of inputs, which implies that the key length of [17, 21] can be pushed to $O(\omega(\log n){\cdot}n)$.

[2] A function f is regular if every image has the same number (say α) of preimages, and it is known- (resp., unknown-) regular if α is efficiently computable (resp., inefficient to approximate). More generally (as introduced in [24]), f is weakly unknown-regular if the fraction of x's with maximal $|f^{-1}(f(x))|$ (which is not necessarily efficiently computable) is noticeable. We stress that here "weakly" is used to describe "regularity" (rather than "one-way-ness" as in "weakly one-way functions").

Table 1: A summary of existing constructions [17, 21, 3, 1] and our work, where KR-OWF and UR-OWF are the shorthands for known-regular and unknown-regular one-way functions respectively, ε-hard KR-OWF additionally assumes that the hardness parameter ε of KR-OWF is known, and n^{-c}-WUR-OWF is the shorthand for weakly unknown-regular one-way functions (see Footnote 2 and formally Definition 5.1).

	Assumption	Output Length	Key Length	# of Calls	Type of Call
[17]	OWP	$\Theta(n)$	$\Theta(n)$	1	non-adaptive
[21, 17]	1-to-1 OWF	$\Theta(n)$	$O(\omega(\log n)\cdot n)$	1	non-adaptive
[21]	KR-OWF	$O(\omega(\log^2 n)\cdot n)$	$O(\omega(\log^2 n)\cdot n)$	$O(\omega(\log n))$	adaptive
[3]	KR-OWF	$O(\omega(\log n)\cdot n)$	$O(\omega(\log n)\cdot n)$	$O(\omega(\log n))$	non-adaptive
[1]	UR-OWF	$\Theta(n)$	$O(\log n\cdot n)$	$\tilde{O}(n)$	adaptive
ours	1-to-1 OWF	$\Theta(n)$	$\Theta(n)$	1	non-adaptive
ours	ε-hard KR-OWF	$\Theta(n)$	$\Theta(n)$	1	non-adaptive
ours	KR-OWF	$O(\omega(1)\cdot n)$	$O(\omega(1)\cdot n)$	$O(\omega(1))$	non-adaptive
ours	n^{-c}-WUR-OWF	$\Theta(n)$	$O(\log n\cdot n)$	$\tilde{O}(n^{2c+1})$	adaptive

Summary of our constructions. In this paper, we give the following constructions from the respective aforementioned one-way functions. The first two constructions achieve optimal parameters simultaneously, the third is almost optimal up to an arbitrarily small super-constant factor $\omega(1)$ (e.g., $\log\log\log n$ or even less), and thus they all improve upon the respective known constructions. The fourth construction generalizes to beyond regular one-way functions (as introduced in [24]) with optimal output length $\Theta(n)$ and key length $O(n\cdot\log n)$. Although there remains a gap between this class of OWFs and arbitrary ones, our construction is nearly practical compared with the best generic construction with seed/key length $\tilde{O}(n^7)$ [13]. We remark that further improvement on key length $O(n\cdot\log n)$ of the fourth construction requires more key-length efficient domain extender (for Merkle-Damgård construction of UOWHFs) than Shoup's [22], which seems far beyond reach of conventional techniques. Finally, the first three constructions have optimal shrinkages (per invocation of OWF) by matching the upper bound of Gennaro et al. [5].

1. For any 1-to-1 one-way function, we construct an optimal family of UOWHFs with key and output length $\Theta(n)$ and a single OWF call. This improves the constructions of Naor and Yung [17] and De Santis and Yung [21] that require key length $O(n\cdot\omega(\log n))$.

2. For any known-regular one-way function with known hardness, we give another optimal construction of UOWHFs with key and output length $\Theta(n)$ and a single call.

3. For any known-regular one-way function, we give a construction of UOWHFs with key and output length $O(\omega(1)\cdot n)$ and $\omega(1)$ non-adaptive calls. This improves the construction of Barhum and Maurer [3] that requires key and output length $O(n\cdot\omega(\log n))$ and $\omega(\log n)$ calls.

4. For any one-way function f that is weakly unknown-regular on a noticeable fraction (i.e., n^{-c} for constant c) of domain [24], we give a construction of UOWHFs with key length $O(n\cdot\mathrm{log}n)$ and output length $\Theta(n)$. This generalizes the construction of Ames et al. [1] that assumes unknown-regular one-way functions (i.e., $c=0$).

On the (a)symmetry to PRGs. Our results further exhibit the inherent "black-box duality" [5, 13, 11] between UOWHFs and PRGs. Firstly, we abstract out a lemma about universal hashing (see Lemma 3.1) that is implicit in previous works [19, 16, 13] and plays a dual role in UOWHF constructions to the leftover hash lemma in PRG constructions. Informally, it says that when applying a universal hash function h to any random variable X of **max-entropy**[3] **no more than a** to produce an $(\mathbf{a}+\mathbf{d})$-bit output, h will be injective on X except for a $\mathbf{2^{-\Omega(d)}}$ fraction. In contrast, the leftover hash lemma states that when hashing any X of **min-entropy** (or Rényi entropy) **no less than a** into $(\mathbf{a}-\mathbf{d})$-bit strings, the resulting output distribution will be $\mathbf{2^{-\Omega(d)}}$-close (in terms of statistical distance) to uniform, where the symmetry is highlighted in bold. Secondly, constructions #2 and #3 above match the best known results about constructions of PRGs from known-regular OWFs (see [25]), namely, seed length $O(\omega(1)\cdot n)$ or even $\Theta(n)$ if the hardness of the underlying OWF is known. Thirdly, construction #4 is symmetric to the recent PRG construction [24] based on the same class of one-way functions with succinct key/seed length $O(n\cdot\log n)$. Finally (and perhaps more interestingly), construction #1 is asymmetric to the case of PRGs, where we do not know how to construct a linear seed length PRG from an arbitrary 1-to-1 one-way function[4].

On the efficiency, feasibility and limits. Constructions #1, #2 and #3 are practically relevant as most one-way function candidates turn out to be known-almost-regular or even 1-to-1. Goldreich, Levin and Nisan [8] showed how to base almost 1-to-1 (except for a negligible fraction) one-way functions on intractable problems such as RSA[5] and DLP, and thus construction #1 enables to build optimal UOWHFs from those problems. A byproduct of construction #2 is the equivalence of almost 1-to-1 one-way functions and known-(almost-)regular one-way functions in certain (known-hardness or non-uniform) settings, where we give an optimal construction of the former from the latter. Moreover, unknown regular one-way functions further reduce the knowledge required about the underlying one-way functions, and the problem of basing cryptographic primitives (PRGs, UOWHFs, etc.) on weaker assumptions is of theoretic interests. It improves our understanding about the feasibility and limits of black-box reductions. In particular, Holenstein and Sinha [14], Barhum and Holenstein [2] showed that $\Omega(n/\log n)$ black-box calls to an arbitrary (including unknown-regular) one-way function is necessary to construct PRGs and UOWHFs, and the lower bound is matched by explicit constructions of PRGs [10] and UOWHFs [1] respectively. The recent work of [24] carried on this line of research even further by considering a more general class of one-way functions (which they call weakly unknown-regular one-way functions), namely, the underlying one-way function can have an arbitrary structure as long as the set of x with maximal number of siblings (i.e., x and x' are siblings of each other if $f(x)=f(x')$) is of noticeable fraction. The authors of [24] gave a construction of PRG with seed length $O(n\cdot\log n)$ from weakly unknown-regular OWFs. However, their analysis is quite ad-hoc (see Remark 5.1), and doesn't seem to generalize to UOWHFs. As an intermediate step of construction #4, we prove that "iterating such a one-way function (weakly regular on only a noticeable fraction) polynomially many times yields a one-way function that is almost-regular on an overwhelming fraction" and thus unify the approach to the two dual objects (i.e., PRGs and UOWHFs). We mention an (arguable) analogue to this problem, namely, hardness amplification of one-way functions [23], where a function that is weakly one-way (for which every efficient algorithm has a noticeable fraction to fail upon) can be turned into strongly one-way (hard to invert on an overwhelming portion) by parallel repetition [23] or even sequential composition (assuming additionally that the underlying function is regular) [10].

The roadmap. We outline below the steps to build UOWHFs from the respective one-way function $f:\{0,1\}^n\rightarrow\{0,1\}^l$ introduced above. We first show that the following assumptions (about output

[3] X has max-entropy a if its support is of size 2^a.

[4] Given a 1-to-1 one-way function f, one might think of getting a PRG by hashing $f(U_n)$ into $n-s$ bits concatenated with $s+1$ hard-core bits of f, where $s\in\omega(\log n)$ is the necessary entropy loss due to the leftover hash lemma. This is in general not possible without knowing the exact hardness of the underlying f. See more discussions and the relaxed solutions to this problem by Goldreich [6, Section 3.5.1.3]. For example, we get a linear seed-length PRG of the following weaker form, i.e., for every $\varepsilon=1/\mathsf{poly}(n)$ there exists a weak PRG of seed length $\Theta(n)$ whose output distribution is ε-indistinguishable from uniform to all PPT distinguishers. Alternatively, we use parallel repetition to obtain a standard PRG with seed length $O(\omega(1)\cdot n)$ [25].

[5] RSA is typically known as a (sequence of) family of trapdoor permutations, which was transformed into a single (sequence of) 1-to-1 one-way function in [8].

length) can be made without loss of generality: $l \in O(n)$ for 1-to-1 one-way functions and length-preserving-ness (i.e., $l = n$) for arbitrary one-way functions. We state this as Fact 1 with a full proof given in Appendix A. Haitner et al. [10] showed that any one-way function implies a length-preserving one-way function, and we show in Fact 1 an even stronger version that (1) any 1-to-1 one-way function $f : \{0,1\}^n \to \{0,1\}^l$ implies a one-way function $f' : \{0,1\}^{n' \in \Theta(n)} \to \{0,1\}^{l' \in \Theta(n)}$ that is 1-to-1 except for a negligible fraction; (2) any one-way function f with $\alpha \leq |f^{-1}(y)| \leq \alpha \cdot \beta$ implies another length-preserving one-way function $f' : \{0,1\}^{n' \in \Theta(n)} \to \{0,1\}^{n'}$ with $\alpha' \leq |f'^{-1}(y)| \leq \alpha' \cdot \beta$ except for a negligible fraction, where the size of range β is preserved, and α' is efficiently computable if α is.

Based on 1-to-1 OWFs. We adapt Naor-Yung's elegant "hash-then-truncate" approach (for one-way permutation) to any 1-to-1 one-way function:

$$\mathcal{G}_1 \stackrel{\mathsf{def}}{=} \{ \ (\mathsf{trunc} \ \circ \ h \ \circ \ f) : \{0,1\}^n \to \{0,1\}^{n-s} \ , \ h \in \mathcal{H} \ \} \ ,$$

where $\mathcal{H}$ is a family of universal[6] hash permutations on l bits, and $\mathsf{trunc} : \{0,1\}^l \to \{0,1\}^{n-s}$ is a truncating function that outputs the first $n - s$ bits of input. We show that if f is a (t,ε)- 1-to-1 OWF then the resulting $\mathcal{G}_1$ is a $(t - n^{O(1)},\, 2^{s+1} \cdot \varepsilon)$-UOWHF family with key and output length $\Theta(n)$ and shrinkage s (see Definition 2.3 and Definition 2.7 for formal definitions). The construction enjoys optimal parameters and somewhat counter-intuitively the security bound drops only by factor 2^s (which is optimal by [5]) rather than by 2^{l-n+s} (i.e., exponential in the number of bits truncated which would render the construction useless). We refer to the proof of Theorem 3.1 and Remark 3.1 for more technical details and further discussions.

Based on known-(almost-)regular ε-hard OWFs. Given an almost-regular f (see Definition 2.6) which is known to be (t,ε)-one-way for some efficiently computable ε, we define the following function family

$$\mathcal{G}_2 \stackrel{\mathsf{def}}{=} \{ \ g : \{0,1\}^n \to \{0,1\}^{n-s}, g(x) = (\ \mathsf{trunc}(h(f(x))), h_1(x) \ ), h \in \mathcal{H}, h_1 \in \mathcal{H}_1 \ \}$$

where $\mathcal{H}$ is a family of universal hash permutations, and let $\mathcal{H}_1$ and trunc be a family of universal hash functions and the truncating function (both with appropriate output sizes) respectively. We show that $\mathcal{G}_2$ is a UOWHF family with key and output length $\Theta(n)$ and shrinkage s. The rationale is that for any[7] $x \neq x'$ colliding on $g \in \mathcal{G}_2$ it either satisfies "$f(x) = f(x') \wedge h_1(x) = h_1(x')$" or "$f(x) \neq f(x') \ \wedge \ \mathsf{trunc}(h(f(x))) = \mathsf{trunc}(h(f(x')))$". The former is unconditionally bounded by universal hashing, and the latter is computationally bounded (and reducible to the one-way-ness of f). Interestingly, by abstracting out function $f'(x, h_1) \stackrel{\mathsf{def}}{=} (f(x), h_1(x), h_1)$ from the above construction, we further show that f' is a one-way function that is 1-to-1 except for a negligible fraction. We refer to Theorem 4.1, Lemma 4.1 and Theorem 4.2 for the details.

Based on known-(almost-)regular OWFs. Next, we consider any known-(almost)-regular OWF f whose hardness parameter is ε unknown (i.e., ε is negligible but may not be efficiently computable). In this case, we run q independent copies of f, and we get a construction by making q non-adaptive calls with shrinkage $q \log n$, key and output length $O(q \cdot n)$, where $q \in \omega(1)$ can be any efficiently computable super-constant. The parallel repetition technique was also used in similar contexts (e.g., the construction of PRG from any known regular OWF [25]). We refer to Theorem 4.3 for the detailed construction and proof.

Based on a more general class of OWFs. We show iterating the class of one-way functions introduced in [24] sufficiently many times yields a one-way function f' that is almost-regular, and thus plugging this f' into the construction of Ames et al.[1] yields a construction of UOWHFs with output length $\Theta(n)$ and key length $O(n \cdot \log n)$.

[6] Many existing UOWHF constructions use pairwise (or even 3-wise) independent hashing to facilitate the analysis, but in fact universal hashing suffices here.

[7] More precisely, x is sampled at random and x' can be any distinct value (i.e., $x' \neq x$) efficiently computable from x and g.

2 Preliminaries

Notations and definitions. We use $[n]$ to denote set $\{1, \ldots, n\}$. We use capital letters (e.g., X, Y) for random variables, standard letters (e.g., x, y) for values, and calligraphic letters (e.g. $\mathcal{X}$, $\mathcal{Y}$) for sets. The support of a random variable X, denoted by $\mathsf{Supp}(X)$, refers to the set of values on which X takes with non-zero probability, i.e., $\{x : \Pr[X = x] > 0\}$. For a binary string $x = x_1 \ldots x_n$, denote by $x_{[t]}$ the first t bits of x, i.e., $x_1 \ldots x_t$. $x\|y$ refers the concatenation of x and y. We denote by $\mathsf{trunc} : \{0,1\}^n \to \{0,1\}^t$ a truncating function that outputs the first t bits of input, i.e., $\mathsf{trunc}(x) = x_{[t]}$. $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. For function $f : \{0,1\}^n \to \{0,1\}^{l(n)}$, we use shorthand $f(\{0,1\}^n) \stackrel{\mathsf{def}}{=} \{f(x) : x \in \{0,1\}^n\}$, and denote by $f^{-1}(y)$ the set of y's preimages under f, i.e., $f^{-1}(y) \stackrel{\mathsf{def}}{=} \{x : f(x) = y\}$. We say f is length-preserving if $l(n) = n$. We use $s \leftarrow S$ to denote sampling an element s according to distribution S, and let $s \xleftarrow{\$} \mathcal{S}$ denote sampling s uniformly from set $\mathcal{S}$, and $y := f(x)$ denote value assignment. We use U_n and $U_{\mathcal{X}}$ to denote uniform distributions over $\{0,1\}^n$ and $\mathcal{X}$ respectively, and let $f(U_n)$ be the distribution induced by applying function f to U_n. For probabilistic algorithm A, we use $\mathsf{A}(x; r)$ to denote the output of A on input x and internal coin r. The min-entropy and max-entropy (see, e.g., [13]) of a random variable X, denoted by $\mathbf{H}_\infty(X)$ and $\mathbf{H}_0(X)$ respectively, are defined as:

$$\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} \log \min_{x \in \mathsf{Supp}(X)} \frac{1}{\Pr[X = x]} \ ; \quad \mathbf{H}_0(X) \stackrel{\text{def}}{=} \log |\mathsf{Supp}(X)| \ .$$

We use '$+/-$' and '$\cdot$' for addition/subtraction and multiplication between field elements respectively. The zero element of any finite field is denoted by $\vec{0}$.

Collision probability. We use $\mathsf{CP}(X)$ to denote the collision probability of X, i.e., $\mathsf{CP}(X) \stackrel{\text{def}}{=} \sum_x \Pr[X = x]^2$, and denote by $\mathsf{CP}(X|Z)$ the average collision probability of X conditioned on another (possibly correlated) random variable Z by

$$\mathsf{CP}(X|Z) \stackrel{\text{def}}{=} \mathbb{E}_{z \leftarrow Z}\left[\ \sum_x \Pr[X = x| \ Z = z]^2 \ \right] \quad .$$

Simplifying Notations. To simplify the presentation, we use the following simplified notations. Throughout, most parameters are functions of the security parameter n (e.g., $t(n)$, $\varepsilon(n)$, $r(n)$) and we often omit n when clear from the context (e.g., t, ε, r). Parameters (e.g., ε, r) are said to be known if they are polynomial-time computable from n. By notation $f : \{0,1\}^n \to \{0,1\}^l$ we refer to the ensemble of functions $\{f : \{0,1\}^n \to \{0,1\}^{l(n)}\}_{n \in \mathbb{N}}$. As slight abuse of notion, poly might be referring to the set of all polynomials or a certain polynomial, and h might be either a function or its description which will be clear from context. For example, in $h(y) \stackrel{\mathsf{def}}{=} h \cdot y$ the first h denotes a function, the second h refers to a string (a finite field element) that describes the function (i.e., multiplying y by h).

Definition 2.1 (ρ-almost universal hashing) *A family of functions $\mathcal{H} = \{h : \{0,1\}^l \to \{0,1\}^t\}$ is ρ-almost universal if for any distinct $x_1, x_2 \in \{0,1\}^l$, it holds that*

$$\Pr_{h \xleftarrow{\$} \mathcal{H}} [h(x_1) = h(x_2)] \le \rho \ .$$

In the special case $\rho = 2^{-t}$, we say that $\mathcal{H}$ is universal.

Definition 2.2 (pairwise independent hashing) *A family of functions $\mathcal{H} = \{h : \{0,1\}^l \to \{0,1\}^t\}$ is pairwise independent if any distinct $x_1, x_2 \in \{0,1\}^l$ and any $v_1, v_2 \in \{0,1\}^t$ it holds that $\Pr_{h \xleftarrow{\$} \mathcal{H}}[\ h(x_1) = v_1 \wedge h(x_2) = v_2 \ ] = 2^{-2t}$.*

Definition 2.3 (one-way functions) *A sequence of functions* $\{f : \{0,1\}^n \to \{0,1\}^{l(n)}\}_{n\in\mathbb{N}}$ *is* $(t(n),\varepsilon(n))$*-one-way if* f *is polynomial-time computable and for any probabilistic algorithm* A *of running time* $t(n)$

$$\Pr_{x\xleftarrow{\$}\{0,1\}^n}[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))] \;\leq\; \varepsilon(n).$$

Hereafter we use simplified notation $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ *for the above one-way function, where* $t(\cdot)$ *and* $1/\varepsilon(\cdot)$ *are super-polynomial.*

Definition 2.4 (a family of one-way functions) *A sequence of function family* $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$*, where* $\mathcal{F}_n = \{f_u : \{0,1\}^n \to \{0,1\}^{l(n)}, u \in \{0,1\}^{q(n)}\}$*, is* $(t(n),\varepsilon(n))$*-one-way if for any* $n \in \mathbb{N}$*,* $u \in \{0,1\}^{q(n)}$ *and* $x \in \{0,1\}^n$*, the value* $f_u(x)$ *can be computed in polynomial time, and for any probabilistic algorithm* A *of running time* $t(n)$*, we have that*

$$\Pr_{x\xleftarrow{\$}\{0,1\}^n;\; u\xleftarrow{\$}\{0,1\}^{q(n)}}[\;\mathsf{A}(1^n, u, f_u(x)) \in f_u^{-1}(f_u(x))\;] \;\leq\; \varepsilon(n) \quad .$$

We use shorthands $\mathcal{F} = \{f_u : \{0,1\}^n \to \{0,1\}^{l(n)}, u \in \{0,1\}^{q(n)}\}$ *for* $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$*.*

Definition 2.5 (almost 1-to-1 functions) *A function* $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ *is* $\varepsilon(n)$*-almost 1-to-1 if there exists a negligible function* $\varepsilon(n)$*, such that for every* $n \in \mathbb{N}$ *we have*

$$\Pr_{x\xleftarrow{\$}\{0,1\}^n}[\;\exists x' :\; x' \neq x \,\wedge\, f(x) = f(x')\;] \leq \varepsilon(n).$$

In particular, f *is 1-to-1 if* $\varepsilon(n) \equiv 0$*.*

Definition 2.6 (almost regular functions) *For integer functions* $\alpha = \alpha(n)$ *and* $\beta = \beta(n)$*, a function* $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ *is* α*-regular if for every* $n \in \mathbb{N}$ *and* $x \in \{0,1\}^n$ *we have*

$$|f^{-1}(f(x))| = \alpha.$$

f *is* $(\alpha,\, \alpha{\cdot}\beta)$*-almost regular if for every* $n \in \mathbb{N}$ *and* $x \in \{0,1\}^n$ *we have*

$$\alpha \;\leq\; |f^{-1}(f(x))| \;\leq\; \alpha \cdot \beta.$$

In particular, f *is known-(almost)-regular if* α *is polynomial-time computable, or otherwise it is called unknown-(almost)-regular. Standard "almost-regularity" for a* (t,ε)*-one-way function* f *refers to that* f *is* $(\alpha,\, \alpha{\cdot}\beta)$*-almost-regular for* $\beta = \mathsf{poly}(n)$ *or at most* $\beta = (1/\varepsilon)^{\Theta(1)}$ *for certain small constant* $0 < \Theta(1) < 1$*.*

Definition 2.7 (UOWHFs [17]) *A sequence of function family* $\mathcal{G} = \{\mathcal{G}_n\}_{n\in\mathbb{N}}$*, where* $\mathcal{G}_n = \{g_u : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)-s(n)}, u \in \{0,1\}^{q(n)}, \ell \in \mathsf{poly}\}$*, is a family of* $(t(n),\varepsilon(n))$*-universal one-way hash functions if for every* $n \in \mathbb{N}$*,* $u \in \{0,1\}^{q(n)}$ *and* $x \in \{0,1\}^{\ell(n)}$*, the value* $g_u(x)$ *can be computed in polynomial time, and for every probabilistic algorithm* A *of running time* $t(n)$*, it holds that*

$$\Pr_{x\xleftarrow{\$}\{0,1\}^{\ell(n)};\; u\xleftarrow{\$}\{0,1\}^{q(n)};\; x'\leftarrow\mathsf{A}(1^n,x,u)}[\;x \neq x' \,\wedge\, g_u(x) = g_u(x')\;] \;\leq\; \varepsilon(n) \quad .$$

The difference between input and output lengths (i.e., $s(n)$*) is called* ***shrinkage****. For succinctness, hereafter we will use shorthand* $\mathcal{G} = \{g_u : \{0,1\}^{\ell} \to \{0,1\}^{\ell-s},\, u \in \{0,1\}^q\}$ *for* $\{\mathcal{G}_n\}_{n\in\mathbb{N}}$ *defined above.*

3 UOWHFs from 1-to-1 One-way Functions

3.1 A Technical Lemma and Its Applications

We state a technical lemma about universal hashing (see Lemma 3.1) below and it plays a symmetric role in the construction of UOWHFs to that the leftover hash lemma does in PRGs. The proof follows from the universality of the hash functions and we include it in Appendix A.1 for completeness.

Lemma 3.1 (The injective hash lemma) *For any integers a, d, k and l satisfying $a \le l$, let Y be any random variable over $\{0,1\}^l$ with $\mathbf{H}_0(Y) \le a$, and let $\mathcal{H} \stackrel{\mathsf{def}}{=} \{h : \{0,1\}^l \to \{0,1\}^{a+d}\}$ be a family of $(k \cdot 2^{-(a+d)})$-almost universal hash functions. Then, we have that*

$$\Pr_{y \leftarrow Y,\ h \xleftarrow{\$} \mathcal{H}} [\ \exists \tilde{y} \in \mathsf{Supp}(Y) :\ \tilde{y} \neq y \wedge h(\tilde{y}) = h(y)\ ] \le k \cdot 2^{-d}\ .$$

Recall that $k = 1$ corresponds to the special case that $\mathcal{H}$ is universal.

In addition to being a technical tool for UOWHF constructions, Lemma 3.1 is also used to reduce the output lengths of one-way functions without loss generality. That is, the input and output lengths of a 1-to-1 one-way function $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ can be assumed to be linearly related (i.e., $l(n) = O(n)$). For almost regular one-way functions, we can even assume that they are length-preserving (i.e., $l(n) = n$). We refer to Appendix A.1 for the proof of Fact 1.

Fact 1 *For any $r_1 = r_1(n)\ \le\ r_2 = r_2(n)$ and any efficiently computable $\kappa = \kappa(n) \in O(n)$, we have*

1. *Any 1-to-1 (t,ε)-one-way function $f : \{0,1\}^n \to \{0,1\}^l$ implies a $(t - n^{O(1)},\ \varepsilon + \mathsf{poly}(n) \cdot 2^{-\kappa})$-one-way function $f' : \{0,1\}^{n' \in \Theta(n)} \to \{0,1\}^{(n'+\kappa) \in \Theta(n)}$ which is 1-to-1 except on a $(\mathsf{poly}(n) \cdot 2^{-\kappa})$-fraction of inputs, i.e.,*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^{n'}} [\ \exists x' \in \{0,1\}^{n'} : x' \neq x\ \wedge\ f'(x) = f'(x')\ ]\ \le\ \mathsf{poly}(n) \cdot 2^{-\kappa}$$

2. *Any $(2^{r_1}, 2^{r_2})$-almost regular (t,ε)-one-way function $f : \{0,1\}^n \to \{0,1\}^l$ implies a length-preserving $(t - n^{O(1)}, \varepsilon + \mathsf{poly}(n) \cdot 2^{-(r_1+\kappa)})$-one-way function $\bar{f} : \{0,1\}^{n' \in \Theta(n)} \to \{0,1\}^{n'}$ which is $(2^{\kappa+r_1}, 2^{\kappa+r_2})$-almost regular except on a $(\mathsf{poly}(n) \cdot 2^{-(r_1+\kappa)})$-fraction of inputs, i.e.,*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^{n'}} [\ 2^{\kappa+r_1}\ \le\ |\bar{f}^{-1}(\bar{f}(x))|\ \le\ 2^{\kappa+r_2}\ ]\ \ge\ 1 - \mathsf{poly}(n) \cdot 2^{-(r_1+\kappa)}\ .$$

Note that $2^{r_2-r_1}$ is arbitrary (i.e., not necessarily bounded by $\mathsf{poly}(n)$), and thus the second statement applies to any one-way function f. It suffices to set $\kappa = \omega(\log n)$ to have a negligible error bound.

Therefore, we will assume in the remainder of the paper that the underlying 1-to-1 one-way function has linear output length (i.e., $l(n) = O(n)$) and that the almost-regular and weakly unknown-regular one-way functions are length-preserving (i.e., $l(n) = n$).

3.2 UOWHFs from 1-to-1 OWFs

For a 1-to-1 OWF $f : \{0,1\}^n \to \{0,1\}^l$, we define a cryptographic game between a challenger C and an inverter Inv. That is, C samples a random $y^* \xleftarrow{\$} \{0,1\}^l$ and sends it to Inv, and Inv wins the game iff he comes up with any x' satisfying $f(x') = y^*$. Note that even unbounded Inv wins this game with advantage no more than $2^{-(l-n)}$ (which is probability that $y^* \in f(\{0,1\}^n)$), and Fact 2 states that the chance to win is even smaller for computationally bounded Inv.

Fact 2 *For any 1-to-1 (t,ε)-one-way function $f:\{0,1\}^n\rightarrow\{0,1\}^l$ and any probabilistic algorithm* Inv *of running time t, it holds that*

$$\Pr_{y^*\xleftarrow{\$}\{0,1\}^l}[\ f(\mathsf{Inv}(y^*))=y^*\ ]\ \leq\ 2^{-(l-n)}\cdot\varepsilon\quad .$$

Proof.

$$\Pr_{y^*\xleftarrow{\$}\{0,1\}^l}[\ f(\mathsf{Inv}(y^*))=y^*\ ]\leq\ \Pr_{y^*\xleftarrow{\$}\{0,1\}^l}[y^*\in f(\{0,1\}^n)]\cdot\Pr_{y^*\xleftarrow{\$}f(\{0,1\}^n)}[\ f(\mathsf{Inv}(y^*))=y^*\ ]\leq\ 2^{-(l-n)}\cdot\varepsilon\quad .$$

□

Remark 3.1 (on the proof sketch of Theorem 3.1) *We use a trick to prove Theorem 3.1. We show that any* A *that ε'-breaks the TCR of the constructed UOWHF implies an* $\mathsf{Inv}^{\mathsf{A}}$ *(of almost the same efficiency as* A*) that wins the above game (i.e., inverting f on a random $y^*\in\{0,1\}^l$) with advantage roughly $2^{n-l-s}\cdot\varepsilon'$. This may seem useless since $l-n$ can be $\Omega(n)$ or even* $\mathsf{poly}(n)$*. However, by Fact 2 this term (i.e., $2^{n-l-s}\cdot\varepsilon'$) is actually upper bounded by $2^{-(l-n)}\cdot\varepsilon$. The conclusion $\varepsilon'\leq 2^s\varepsilon$ immediately follows by cancelling the factor $(l-n)$. In other words, the security bound does not depend on the number of bits truncated (i.e., $l-n+s$), but only on shrinkage s, and it is tight due to [5].*

Theorem 3.1 (UOWHFs from 1-to-1 OWFs) *Let $f:\{0,1\}^n\rightarrow\{0,1\}^{l\in O(n)}$ be any 1-to-1 (t,ε)-one-way function, let $\mathcal{H}$ be a family of permutations*[8] *over $\{0,1\}^l$ as follows:*

$$\mathcal{H}=\{h:\{0,1\}^l\rightarrow\{0,1\}^l\ ,\ h(y)\stackrel{\mathsf{def}}{=}h\cdot y,\quad \textit{where}\ \ y\in GF(2^l),\ \vec{0}\neq h\in GF(2^l)\ \}\ ,$$

let $\mathsf{trunc}:\{0,1\}^l\rightarrow\{0,1\}^{n-s}$ *be a truncating function, where $s=s(n)$ is efficiently computable. Then, we have that*

$$\mathcal{G}_1\stackrel{\mathsf{def}}{=}\{\ (\mathsf{trunc}\circ h\circ f\ ):\{0,1\}^n\rightarrow\{0,1\}^{n-s}\ ,\ h\in\mathcal{H}\ \}$$

is a family of $(t-n^{O(1)},\,2^{s+1}\cdot\varepsilon)$-UOWHFs with key and output length $\Theta(n)$, and shrinkage s.

Proof. Suppose for contradiction that there exists a $\mathcal{G}_1$-collision finder A of running time t' that on input (x,h), breaks the target collision resistance with some non-negligible probability ε', i.e.,

$$\Pr_{x\xleftarrow{\$}\{0,1\}^n,h\xleftarrow{\$}\mathcal{H}}[\ x'\leftarrow\mathsf{A}(x,h):\ x\neq x'\wedge h(f(x))_{[n-s]}=h(f(x'))_{[n-s]}\ ]\ =\ \varepsilon'\ >\ 2^{s+1}\cdot\varepsilon$$

We define algorithm $\mathsf{Inv}^{\mathsf{A}}$ (that inverts f on input $y^*\xleftarrow{\$}\{0,1\}^l$ by invoking A) as in Algorithm 1. Define event $\mathcal{E}_{\mathsf{neq}}\stackrel{\mathrm{def}}{=}(f(x)\neq y^*)$. We argue that $\mathsf{Inv}^{\mathsf{A}}$ inverts f with the following probability (see the rationale below)

$$\begin{aligned}
&\Pr_{y^*\xleftarrow{\$}\{0,1\}^l,\ x\xleftarrow{\$}\{0,1\}^n,\ v\xleftarrow{\$}\mathcal{V}}[\ f(\mathsf{Inv}^{\mathsf{A}}(y^*))=y^*\ ]\\
\geq\quad&\Pr_{y^*\xleftarrow{\$}\{0,1\}^l,\ x\xleftarrow{\$}\{0,1\}^n}[\ \mathcal{E}_{\mathsf{neq}}\ ]\quad\cdot\quad\Pr_{y^*\xleftarrow{\$}\{0,1\}^l\setminus\{f(x)\},\ x\xleftarrow{\$}\{0,1\}^n,\ v\xleftarrow{\$}\mathcal{V}}[\ f(\mathsf{Inv}^{\mathsf{A}}(y^*))=y^*\mid\mathcal{E}_{\mathsf{neq}}\ ]\\
\geq\quad&(1-2^{-l})\cdot\Pr_{x\xleftarrow{\$}\{0,1\}^n,h\xleftarrow{\$}\mathcal{H},x'\leftarrow\mathsf{A}(x,h),v\xleftarrow{\$}\mathcal{V}}[x\neq x'\ \wedge\ h(f(x))_{[n-s]}=h(f(x'))_{[n-s]}\ \wedge\ y^*=f(x')\ ]\\
\geq\quad&(1-2^{-l})\cdot\varepsilon'\cdot\Pr_{v\xleftarrow{\$}\mathcal{V}}[y^*=f(x')\mid\mathcal{E}_{\mathsf{neq}}\ \wedge\ x\neq x'\wedge h(f(x))_{[n-s]}=h(f(x'))_{[n-s]}\ ]\\
=\quad&\frac{(1-2^{-l})\cdot\varepsilon'}{|\mathcal{V}|}\ =\ \frac{(1-2^{-l})\cdot\varepsilon'}{2^{l-n+s}-1}\ >\ \frac{\varepsilon'/2}{2^{l-n+s}}\ >\ \varepsilon\cdot 2^{-(l-n)}\ ,
\end{aligned}$$

[8] In fact, $\mathcal{H}$ constitutes a family of universal hash permutations. However, our proofs only use the concrete construction of $\mathcal{H}$ and benefit from its algebraic property over finite fields, rather than assuming a universal $\mathcal{H}$ plus a constructible property [13] (given any x and y there exists a PPT sampler to output $h\xleftarrow{\$}\{h\in\mathcal{H}:h(x)=y\}$).

Algorithm 1 $\mathsf{Inv}^{\mathsf{A}}$ that inverts f on input y^* using random coins (x, v).

Input: $y^* \xleftarrow{\$} \{0,1\}^l$

Sample $x \xleftarrow{\$} \{0,1\}^n$
if $f(x) = y^*$ **then**
  Output x and **terminate**.
end if
sample $h := (f(x) - y^*)^{-1} \cdot v$, where $v \xleftarrow{\$} \mathcal{V} = \{v \in \{0,1\}^l \setminus \{\vec{0}\} : v_{[n-s]} = \overbrace{0 \ldots 0}^{n-s}\}$
{The above implies $h \xleftarrow{\$} \{h \in \mathcal{H} : \ h(f(x))_{[n-s]} = h(y^*)_{[n-s]}\}$ by the $GF(2^l)$ arithmetics. }
$x' \leftarrow \mathsf{A}(x, h)$
if $f(x') = y^*$ **then**
  Output x'
else
  Output $\perp$
end if
Terminate

where the first inequality is straightforward (note that conditioned on $\mathcal{E}_{\mathsf{neq}}$ the sampling of y^* and x are uniform over $\{0,1\}^l \setminus \{f(x)\}$ and $\{0,1\}^n$ respectively), the second inequality follows from Claim 1, namely, conditioned on $\mathcal{E}_{\mathsf{neq}}$ it is equivalent to consider $(x,\ h,\ v) \xleftarrow{\$} \{0,1\}^n \times \mathcal{H} \times \mathcal{V}$ and then $y^* := f(x) - v \cdot h^{-1}$, and the third inequality is due to that A takes only x and h as input (i.e., independent of v). That is, conditioned on that A produces a valid $x' \neq x$ satisfying $h(f(x'))_{[n-s]} = h(f(x))_{[n-s]}$, we have by Claim 1 that string y^* is uniformly distributed over set $\mathcal{Y}^* \stackrel{\mathsf{def}}{=} \{f(x) - v \cdot h^{-1}, v \in \mathcal{V}\}$. Note that the already fixed $f(x')$ is also an element of $\mathcal{Y}^*$ and thus y^* hits $f(x')$ with probability $1/|\mathcal{Y}^*| = 1/|\mathcal{V}|$. We complete the proof by reaching a contradiction to Fact 2. □

Claim 1 (equivalent sampling) *Let the values h, v, x, y^* be sampled as in Algorithm 1 (or as in Algorithm 2), and conditioned on event $\mathcal{E}_{\mathsf{neq}} \stackrel{def}{=} (f(x) \neq y^*)$, it is equivalent to sample $(x,\ h,\ v) \xleftarrow{\$} \{0,1\}^n \times \mathcal{H} \times \mathcal{V}$ uniformly and independently and then determine $y^* := f(x) - v \cdot h^{-1}$.*

Proof of Claim 1. We know that (x, v) is uniformly sampled from $\{0,1\}^n \times \mathcal{V}$ by definition, and thus it suffices to show that "fix any (x, v), and conditioned on $y^* \neq f(x)$ (i.e., Y^* is uniform distributed over $\{0,1\}^l \setminus \{f(x)\}$), it holds that h is uniform over $\mathcal{H}$". This follows from that $v \neq \vec{0}$ ($\mathcal{V}$ excludes $\vec{0}$ by definition) and hence $h = (f(x) - Y^*)^{-1} \cdot v$ is uniform over $\{0,1\}^l \setminus \{\vec{0}\}$, namely, $h \xleftarrow{\$} \mathcal{H}$. Finally, for any given $(x,\ h,\ v)$, one efficiently determines the value $y^* = f(x) - v \cdot h^{-1}$ due to the arithmetics over the finite field. □

A simple corollary of Theorem 3.1 below assumes an almost 1-to-1 one-way function. See its proof in Appendix A.3.

Corollary 3.1 (UOWHFs from almost 1-to-1 OWFs) *Let f, $\mathcal{H}$, trunc and $\mathcal{G}_1$ be the same as assumed/defined in Theorem 3.1 except that f is δ-almost 1-to-1. Then, $\mathcal{G}_1$ is a family of $(t - n^{O(1)}$, $2^{s+1} \cdot \varepsilon + \delta)$-universal one-way hash functions with key and output length $\Theta(n)$, and shrinkage s.*

4 UOWHFs from Known Regular OWFs

We proceed to the more general case that f is a known almost-regular function. Recall that by Fact 1 we can assume WLOG that the underlying almost regular one-way function is length-preserving. We first show a construction where the hardness parameter ε is known, and then remove the dependency on ε.

4.1 Compressing the Output is Necessary but Not Sufficient

We attempt to generalize the Naor-Yung approach for one-way permutations (and 1-to-1 one-way functions) to almost regular one-way functions by compressing (using $\mathsf{trunc} \circ h$) the output $Y = f(X)$ into $\mathbf{H}_\infty(Y) - s'$ bits for $s' \in O(\log(1/\varepsilon))$. However, this only gives a weak form of guarantee, as stated in Lemma 4.1 below, that given a random x it is infeasible for efficient algorithms to find any $f(x') \neq f(x)$ such that $\mathsf{trunc}(h(f(x'))) = \mathsf{trunc}(h(f(x)))$. Otherwise said, it does not rule out the possibility that one may easily find $x' \neq x$ satisfying $f(x') = f(x)$. Hence, compressing the output is only a useful intermediate step to obtain UOWHFs. Lemma 4.1 below further generalizes Theorem 3.1 to known-(almost-)regular functions, whose proof is similar to that of Theorem 3.1 and thus we defer the redundancy to Appendix A.

Lemma 4.1 *For any constant c, any efficiently computable $r = r(n)$ and $s' = s'(n)$, let $f : \{0,1\}^n \to \{0,1\}^n$ be any $(2^r, 2^r n^c)$-almost regular (length-preserving) (t, ε)-one-way function, let $\mathcal{H}$ be a family of permutations over $\{0,1\}^n$ as below*

$$\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^n \ , \ h(y) \stackrel{\mathsf{def}}{=} h \cdot y, \quad \textit{where} \ \ y \in GF(2^n), \ \vec{0} \neq h \in GF(2^n) \ \} \ ,$$

let $\mathsf{trunc} : \{0,1\}^n \to \{0,1\}^{n-r-c\cdot\log n - s'}$ be a truncating function. Then, for any $\tilde{\mathsf{A}}$ of running time $t - n^{O(1)}$ (for some universal constant $O(1)$) we have that

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n, \ h \xleftarrow{\$} \mathcal{H}, \ x' \leftarrow \tilde{\mathsf{A}}(x,h)} [\ f(x) \neq f(x') \ \wedge \ \mathsf{trunc}(h(f(x))) = \mathsf{trunc}(h(f(x'))) \ ] \ \leq \ n^c \cdot 2^{s'+1} \cdot \varepsilon \ .$$

4.2 Known (Almost-)Regular OWFs with Known Hardness

We first give an optimal construction assuming that the inversion probability upper bound ε is known. Note that in addition to hashing the output $f(x)$ (as we did in Lemma 4.1), we also hash the input x to ensure that no distinct x' collides with x with respect to the resulting function.

Theorem 4.1 (UOWHFs from known-almost-regular ε-hard OWFs) *Let $f : \{0,1\}^n \to \{0,1\}^n$ be any $(2^r, 2^r n^c)$-almost regular (length-preserving) (t,ε)-one-way function as assumed in Lemma 4.1. Let shrinkage $s = s(n)$ be any efficiently computable function, and let $\mathcal{H}$ and trunc be as defined in Lemma 4.1 with $s' = (s + \log(1/\varepsilon) - c\log n)/2$, and let $\mathcal{H}_1 = \{h_1 : \{0,1\}^n \to \{0,1\}^{r + c\log n + s' - s}\}$ be a family of universal hash functions. Then, we have that*

$$\mathcal{G}_2 \stackrel{\mathsf{def}}{=} \{ \ g : \{0,1\}^n \to \{0,1\}^{n-s} \ , \ g(x) \stackrel{\mathsf{def}}{=} (g_1(x), h_1(x)), \ h \in \mathcal{H} \ , h_1 \in \mathcal{H}_1 \ \}$$

where $g_1 \stackrel{\mathsf{def}}{=} (\mathsf{trunc} \circ h \circ f)$, is a $(t - n^{O(1)}, O(\sqrt{2^s \cdot n^c \cdot \varepsilon}))$-universal one-way hash function family with key and output length $\Theta(n)$.

Proof. Define shorthands $\mathcal{E}_1 \stackrel{\text{def}}{=} \big(x \neq x' \wedge f(x) = f(x') \wedge h_1(x) = h_1(x')\big)$ and $\mathcal{E}_2 \stackrel{\text{def}}{=} \big(f(x) \neq f(x') \wedge g_1(x) = g_1(x')\big)$. For any $\mathcal{G}_2$-collision finder A, we have

$$\begin{aligned}
&\Pr_{x \xleftarrow{\$} \{0,1\}^n,\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1),\ x' \leftarrow \mathsf{A}(x,h,h_1)}[\ x \neq x' \ \wedge\ g(x) = g(x')\ ] \\
\leq\ & \Pr_{x \xleftarrow{\$} \{0,1\}^n,\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1),\ x' \leftarrow \mathsf{A}(x,h,h_1)}[\ \mathcal{E}_1 \ \vee\ \mathcal{E}_2\ ] \\
\leq\ & \Pr_{x \xleftarrow{\$} \{0,1\}^n,\ h_1 \xleftarrow{\$} \mathcal{H}_1}[\ \exists\, x' \neq x \ \wedge\ f(x) = f(x') \ \wedge\ h_1(x) = h_1(x')\ ] \\
&+ \Pr_{x \xleftarrow{\$} \{0,1\}^n,\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1),\ x' \leftarrow \mathsf{A}(x,h,h_1)}[\ f(x) \neq f(x') \ \wedge\ g_1(x) = g_1(x')\ ] \\
\leq\ & 2^{-(s'-s)} \ +\ n^c \cdot 2^{s'+1} \cdot \varepsilon \ =\ \sqrt{2^s \cdot n^c \cdot \varepsilon} \ +\ 2\sqrt{2^s \cdot n^c \cdot \varepsilon} \ =\ 3\sqrt{2^s \cdot n^c \cdot \varepsilon}\ ,
\end{aligned}$$

where the first inequality refers to that any collision on $g \in \mathcal{G}_2$ (for $x' \neq x$) must satisfy either $\mathcal{E}_1$ or $\mathcal{E}_2$ and the second inequality follows by a union bound. We already know by Lemma 4.1 that the second term is bounded by $n^c \cdot 2^{s'+1}\varepsilon$, and it thus remains to show that the first term is bounded by $2^{-(s'-s)}$. Conditioned on any $y = f(X)$ random variable X is a flat distribution on a set of size at most $2^r \cdot n^c$, so we apply Lemma 3.1 (setting $a = r + c \cdot \log n$, $d \geq s' - s$ and $k = 1$) to get

$$\begin{aligned}
&\Pr_{x \xleftarrow{\$} \{0,1\}^n,\ h_1 \xleftarrow{\$} \mathcal{H}_1}[\ \exists\, x' \neq x \ \wedge\ f(x) = f(x') \ \wedge\ h_1(x) = h_1(x')\ ] \\
&= \ \mathbb{E}_{y \leftarrow f(U_n)}\Big[\ \Pr_{x \xleftarrow{\$} f^{-1}(y),\ h_1 \xleftarrow{\$} \mathcal{H}_1}[\ \exists\, x' \neq x \ \wedge\ f(x) = f(x') \ \wedge\ h_1(x) = h_1(x')\ ]\ \Big] \\
&\leq \ \mathbb{E}_{y \leftarrow f(U_n)}[\ 2^{-(s'-s)}\ ] \ = 2^{-(s'-s)}\ ,
\end{aligned}$$

which completes the proof. □

4.3 An Alternative Approach to Section 4.2

A neater (and perhaps more intuitive) approach is to construct an almost 1-to-1 one-way function f' (with input and output lengths $\Theta(n)$) based on f (stated as Theorem 4.2) and then plug f' into Corollary 3.1 (using f' in place of f). This statement is interesting in its own right as it implies that almost 1-to-1 one-way functions and known-(almost-)regular one-way functions (with known hardness) are equivalent. Taking a closer look at Theorem 4.2 we find that this almost 1-to-1 f' is also present (as an intermediate function) in construction $\mathcal{G}_2$ of Theorem 4.1 (except with slightly different length parameters). Lemma 4.2 and Lemma 4.3 state the almost injectiveness and one-way-ness of f' respectively, for which we determine a judicious value for d (assuming knowledge about ε) in Theorem 4.2 to achieve injectiveness and one-way-ness simultaneously.

Theorem 4.2 (almost 1-to-1 OWF from almost-regular ε-hard OWF) *Let $f : \{0,1\}^n \to \{0,1\}^n$ be any $(2^r, 2^r n^c)$-almost regular (length-preserving) (t,ε)-one-way function as assumed in Lemma 4.1. For efficiently computable $d = d(n) \in \mathbb{N}$, define*

$$f' : \{0,1\}^n \times \mathcal{H}_1 \to \{0,1\}^n \times \{0,1\}^{r+c\cdot\log n+d} \times \mathcal{H}_1$$

$$f'(x, h_1) \stackrel{\text{def}}{=} (f(x), h_1(x), h_1)$$

where $\mathcal{H}_1$ is a family of universal hash functions from n bits to $r + c\cdot\log n + d$ bits. Then, for $d = \frac{\log(1/\varepsilon) - c\cdot\log n - 3}{3}$ we have that f' is $2\sqrt[3]{\varepsilon \cdot n^c}$-almost 1-to-1 and $(t - O(n),\ 2\sqrt[3]{\varepsilon \cdot n^c})$-one-way with input and output lengths $\Theta(n)$.

Proof. The almost 1-to-1-ness and one-way-ness of f' follow from Lemma 4.2 and Lemma 4.3 respectively by setting parameter $d = \frac{\log(1/\varepsilon) - c \cdot \log n - 3}{3}$. □

We give the proofs of Lemma 4.2 and Lemma 4.3 in Appendix A.3.

Lemma 4.2 (f' is almost 1-to-1) *f' defined in Theorem 4.2 is 2^{-d}-almost 1-to-1.*

Lemma 4.3 (f' is one-way) *f' defined in Theorem 4.2 is a $(t - O(n),\ \sqrt{2^{d+3} \cdot n^c \cdot \varepsilon})$-one-way function.*

4.4 UOWHFs from any Known (Almost-)Regular OWFs

Removing the dependency on ε. Unfortunately, Theorem 4.1 doesn't immediately apply to an arbitrary regular function as in general we assume no knowledge about ε (other than that ε is negligible). To see the difficulty, check the proof of Theorem 4.1 where the security of the resulting UOWHF is bounded by the sum of two terms, i.e., $2^{-(s'-s)} + n^c{\cdot}2^{s'+1} \cdot \varepsilon$. Without knowing ε, one may end up setting some super-polynomial $2^{s'}$ (to make the first term negligible) which kills the second term $n^c{\cdot}2^{s'+1} \cdot \varepsilon$. Same problems arise in similar situations (e.g., construction of PRGs from regular OWFs [25]). A remedy for this is parallel repetition: run $q \in \omega(1)$ copies of f on $\vec{x} = (x_1, \ldots, x_q)$, apply hash-then-truncate (setting $s' = 2\log n$) to every copy $f(x_i)$, which shrinks the entropies by $2q\log n$ bits and yields a bound $O(\varepsilon{\cdot}n^{c+2})$. Next, apply a single hashing to $\vec{x}$ that expands $q{\cdot}\log n$ bits (to yield another negligible term n^{-q}). This gives a family of UOWHFs with shrinkage $2q\log n - q\log n = q\log n$, and key and output length $O(q \cdot n)$ for any (efficiently computable) $q \in \omega(1)$. The proof is similar in spirit to that of Theorem 4.1 and we include it in Appendix A.4 due to lack of space.

Definition 4.1 (parallel repetition) *For any function $g : \mathcal{X} \to \mathcal{Y}$, we define its q-fold parallel repetition $g^q : \mathcal{X}^q \to \mathcal{Y}^q$ as*

$$g^q(x_1, ..., x_q) = (\ g(x_1)\ , ...,\ g(x_q)\ )\quad .$$

For simplicity, we use shorthand $\vec{x} \stackrel{def}{=} (x_1, \ldots, x_q)$ and thus $g^q(\vec{x}) {=} g^q(x_1, \ldots, x_q)$.

Theorem 4.3 (UOWHFs from any known almost-regular OWFs) *Let $f : \{0,1\}^n \to \{0,1\}^n$ be any $(2^r, 2^r n^c)$-almost regular (length-preserving) (t,ε)-one-way function as assumed in Lemma 4.1. Then, for any efficiently computable $q = q(n) = \omega(1)$, let $\mathcal{H}$ and trunc be as defined in Lemma 4.1 with $s' = 2\log n$, and let $\mathcal{H}_1 = \{h_1 : \{0,1\}^{q \cdot n} \to \{0,1\}^{q(r+(c+1)\log n)}\}$ be a family of universal hash functions, we have that*

$$\mathcal{G}_3 \stackrel{\mathsf{def}}{=} \{\ g : \{0,1\}^{qn} \to \{0,1\}^{qn - q\log n}\ ,\ g(\vec{x}) \stackrel{\mathsf{def}}{=} (g_1(\vec{x}), h_1(\vec{x})),\ h \in \mathcal{H}\ , h_1 \in \mathcal{H}_1\ \}$$

where $g_1 \stackrel{\mathsf{def}}{=} (\mathsf{trunc} \circ h \circ f)^q$, is a $(t - n^{O(1)}, n^{-q} + 2q{\cdot}n^{c+2} \cdot \varepsilon)$-universal one-way hash function family with key and output length $O(q \cdot n)$, and shrinkage $q{\cdot}\log n$.

5 Going Beyond Almost-Regular OWFs

Although (almost) optimal, our foregoing constructions need at least almost-regularity, i.e., the one-way function f satisfies $\alpha \le |f^{-1}(f(x))| \le \alpha \cdot \beta$ for all (or at least an overwhelming portion of) x, where α is efficiently computable and $\beta = \mathsf{poly}(n)$ (or at most $\beta = O(\log(1/\varepsilon))$ for an $(\varepsilon^{-1},\varepsilon)$-hard f). Complementary to our work, Ames et al. [1] gave an elegant construction from unknown-(almost-)regular one-way functions (see Appendix B.1), namely, without knowledge about α, for which they pay a cost of much increased number of one-way function calls (i.e., $O(n/\mathrm{log}n)$, which is necessary due to [2]) and key length $O(n\log n)$. In this section, we further weaken the assumption so that f can have an arbitrary structure (i.e., β is not bounded) as long as the fraction of x's with (nearly) maximal number of siblings is noticeable.

5.1 A More General Class of OWFs

The following class of one-way functions was introduced in [24] as a relaxation to unknown-(almost-)regular one-way functions.

Definition 5.1 (weakly unknown-regular OWFs [24]) *Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function, and for every $n \in \mathbb{N}$, divide domain $\{0,1\}^n$ into sets $\mathcal{X}_1, \ldots, \mathcal{X}_n$ (i.e., $\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_n = \{0,1\}^n$) such that $\mathcal{X}_j \stackrel{\mathsf{def}}{=} \{x : 2^{j-1} \le |f^{-1}(f(x))| < 2^j\}$, and define $\max = \max(n)$ to be the maximal subscript of the non-empty sets, i.e., $|\mathcal{X}_{\max}| > 0$ and $|\mathcal{X}_{\max+1} \cup \ldots \cup \mathcal{X}_n| = 0$. We say that f is **weakly unknown-regular** if there exists a constant c such that for all sufficiently large n :*

$$\Pr[U_n \in \mathcal{X}_{\max}] \ge n^{-c} \ . \tag{1}$$

Note that $\max(\cdot)$ can be arbitrary (not necessarily efficient) functions and thus unknown-regular one-way functions fall into a special case[9] for $c = 0$.

5.2 UOWHFs from Beyond Almost-Regular OWFs

We state below the main results of this section, namely, the fourth construction which is based on weakly unknown-regular one-way functions (see Definition 5.1).

Theorem 5.1 *Assume that f is a weakly unknown-regular one-way function on an n^{-c}-fraction of domain for constant c. Then, there exists an explicit construction of UOWHF family (stated as Construction 1 in Appendix B.4) with output length $\Theta(n)$, key length $O(n \cdot \mathrm{log} n)$ by making $n^{2c+1} \cdot \omega(1)$ black-box calls to f.*

The main idea is to transform any weakly unknown-regular one-way function f into a family of functions $\mathcal{F} = \{f_u : u \in \{0,1\}^{O(n \log n)}\}$ such that $\mathcal{F}$ is almost regular and that it preserves the one-way-ness of f. $\mathcal{F}$ is constructed based on (the derandomized version of) the randomized iterate with a succinct description u. Finally, we sample a random $f_u \xleftarrow{\$} \mathcal{F}$ and plug it into the construction by Ames et al. (see Theorem B.1) to get the UOWHFs as desired.

Definition 5.2 (the randomized iterate [10, 7]) *Let $n \in \mathbb{N}$, function $f : \{0,1\}^n \to \{0,1\}^n$, and let $\mathcal{H}$ be a family of pairwise-independent length-preserving hash functions over $\{0,1\}^n$. For $k \in \mathbb{N}$, $x_1 \in \{0,1\}^n$ and vector $\vec{h}^k = (h_1, \ldots, h_k) \in \mathcal{H}^k$, recursively define the i^{th} randomized iterate by:*

$$x_1 \xrightarrow{f} y_1 \xrightarrow{h_1} x_2 \xrightarrow{f} y_2 \xrightarrow{h_2} \ \cdots \ x_k \xrightarrow{f} y_k \xrightarrow{h_k}$$

$$y_i = f(x_i), \ x_{i+1} = h_i(y_i) \ .$$

We denote the i^{th} iterate by function f^i, i.e., $y_i = f^i(x_1, \vec{h}^k)$, where $\vec{h}^k$ is possibly redundant as for $i \le k+1$ y_i only depends on $\vec{h}^{i-1}$.

*The **randomized version** refers to the case where $x_1 \xleftarrow{\$} \{0,1\}^n$ and $\vec{h}^k \xleftarrow{\$} \mathcal{H}^k$.*

*The **derandomized version** refers to that $x_1 \xleftarrow{\$} \{0,1\}^n$, $u \xleftarrow{\$} \{0,1\}^{q \in O(n \cdot \mathrm{log} n)}$, $\vec{h}^k := BSG(u)$, where $BSG : \{0,1\}^q \to \{0,1\}^{k \cdot \log|\mathcal{H}|}$ is a bounded-space generator that 2^{-2n}-fools every $(2n+1, k, \log|\mathcal{H}|)$-LBP (see Definition B.2), and $\log|\mathcal{H}|$ is the description length of $\mathcal{H}$ (e.g., $2n$ bits for concreteness).*

We refer to Definition B.2 for the definition of bounded-width layered branching program (LBP). Note that the aforementioned bounded-space generators exist due to Theorem B.2 (see Appendix B.2) by setting $s = 2n+1$, $v = \log|\mathcal{H}| = \Theta(n)$, $k = \mathsf{poly}(n)$ and $\varepsilon = 2^{-2n}$ and thus $q{=}O(n \cdot \log n)$.

[9] In fact, our construction #4 only assumes a relaxed condition than (1), i.e., $\Pr[U_n \in \mathcal{X}_{\max - O(\log n)} \cup \ldots \cup \mathcal{X}_{\max}] \ge n^{-c}$, so that unknown-almost-regular one-way functions become a special case for $c = 0$.

Remark 5.1 (on what is proven in [24]) *The authors of [24] introduced weakly unknown-regular one-way functions from which they constructed a pseudorandom generator with seed length $O(n \cdot \log n)$ based on the randomized iterate. They showed that "every $k = n^{2c} \cdot \log n \cdot \omega(1)$ iterations are hard-to-invert", i.e., for any j it is hard to predict x_j given $y_{j+k} = f^{j+k}(x_1, BSG(u))$ and u. A PRG thus follows by outputting $\log n$ hardcore bits for every k iterations. In this paper, we first adapt their findings to show that $f_u(\cdot) = f^k(\cdot, BSG(u))$ constitutes a family of one-way functions, i.e., given $y_k = f_u(x_1)$ and u it is infeasible to find any x_1' such that $y_k = f^k(x_1', BSG(u))$. This is stated as Lemma 5.1 with proof given in Appendix B.2. However, it is still insufficient to construct UOWHFs with the one-way-ness of f_u. We further show in Lemma 5.2 that a random $f_u \xleftarrow{\$} \mathcal{F}$ is almost regular (in a slightly weaker sense than Definition 2.6 but already suffices for our needs).*

Following [24], we define the following events. Some inequalities about these events from [24] are stated as Lemma B.2 (along with proof reproduced) in Appendix B.2.

Definition 5.3 (events) *For any $n, j{\le}k \in \mathbb{N}$, define events*

$$\mathcal{E}_j \stackrel{\mathsf{def}}{=} \left((X_1, \vec{H}^k) \in \{ \ (x_1, \vec{h}^k) : y_j = f^j(x_1, \vec{h}^k) \in \mathcal{Y}_{\max} \ \} \right)$$

$$\mathcal{E}_j' \stackrel{\mathsf{def}}{=} \left((X_1, U_q) \in \{ \ (x_1, u) : y_j = f^j(x_1, BSG(u)) \in \mathcal{Y}_{\max} \ \} \right)$$

where $\mathcal{Y}_{\max} \stackrel{def}{=} \{y : 2^{\max -1} {\le} |f^{-1}(y)| < 2^{\max}\}$, $(X_1, \vec{H}^k)$ and (X_1, U_q) are uniform over $\{0,1\}^n \times \mathcal{H}^k$ and $\{0,1\}^n \times \{0,1\}^q$ respectively. Note that by definition $\mathcal{Y}_{\max} = f(\mathcal{X}_{\max})$ (see Definition 5.1) and thus $\Pr[f(U_n) \in \mathcal{Y}_{\max}] \ge n^{-c}$.

Lemma 5.1 ($\mathcal{F}$ is one-way) *Assume that f is a (t, ε)-OWF that is weakly unknown-regular on an n^{-c} fraction of domain, define a family of functions*

$$\mathcal{F} \stackrel{def}{=} \{ \ f_u : \{0,1\}^n \to \{0,1\}^n, f_u(x){=}f^k(x, BSG(u)), u \in \{0,1\}^{O(n \cdot \log n)} \ \} \tag{2}$$

where $\mathcal{H}$,f^k and $BSG : \{0,1\}^{q \in O(n \cdot \log n)} \to \{0,1\}^{k \cdot \log|\mathcal{H}|}$ are as defined in Definition 5.2. Then, for any A of running time $t - n^{O(1)}$ it holds that

$$\Pr_{u \xleftarrow{\$} \{0,1\}^q, \ x \xleftarrow{\$} \{0,1\}^n} [\ \mathsf{A}(u, f_u(x)) \in f_u^{-1}(\ f_u(x) \ ) \ ] \ \le \ \sqrt{2^8 \cdot k^4 \cdot n^{3c} \cdot \varepsilon} + 2^{-k/n^{2c}} \ + \ 2^{-2n} \ . \tag{3}$$

Lemma 5.1 is mainly attributed to and adapted from [24] (see Remark 5.1). We refer to Appendix B.2 for the adapted proof.

Lemma 5.2 ($\mathcal{F}$ is almost-regular) *Let $\mathcal{F} = \{f_u\}$ be as defined in Lemma 5.1. Then, for any $a \ge 0$ it holds that*

$$\Pr_{u \xleftarrow{\$} \{0,1\}^q, \ x \xleftarrow{\$} \{0,1\}^n} [\ 2^{\max - a - 1} \ \le \ |f_u^{-1} \ (\ f_u(x) \ )| \ \le \ 2^{\max + a + 1} \ ] \ \ge \ 1 \ - \ \frac{k}{2^{a-2}} \ - \ \frac{1}{2^{k/n^{2c}}} \ - \ \frac{1}{2^{2n}}, \tag{4}$$

where $u \in \{0,1\}^{q \in O(n \cdot \log n)}$ and $f_u(x){=}f^k(x, BSG(u))$.

Proof. We define $\mathcal{S}_{low} \stackrel{\mathsf{def}}{=} \left((X_1, U_q) \in \{(x, u) : 0 < |f_u^{-1}(f_u(x))| < 2^{\max - a - 1}\} \right)$ and $\mathcal{S}_{up} \stackrel{\mathsf{def}}{=} \Big((X_1, U_q) \in \{(x, u) : |f_u^{-1}(f_u(x))| > 2^{\max + a + 1}\} \Big)$, where X_1 is uniform over $\{0,1\}^n$. The left-hand of (4) is lower bounded by $1 - \Pr[\mathcal{S}_{low}] - \Pr[\mathcal{S}_{up}]$ and thus it suffices to upper bound both $\Pr[\mathcal{S}_{low}]$ and $\Pr[\mathcal{S}_{up}]$. We have

$$\begin{aligned}
\Pr[\mathcal{S}_{low}] &= \Pr[\mathcal{S}_{low} \wedge (\mathcal{E}_1' \vee \mathcal{E}_2' \vee \ldots \vee \mathcal{E}_k')] \;+\; \Pr[\mathcal{S}_{low} \wedge \neg(\mathcal{E}_1' \vee \mathcal{E}_2' \vee \ldots \vee \mathcal{E}_k')] \\
&\leq \Pr[\bigvee_{j=1}^{k} (\mathcal{S}_{low} \wedge \mathcal{E}_j')] \;+\; \Pr[\neg(\mathcal{E}_1' \vee \mathcal{E}_2' \vee \ldots \vee \mathcal{E}_k')] \\
&\leq \sum_{j=1}^{k} \Pr[\mathcal{S}_{low} \wedge \mathcal{E}_j'] \;+\; (2^{-k/n^{2c}} \;+\; 2^{-2n}) \\
&\leq k{\cdot}2^{-a} \;+\; 2^{-k/n^{2c}} \;+\; 2^{-2n}
\end{aligned}$$

where the first inequality is trivial, the second is by the union bound and (12), and the third is due to that for every $j \in [k]$ with shorthand $f_{u,j}(x) \stackrel{\text{def}}{=} f^j(x, BSG(u))$ it holds that

$$\begin{aligned}
\Pr[\mathcal{S}_{low} \wedge \mathcal{E}_j'] \;&=\; \sum_u \Pr[U_q = u] \cdot \sum_{x:\ f_{u,j}(x) \in \mathcal{Y}_{\max}\ \wedge\ 0 < |f_u^{-1}(f_u(x))| < 2^{\max - a - 1}} \Pr[X_1 = x | U_q = u] \\
&\leq \sum_u \Pr[U_q = u] \cdot \sum_{x:\ f_{u,j}(x) \in \mathcal{Y}_{\max}\ \wedge\ 0 < |f_{u,j}^{-1}(f_{u,j}(x))| < 2^{\max - a - 1}} \Pr[X_1 = x \mid U_q = u] \\
&\leq \sum_u \Pr[U_q = u] \cdot |\mathcal{Y}_{\max}| {\cdot} 2^{\max - a - 1} {\cdot} 2^{-n} \\
&\leq 2^{n+1-\max} \cdot 2^{-n+\max - a - 1} \;=\; 2^{-a}
\end{aligned}$$

where the first inequality is due to Fact 3 (setting $f_1 {=} f_{u,j}$, $f_2 = f \circ h_{k-1} \circ \ldots \circ f \circ h_j$ and thus $\bar{f} = f_u$), the second follows from the fact that there are $|\mathcal{Y}_{\max}|$ possible values for $f_{u,j}(x) \in \mathcal{Y}_{\max}$ and every $f_{u,j}(x)$ has less than $2^{\max - a - 1}$ preimages (by definition of $\mathcal{S}_{low}$), and the third is due to $|\mathcal{Y}_{\max}| {\leq} 2^{n+1-\max}$. Next we proceed to bounding the second term, i.e., $\Pr[\mathcal{S}_{up}] \;\leq\; k{\cdot}2^{-a+1}$.

$$\begin{aligned}
k{\cdot}2^{\max - n + 1} + 2^{-2n} \quad &\geq \quad \mathsf{CP}(\ Y_k' \mid U_q) \;= \mathbb{E}_{u \leftarrow U_q}\Big[\sum_y \Pr[\ f_u(X_1) = y \mid U_q = u]^2 \Big] \\
&> \quad 2^{\max + a - n + 1} \cdot \mathbb{E}_{u \leftarrow U_q}\Big[\sum_{y:\ |f_u^{-1}(y)| > 2^{\max + a + 1}} \Pr[\ f_u(X_1) = y \mid U_q = u] \Big] \\
&= \quad 2^{\max + a - n + 1} \cdot \Pr[\mathcal{S}_{up}] \ ,
\end{aligned}$$

where the first inequality is by (8), and the second is due to that for any (y, u) satisfying $|f_u^{-1}(y)| > 2^{\max + a + 1}$ and it holds that

$$\Pr[\ f_u(X_1) = y \mid U_q = u] \;=\; \Pr[\ X_1 \in f_u^{-1}(y)\ ] \;> 2^{-n} {\cdot} 2^{\max + a + 1} = 2^{\max + a - n + 1} \ .$$

It follows that $\Pr[\mathcal{S}_{up}] \leq (k{\cdot}2^{\max - n + 1} + 2^{-2n}) / 2^{\max + a - n + 1} {\leq} 2^{-a+1}$ and hence completes the proof. □

Fact 3 *Let $f_1 : \mathcal{X} \to \mathcal{Y}$ and $f_2 : \mathcal{Y} \to \mathcal{Z}$ be any functions, and let $\bar{f} \stackrel{def}{=} f_2 \circ f_1$. Then for any $t \in \mathbb{N}^+$ it holds that*

$$\{x : 0 < |\bar{f}^{-1}(\bar{f}(x))| < t\} \;\subseteq\; \{x : 0 < |f_1^{-1}(f_1(x))| < t\} \ .$$

Proof. Any x satisfying $0 < |\bar{f}^{-1}(\bar{f}(x))| < t$ implies $0 < |f_1^{-1}(f_1(x))| < t$. □

Given that $\mathcal{F}$ is a family of unknown-(almost-)regular one-way functions with description length $O(n \cdot \log n)$, we just plug a random $f_u \in \mathcal{F}$ into the Ames et al.'s construction [1] to yield a family of UOWHFs with output length $\Theta(n)$ and key length $O(n \cdot \log n)$. We refer to Appendix B.4 for Construction 1 and the proof Theorem 5.1, where we put together all the necessary technical details.

References

[1] Scott Ames, Rosario Gennaro, and Muthuramakrishnan Venkitasubramaniam. The generalized randomized iterate and its application to new efficient constructions of UOWHFs from regular one-way functions. In *ASIACRYPT*, pages 154–171, 2012.

[2] Kfir Barhum and Thomas Holenstein. A cookbook for black-box separations and a recipe for uowhfs. In *Proceedings of the 5th Theory of Cryptography Conference (TCC 2013)*, pages 662–679, 2013.

[3] Kfir Barhum and Ueli Maurer. UOWHFs from OWFs: Trading regularity for efficiency. In *LATINCRYPT*, pages 234–253, 2012.

[4] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[5] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

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

[7] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. *SIAM Journal on Computing*, 22(6):1163–1175, 1993.

[8] Oded Goldreich, Leonid A. Levin, and Noam Nisan. On constructing 1-1 one-way functions. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 13–25. 2011.

[9] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[10] Iftach Haitner, Danny Harnik, and Omer Reingold. On the power of the randomized iterate. In *Proceedings of the 26th International Cryptology Conference (CRYPTO 2006)*, pages 22–40, 2006.

[11] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *EUROCRYPT*, pages 616–637, 2010.

[12] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM Journal on Computing*, 39(3):1153–1218, 2009.

[13] Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Inaccessible entropy. In *Proceedings of the 41st ACM Symposium on the Theory of Computing*, pages 611–620, 2009.

[14] Thomas Holenstein and Makrand Sinha. Constructing a pseudorandom generator requires an almost linear Number of calls. In *Proceedings of the 53rd IEEE Symposium on Foundation of Computer Science*, pages 698–707, 2012.

[15] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pages 356–364, 1994.

[16] Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. *IACR Cryptology ePrint Archive*, 2005. http://eprint.iacr.org/2005/328.

[17] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In D. S. Johnson, editor, *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 33–43, Seattle, Washington, 15–17 May 1989.

[18] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[19] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 387–394, Baltimore, Maryland, 14–16 May 1990.

[20] John Rompel. *Techniques for computing with low-independence randomness.* PhD thesis, Massachusetts Institute of Technology, 1990. http://dspace.mit.edu/handle/1721.1/7582.

[21] Alfredo De Santis and Moti Yung. On the design of provably secure cryptographic hash functions. In *EUROCRYPT*, pages 412–431, 1990.

[22] Victor Shoup. A composition theorem for universal one-way hash functions. In Bart Preneel, editor, *Advances in Cryptology—EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 445–452. Springer-Verlag, 2000.

[23] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *Proceedings of the 23rd IEEE Symposium on Foundation of Computer Science*, pages 80–91, 1982.

[24] Yu Yu, Dawu Gu, Xiangxue Li, and Jian Weng. The randomized iterate revisited - almost linear seed length PRGs from a broader class of one-way functions. In *12th Theory of Cryptography Conference (TCC 2015)*, pages 7–35, 2015.

[25] Yu Yu, Xiangxue Li, and Jian Weng. Pseudorandom generators from regular one-way functions: New constructions with improved parameters. In *ASIACRYPT*, pages 261–279, 2013.

A Lemmata and Proofs Omitted

A.1 The Hash Lemma and Output Length Simplifications

Proof of Lemma 3.1.

$$\begin{aligned}
&\Pr_{y\leftarrow Y,\ h\xleftarrow{\$}\mathcal{H}}[\ \exists\tilde{y}\in\mathsf{Supp}(Y):\ \tilde{y}\neq y\wedge h(\tilde{y})=h(y)\ ]\\
\leq\ &\max_{y}\{\Pr_{h\xleftarrow{\$}\mathcal{H}}[\ \exists\tilde{y}\in\mathsf{Supp}(Y):\ \tilde{y}\neq y\wedge h(\tilde{y})=h(y)\ ]\ \}\\
\leq\ &\max_{y}\{\sum_{\tilde{y}\in\mathsf{Supp}(Y)\setminus\{y\}}\Pr_{h\xleftarrow{\$}\mathcal{H}}[\ h(\tilde{y})=h(y)\ ]\ \}\\
\leq\ &|\mathsf{Supp}(Y)|\cdot k\cdot 2^{-(a+d)}\ \leq\ k\cdot 2^{-d}\ ,
\end{aligned}$$

where the second inequality is a union bound, the third inequality follows by the universality of $\mathcal{H}$ and the last one is due to $\mathbf{H}_0(Y)\leq a$. ☐

It is folklore that almost universal hash functions can be efficiently constructed. See for example the following construction.

Fact 4 (efficient constructions of almost universal hashing) *For any integers $t\leq l$, there exists a family of $O(l/t)\cdot 2^{-t}$-almost universal hash functions $\mathcal{H}=\{h:\{0,1\}^l\rightarrow\{0,1\}^t\}$ such that $\mathcal{H}$ has description length $O(t)$ and every $h\in\mathcal{H}$ is computable in time $\mathsf{poly}(l)$.*

A CONCRETE EXAMPLE. Assume without loss of generality that t divides l , i.e., $l = k{\cdot}t$ for some $k \in \mathbb{N}$ (otherwise use $l' = \lceil (l/t) \rceil \cdot t$ instead of l), and parse x as a sequence of t-bit strings $(x_1, \ldots, x_k)$. Then, we have that $\mathcal{H} = \{h_a : h_a(x) \stackrel{\text{def}}{=} \sum_{i=1}^{k} a^i{\cdot}x_i,\ a, x_i \in GF(2^t)\}$ is a family of $k{\cdot}2^{-t}$-almost universal hash functions of description length t.

Lemma A.1 (regularity-preserving OWF) *For any $r_1 = r_1(n) \leq r_2 = r_2(n)$, and any efficiently computable $\kappa = \kappa(n) \in O(n)$, let $f : \{0,1\}^n \to \{0,1\}^l$ be any $(2^{r_1}, 2^{r_2})$-almost regular (t,ε)-one-way function, let $\mathcal{H} = \{h : \{0,1\}^l \to \{0,1\}^{n+\kappa}\}$ be a family of $(\mathsf{poly}(n){\cdot}2^{-(n+\kappa)})$-almost universal hash functions with description length*[10] *$O(n)$, define function $f' : \{0,1\}^n \times \mathcal{H} \to \{0,1\}^{n+\kappa} \times \mathcal{H}$ as*

$$f'(x, h) = (h(f(x)), h) \quad . \tag{5}$$

Then, we have

1. REGULARITY-PRESERVING. *f' is $(2^{r_1}, 2^{r_2})$-regular except on a $\mathsf{poly}(n){\cdot}2^{-(r_1+\kappa)}$-fraction of inputs, i.e.,*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n,\ h \xleftarrow{\$} \mathcal{H}} [\ 2^{r_1} \ \leq\ |f'^{-1}(f'(x, h))| \leq 2^{r_2}\ ]\ \geq\ 1 - \mathsf{poly}(n){\cdot}2^{-(r_1+\kappa)} \quad .$$

2. HARDNESS-PRESERVING. *f' is a $(t - n^{O(1)},\ \varepsilon + \mathsf{poly}(n){\cdot}2^{-(r_1+\kappa)})$-one-way function.*

Proof of Lemma A.1. As for every $y = f(x)$ we have $2^{r_1} {\leq} |f^{-1}(y)| {\leq} 2^{r_2}$ it suffices to show that the fraction of y's (drawn from $Y = f(U_n)$) on which h is 1-to-1 is overwhelming, i.e.,

$$\begin{aligned}
&\Pr_{x \xleftarrow{\$} \{0,1\}^n,\ h \xleftarrow{\$} \mathcal{H}} [\ 2^{r_1} \ \leq\ |f'^{-1}(f'(x, h))| \ \leq\ 2^{r_2}\ ] \\
\geq\ & \Pr_{y \leftarrow f(U_n),\ h \xleftarrow{\$} \mathcal{H}} [\ \neg\exists \tilde{y}{\in} f(\{0,1\}^n) : h(\tilde{y}) = h(y)\ \wedge\ y \neq \tilde{y}\ ] \\
=\ & 1\ -\ \Pr_{y \leftarrow f(U_n),\ h \xleftarrow{\$} \mathcal{H}} [\ \exists \tilde{y}{\in} f(\{0,1\}^n) : h(\tilde{y}) = h(y)\ \wedge\ y \neq \tilde{y}\ ] \\
\geq\ & 1 - \mathsf{poly}(n){\cdot}2^{-(r_1+\kappa)}\ ,
\end{aligned}$$

where the second inequality is by Lemma 3.1 (setting $Y = f(U_n)$, $a {\leq} n - r_1$, $d = r_1 + \kappa$, $k = \mathsf{poly}(n)$). Further, it is not hard to see that any f'-inverting algorithm $\mathsf{A}_{f'}$ implies an f-inverting algorithm A_f. That is, on input $f(x)$, A_f applies random h to $f(x)$, and then invokes $\mathsf{A}_{f'}$ on $(h(f(x)), h)$ to recover x. The inversion probability of A_f is

$$\begin{aligned}
&\Pr_{y \leftarrow f(U_n)} [\mathsf{A}_f(y) \in f^{-1}(y)] \\
\geq\ & \Pr_{y \leftarrow f(U_n), h \xleftarrow{\$} \mathcal{H}} [\mathsf{A}_{f'}(h(y), h) \in f'^{-1}(h(y), h) \wedge \neg \Big(\exists \tilde{y}{\in} f(\{0,1\}^n) : h(\tilde{y}) = h(y)\ \wedge\ y \neq \tilde{y} \Big)] \\
=\ & 1 - \Pr_{y \leftarrow f(U_n), h \xleftarrow{\$} \mathcal{H}} [\mathsf{A}_{f'}(h(y), h) \notin f'^{-1}(h(y), h) \vee \Big(\exists \tilde{y}{\in} f(\{0,1\}^n) : h(\tilde{y}) = h(y) {\wedge} y \neq \tilde{y} \Big)] \\
\geq\ & 1 - \Pr_{y \leftarrow f(U_n), h \xleftarrow{\$} \mathcal{H}} [\mathsf{A}_{f'}(h(y), h) \notin f'^{-1}(h(y), h)] - \Pr_{y \leftarrow f(U_n), h \xleftarrow{\$} \mathcal{H}} [\exists \tilde{y}{\in} f(\{0,1\}^n) : h(\tilde{y}) = h(y) {\wedge} y \neq \tilde{y}\ ] \\
\geq\ & \Pr_{y \leftarrow f(U_n), h \xleftarrow{\$} \mathcal{H}} [\mathsf{A}_{f'}(h(y), h) \in f'^{-1}(h(y), h)]\ -\ \mathsf{poly}(n){\cdot}2^{-(r_1+\kappa)}\ ,
\end{aligned}$$

[10] Such efficient $\mathcal{H}$ exists for any efficiently computable $l \in \mathsf{poly}(n)$ and $\kappa \in O(n)$ by Fact 4.

where the first inequality refers to that A_f inverts f if $\mathsf{A}_{f'}$ inverts f' on those $(h(y), h)$ for which there exists no $\tilde{y} \neq y$ satisfying $h(\tilde{y}) = h(y)$, the second inequality is the union bound, and the third is due to the probability that h is not injective on Y as given above. This completes the proof. □

Proof of Fact 1. The first statement immediately follows from Lemma A.1 by setting $r_1 = r_2 = 0$. As for the second statement, let f' be as defined in (5) from Lemma A.1, we further define a padded function $\bar{f} : \{0,1\}^{n+\kappa} \times \mathcal{H} \to \{0,1\}^{n+\kappa} \times \mathcal{H}$ as

$$\bar{f}(x, \mathsf{dummy}, h) \stackrel{\text{def}}{=} f'(x, h) \ ,$$

where $x \in \{0,1\}^n$, $\mathsf{dummy} \in \{0,1\}^\kappa$, and $h \in \mathcal{H}$ (which is of size $O(n)$). Note that for every $(h(f(x)), h)$ the preimage-size of $\bar{f}$ is multiplied by a factor of 2^κ than that of f' due to the κ-bit padding dummy. This concludes the second statement. □

A.2 Proof for Lemma 4.1

Fact 5 *For any $(2^r, 2^r n^c)$-almost regular (length-preserving) $(t,\ \varepsilon)$-one-way function $f : \{0,1\}^n \to \{0,1\}^n$ and any inverter Inv of running t, it holds that*

$$\Pr_{y^* \xleftarrow{\$} \{0,1\}^l} [\ f(\mathsf{Inv}(y^*)) = y^*\ ] \ \leq\ 2^{-r} \cdot \varepsilon \quad .$$

Proof.

$$\begin{aligned}
&\Pr_{y^* \xleftarrow{\$} \{0,1\}^n} [\ f(\mathsf{Inv}(y^*)) = y^*\ ] \\
&= \sum_{y \in f(\{0,1\}^n)} \Pr_{y^* \xleftarrow{\$} \{0,1\}^n} [y = y^*] \cdot \Pr[f(\mathsf{Inv}(y)) = y] \\
&\leq\ 2^{-r} \cdot \sum_{y \in f(\{0,1\}^n)} \Pr[f(U_n) = y] \cdot \Pr[f(\mathsf{Inv}(y)) = y] \\
&\leq\ 2^{-r} \cdot \varepsilon
\end{aligned}$$

where the first inequality is due to $\Pr[f(U_n) = y] \geq 2^{r-n}$ and the second is due to f's one-way-ness. □

Proof of Lemma 4.1. Suppose for contradiction that there exists an efficient $\tilde{\mathsf{A}}$ of running time t' such that

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n, h \xleftarrow{\$} \mathcal{H}} [\ x' \leftarrow \tilde{\mathsf{A}}(x, h) :\ f(x) \neq f(x') \wedge h(f(x))_{[u]} = h(f(x'))_{[u]}\ ] \ =\ \varepsilon' \ >\ n^c \cdot 2^{s'+1} \cdot \varepsilon$$

where $u = n - r - c\log n - s'$. We proceed to the definition of algorithm $\mathsf{Inv}^{\tilde{\mathsf{A}}}$ (that inverts f by invoking $\tilde{\mathsf{A}}$) as in Algorithm 2. By Claim 1, conditioned on $f(x) {\neq} y^*$ it is equivalent to consider that $\mathsf{Inv}^{\tilde{\mathsf{A}}}$ samples (x,h,v) from $\{0,1\}^n \times \mathcal{H} \times \mathcal{V}$ uniformly and independently, from which y^* can be determined. Then, we argue that $\mathsf{Inv}^{\tilde{\mathsf{A}}}$ inverts f with the following probability (see the rationale below)

$$\begin{aligned}
&\Pr_{y^* \xleftarrow{\$} \{0,1\}^n,\ x \xleftarrow{\$} \{0,1\}^n,\ v \xleftarrow{\$} \mathcal{V}} [\ f(\mathsf{Inv}^{\tilde{\mathsf{A}}}(y^*)) = y^*\ ] \\
\geq\ & \Pr_{y^* \xleftarrow{\$} \{0,1\}^n,\ x \xleftarrow{\$} \{0,1\}^n} [\ \mathcal{E}_{\mathsf{neq}}\ ] \cdot \Pr_{y^* \xleftarrow{\$} \{0,1\}^n \setminus \{f(x)\},\ x \xleftarrow{\$} \{0,1\}^n,\ v \xleftarrow{\$} \mathcal{V}} [\ f(\mathsf{Inv}^{\tilde{\mathsf{A}}}(y^*)) = y^* \mid \mathcal{E}_{\mathsf{neq}}\ ] \\
\geq\ & (1 - 2^{-n}) \cdot \Pr_{x \xleftarrow{\$} \{0,1\}^n, h \xleftarrow{\$} \mathcal{H}, x' \leftarrow \tilde{\mathsf{A}}(x,h), v \xleftarrow{\$} \mathcal{V}} [\ f(x) {\neq} f(x')\ \wedge\ h(f(x))_{[u]} = h(f(x'))_{[u]}\ \wedge\ y^* = f(x')\ ] \\
\geq\ & (1 - 2^{-n})\varepsilon' \cdot \Pr_{v \xleftarrow{\$} \mathcal{V}} [\ y^* = f(x') \mid \mathcal{E}_{\mathsf{neq}}\ \wedge\ f(x) \neq f(x')\ \wedge\ h(f(x))_{[u]} = h(f(x'))_{[u]}\ ] \\
=\ & (1 - 2^{-n})\varepsilon' \cdot \frac{1}{|\mathcal{V}|}\ >\ \frac{\varepsilon'}{2} \cdot \frac{1}{2^{r + c\log n + s'} - 1}\ >\ \frac{n^c \cdot 2^{s'+1} \cdot \varepsilon}{2^{r + c\log n + s' + 1}}\ =\ \varepsilon \cdot 2^{-r} \quad ,
\end{aligned}$$

Algorithm 2 $\mathsf{Inv}^{\tilde{\mathsf{A}}}$ that inverts f on input y^* using random coins (x, v).

Input: $y^* \xleftarrow{\$} \{0,1\}^n$

Sample $x \xleftarrow{\$} \{0,1\}^n$
if $f(x) = y^*$ **then**
  Output x and **terminate**.
end if
sample $h := (f(x) - y^*)^{-1} \cdot v$, where $v \xleftarrow{\$} \mathcal{V} = \{v \in \{0,1\}^n \setminus \{\vec{0}\} : v_{[u]} = \overbrace{0\ldots0}^{u}\}$
$\{$*note*: The above implies $h \xleftarrow{\$} \{h \in \mathcal{H} : \ h(f(x))_{[u]} = h(y^*)_{[u]}\}$ by the algebraic structure of h. $\}$
$x' \leftarrow \tilde{\mathsf{A}}(x, h)$
if $f(x')$=y^* **then**
  Output x'
else
  Output $\perp$
end if
Terminate

where the first inequality is straightforward (note that conditioned on $\mathcal{E}_{\mathsf{neq}}$ the sampling of y^* and x are independent and uniform over $\{0,1\}^n \setminus \{f(x)\}$ and $\{0,1\}^n$ respectively), the second inequality follows from Claim 1, namely, conditioned on $\mathcal{E}_{\mathsf{neq}}$ it is equivalent to consider $(x,\, h,\, v) \xleftarrow{\$} \{0,1\}^n \times \mathcal{H} \times \mathcal{V}$ and then $y^* := f(x) - v \cdot h^{-1}$, and the third inequality is due to that $\tilde{\mathsf{A}}$ takes only x and h as input (i.e., independent of v). That is, given that $\tilde{\mathsf{A}}$ produces a valid $f(x') \neq f(x)$ satisfying $h(f(x'))_{[n-s]} = h(f(x))_{[n-s]}$, we have by Claim 1 that string y^* is uniformly distributed over set $\mathcal{Y}^* \stackrel{\mathsf{def}}{=} \{f(x) - v \cdot h^{-1}, v \in \mathcal{V}\}$. Note that the already fixed $f(x')$ is also an element of $\mathcal{Y}^*$ and thus y^* hits $f(x')$ with probability $1/|\mathcal{Y}^*|$=$1/|\mathcal{V}| = 1/(2^{n-u} - 1)$. We thus complete the proof by reaching a contradiction to Fact 5. □

A.3 The Equivalence Between 1-to-1 and Known-Almost-Regular One-way Functions

Proof of Corollary 3.1. Recall the construction of UOWHF as below

$$\mathcal{G}_1 \stackrel{\mathsf{def}}{=} \{\ (\mathsf{trunc} \circ h \circ f\ ) : \{0,1\}^n \to \{0,1\}^{n-s}\ ,\ h \in \mathcal{H}\ \}\ ,$$

where h and trunc are defined as in Theorem 3.1 but now f is 1-to-1 only on a $(1-\delta)$-fraction. In fact, we already prove (by checking the proof of Theorem 3.1) a more general version than Theorem 3.1: for any (t,ε)-one-way function f and any A of running time $t - n^{O(1)}$ it holds that

$$\begin{aligned} &\Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)}[\ f(x) \neq f(x') \wedge g_1(x) = g_1(x')\ ] \\ &\leq\ 2^{l-n+s+1} \cdot \max_{time(\mathsf{Inv})\ \leq\ t} \left\{ \Pr_{y^* \xleftarrow{\$} \{0,1\}^l}[\ f(\mathsf{Inv}(y^*)) = y^*\ ] \right\}\ , \end{aligned}$$

where the maximum is taken over all Inv of running time t. Hence, the statement of Theorem 3.1 follows from Fact 2 and that for 1-to-1 f "$f(x) \neq f(x')$" is equivalent to "$x \neq x'$". Now that f is not strictly 1-to-1, we need to adapt the proofs. First we observe that Fact 2 holds for any (t,ε)-one-way function

f, i.e.,

$$
\begin{aligned}
& \Pr_{y^* \xleftarrow{\$} \{0,1\}^l} [\ f(\mathsf{Inv}(y^*)) = y^*\ ] \\
= \ & \sum_{y^* \in \{0,1\}^l} 2^{-l} \cdot \Pr[\ f(\mathsf{Inv}(y^*)) = y^*\ ] \\
= \ & \sum_{y^* \in f(\{0,1\}^n)} 2^{-l} \cdot \Pr[\ f(\mathsf{Inv}(y^*)) = y^*\ ] \\
\leq \ & 2^{n-l} \cdot \sum_{y^* \in f(\{0,1\}^n)} \Pr[f(U_n) = y^*] \cdot \Pr[\ f(\mathsf{Inv}(y^*)) = y^*\ ] \\
\leq \ & 2^{n-l} \cdot \varepsilon \ ,
\end{aligned}
$$

where the first inequality is due to for any $y^* \in f(\{0,1\}^n)$ it holds that $\Pr[f(U_n) = y^*] \geq 2^{-n}$ and the second inequality is due to the one-way-ness of f. Therefore, we have

$$
\Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)} [\ f(x) \neq f(x') \wedge g_1(x) = g_1(x')\ ] \ \leq \ 2^{s+1}\varepsilon \ .
$$

Denote by $\mathcal{X}_1 \stackrel{\text{def}}{=} \{x : |f^{-1}(f(x))| = 1\}$ and thus

$$
\begin{aligned}
& \Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)} [\ x \neq x' \wedge g_1(x) = g_1(x')\ ] \\
\leq \ & \Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)} [\ x \in \mathcal{X}_1 \ \wedge \ x \neq x' \wedge g_1(x) = g_1(x')\ ] \ + \Pr_{x \xleftarrow{\$} \{0,1\}^n} [x \notin \mathcal{X}_1] \\
\leq \ & \Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)} [\ x \in \mathcal{X}_1 \ \wedge \ f(x) \neq f(x') \wedge g_1(x) = g_1(x')\ ] \ + \ \delta \\
\leq \ & \Pr_{x \xleftarrow{\$} \{0,1\}^n, g_1 \xleftarrow{\$} \mathcal{G}_1, x' \leftarrow \mathsf{A}(x,h)} [\ f(x) \neq f(x') \wedge g_1(x) = g_1(x')\ ] \ + \ \delta \\
\leq \ & 2^{s+1}\varepsilon \ + \ \delta \ ,
\end{aligned}
$$

where the second inequality follows from the fact that for any $x \in \mathcal{X}_1$ the condition "$x \neq x'$" implies "$f(x) \neq f(x')$" (by considering both cases $x' \in \mathcal{X}_1$ and $x' \notin \mathcal{X}_1$). This completes the proof. $\square$

Proof of Lemma 4.2.

$$
\begin{aligned}
& \Pr_{x \xleftarrow{\$} \{0,1\}^n, \ h_1 \xleftarrow{\$} \mathcal{H}_1} [\ \exists x' : x' \neq x \ \wedge \ f'(x, h_1) = f'(x', h_1)] \\
= \ & \mathbb{E}_{y \leftarrow f(U_n)} \left[\Pr_{x \xleftarrow{\$} f^{-1}(y), \ h_1 \xleftarrow{\$} \mathcal{H}_1} [\exists x' \in f^{-1}(y) : x' \neq x \ \wedge \ h_1(x) = h_1(x')] \right] \\
\leq \ & \mathbb{E}_{y \leftarrow f(U_n)} [\ 2^{-d}\ ] \ = \ 2^{-d} \ ,
\end{aligned}
$$

where the inequality is due to that conditioned on any $y = f(X)$ random variable X is a flat distribution on a set of size at most $2^r \cdot n^c$, so we apply Lemma 3.1 (setting $a \leq r + c \cdot \log n$, and $k = 1$). $\square$

Proof of Lemma 4.3. Suppose that there exists some inverter A' for f' of running time $t - O(n)$ such that

$$
\Pr[\ \mathsf{A}'(f'(X, H_1)) \in \ f'^{-1} \ (\ f'(X, H_1)\ )\ ] \ > \ \sqrt{2^{d+3} \cdot n^c \cdot \varepsilon} \ .
$$

First we consider the collision probability of $H_1(X)$ given $f(X)$ and H_1, i.e.,

$$\begin{aligned}
& \mathsf{CP}(\ H_1(X) \mid f(X), H_1\ ) \\
= \ & \mathbb{E}_{y \leftarrow f(U_n)}[\ \mathsf{CP}(\ H_1(X) \mid f(X) = y,\ H_1\ )\ ] \\
\leq \ & \max_y \left\{\ \mathsf{CP}(\ X \mid f(X) = y\ )\ +\ \max_{x_1 \neq x_2, f(x_1) = f(x_2)} \{\ \Pr[H_1(x_1) = H_1(x_2)]\ \}\ \right\} \\
\leq \ & \max_y \{\ 2^{-r}\ +\ 2^{-r-c\log n - d}\ \} \\
\leq \ & 2^{-r+1}
\end{aligned}$$

where the first inequality follows from the fact that condition on $f(X) = y$ the collision probability of $H_1(X)$ is bounded by the collision probability of X and the probability of $H_1(x_1) = H_1(x_2)$ for $x_1 \neq x_2$. To apply Lemma B.1, let $\mathcal{W} = \{0,1\}^{m = r + c\log n + d}$, $\mathcal{Z} = f(\{0,1\}^n) \times \mathcal{H}_1$ and thus $e = d + c \cdot \log n + 1$, and define Adv as the success probability of A' on the corresponding input, i.e.,

$$\mathsf{Adv}(w, z = (y, h_1)) \overset{\text{def}}{=} \Pr[\ \mathsf{A}'(y, w, h_1) \in f'^{-1}(y, w, h_1)\ ]\ ,$$

where the probability is taken over the internal coins of A'. Thus,

$$\begin{aligned}
& \Pr[\ \mathsf{A}'(f(X), U_{r+c\log n+d}, H_1) \in f'^{-1}(f(X), U_{r+c\log n+d}, H_1)\ ] \\
= \ & \mathbb{E}[\mathsf{Adv}(U_{\mathcal{W}}, Z)]\ \geq\ \mathbb{E}[\mathsf{Adv}(W, Z)]^2/2^{e+2} \\
= \ & \Pr[\ \mathsf{A}'(f(X), H_1(X), H_1) \in f'^{-1}(f(X), H_1(X), H_1)\ ]^2/2^{d+c\cdot\log n+3} \\
> \ & (\sqrt{2^{d+3} \cdot n^c \cdot \varepsilon})^2/2^{d+c\cdot\log n+3}\ =\ \varepsilon
\end{aligned}$$

where the first inequality is due to Lemma B.1 and the second is by the assumption. This immediately implies another inverter A for f that on input y, it samples $h_1 \xleftarrow{\$} \mathcal{H}_1$, $w \leftarrow U_{r+c\log n+d}$, invokes $(x', h_1') \leftarrow \mathsf{A}'(y, w, h_1)$ and produces x' as output. In particular, A inverts f with the following probability

$$\begin{aligned}
& \Pr[\mathsf{A}(f(X)) \in f^{-1}(f(X))] \\
\geq \ & \Pr[\ \mathsf{A}'(f(X), U_{r+c\log n+d}, H_1) \in f'^{-1}(f(X), U_{r+c\log n+d}, H_1)\ ] \\
> \ & \varepsilon
\end{aligned}$$

which is a contradiction to the one-way-ness of f and thus completes the proof. □

A.4 The Third Construction

Proof of Theorem 4.3. Similar to the proof of Theorem 4.1, define $\mathcal{E}_1 \overset{\text{def}}{=} (\vec{x} \neq \vec{x'}\ \wedge\ f^q(\vec{x}) = f^q(\vec{x'})\ \wedge\ h_1(\vec{x}) = h_1(\vec{x'}))$ and $\mathcal{E}_2 \overset{\text{def}}{=} (f^q(\vec{x}) \neq f^q(\vec{x'})\ \wedge g_1(\vec{x}) = g_1(\vec{x'}))$, we have

$$\begin{aligned}
& \Pr_{\vec{x} \xleftarrow{\$} \{0,1\}^{qn},\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1),\ \vec{x'} \leftarrow \mathsf{A}(\vec{x},h,h_1)} [\ \vec{x} \neq \vec{x'}\ \wedge\ g(\vec{x}) = g(\vec{x'})] \\
\leq \ & \Pr_{\vec{x} \xleftarrow{\$} \{0,1\}^{qn},\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1),\ \vec{x'} \leftarrow \mathsf{A}(\vec{x},h,h_1)} [\ \mathcal{E}_1\ \vee \mathcal{E}_2\ ] \\
\leq \ & \Pr_{\vec{x} \xleftarrow{\$} \{0,1\}^{qn},\ h_1 \xleftarrow{\$} \mathcal{H}_1} [\ \exists\ \vec{x'} \neq \vec{x}\ \wedge\ f^q(\vec{x}) = f^q(\vec{x'})\ \wedge\ h_1(\vec{x}) = h_1(\vec{x'})\ ] \\
& + \Pr_{\vec{x} \xleftarrow{\$} \{0,1\}^{qn},\ (h,h_1) \xleftarrow{\$} (\mathcal{H},\mathcal{H}_1)} [\ \vec{x'} \leftarrow \mathsf{A}(\vec{x}, h, h_1)\ \wedge\ f^q(\vec{x}) \neq f^q(\vec{x'})\ \wedge\ g_1(\vec{x}) = g_1(\vec{x'})\ ] \\
\leq \ & 2^{-q\log n}\ +\ 2q \cdot n^{c+2} \cdot \varepsilon\ =\ n^{-q}\ +\ 2q \cdot n^{c+2} \cdot \varepsilon\ ,
\end{aligned}$$

where the second inequality follows by a union bound, and the first term of the third inequality is due to that conditioned on any $\vec{y} = f^q(\vec{X})$ random variable $\vec{X}$ is uniform over some set of size at most $(2^r{\cdot}n^c)^q$, so we apply Lemma 3.1 (setting $a = q(r + c \cdot \log n)$, $d{\geq}q\log n$ and $k = 1$) to get

$$\begin{aligned}
&\Pr_{\vec{x}\xleftarrow{\$}\{0,1\}^{qn},\ h_1\xleftarrow{\$}\mathcal{H}_1}[\ \exists\ \vec{x'} \neq \vec{x}\ \wedge\ f^q(\vec{x}) = f^q(\vec{x'})\ \wedge\ h_1(\vec{x}) = h_1(\vec{x'})\ ] \\
&= \mathbb{E}_{\vec{y}\leftarrow f^q(U_{qn})}\left[\ \Pr_{\vec{x}\xleftarrow{\$}(f^q)^{-1}(\vec{y}),\ h_1\xleftarrow{\$}\mathcal{H}_1}[\ \exists\ \vec{x'} \neq \vec{x}\ \wedge\ f^q(\vec{x}) = f^q(\vec{x'})\ \wedge\ h_1(\vec{x}) = h_1(\vec{x'})\ ]\ \right] \\
&\leq\ \mathbb{E}_{\vec{y}\leftarrow f^q(U_{qn})}[\ 2^{-q\log n}\ ]\ =\ n^{-q}\quad .
\end{aligned}$$

We proceed to the proof of bounding the second term. Suppose for contradiction that there exists A_{g_1} of running time $t - n^{O(1)}$ such that

$$\Pr_{\vec{x}\xleftarrow{\$}\{0,1\}^{qn},\ (h,h_1)\xleftarrow{\$}(\mathcal{H},\mathcal{H}_1)}[\ \vec{x'} \leftarrow \mathsf{A}_{g_1}(\vec{x}, h, h_1)\ \wedge\ f^q(\vec{x}){\neq}f^q(\vec{x'})\ \wedge\ g_1(\vec{x}) = g_1(\vec{x'})\ ]\ >\ 2q{\cdot}n^{c+2}\cdot\varepsilon$$

Then, define $\tilde{\mathsf{A}}$ as in Algorithm 3. Conditioned on that A_{g_1} finds a collision, i.e., $f^q(\vec{x}){\neq}f^q(\vec{x'})$ and $g_1(\vec{x}) = g_1(\vec{x'})$, there exists at least one $i^* \in [q]$ satisfying $f(x_{i^*}){\neq}f(x'_{i^*})$ and $\mathsf{trunc}(h(f(x_{i^*}))) = \mathsf{trunc}(h(f(x'_{i^*})))$. We have

Algorithm 3 $(\mathsf{trunc} \circ h)$-collision finder $\tilde{\mathsf{A}}$ on input (x, h).

Input: $(x, h) \xleftarrow{\$} \{0, 1\}^n \times \mathcal{H}$

Sample $\vec{x} = (x_1, \ldots, x_q) \xleftarrow{\$} \{0, 1\}^{qn}, h_1 \xleftarrow{\$} \mathcal{H}_1,\ i \xleftarrow{\$} [q]$

$\vec{x'} = (x'_1, \ldots, x'_q) \leftarrow \mathsf{A}_{g_1}(\ (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_q)\ , h, h_1)$ {i.e., replace x_i with x }

return x'_i

$$\begin{aligned}
&\Pr_{x\xleftarrow{\$}\{0,1\}^{n},\ h\xleftarrow{\$}\mathcal{H}}[\ x' \leftarrow \tilde{\mathsf{A}}(x, h)\ \wedge\ f(x){\neq}f(x')\ \wedge\ \mathsf{trunc}(h(f(x))) = \mathsf{trunc}(h(f(x')))\ ] \\
\geq\ &\Pr_{\vec{x}\xleftarrow{\$}\{0,1\}^{qn},\ (h,h_1)\xleftarrow{\$}(\mathcal{H},\mathcal{H}_1)}[\ \vec{x'} \leftarrow \mathsf{A}_{g_1}(\vec{x}, h, h_1)\ \wedge\ f^q(\vec{x}){\neq}f^q(\vec{x'})\ \wedge\ g_1(\vec{x}) = g_1(\vec{x'})\ \wedge\ i = i^*\ ] \\
>\ &2q{\cdot}n^{c+2}\varepsilon \cdot (1/q)\ =\ 2n^{c+2}\varepsilon\ ,
\end{aligned}$$

which is a contradiction to Lemma 4.1 (recall that $s' = 2\log n$) and thus completes the proof. □

B Preliminaries and Details for the Fourth Construction

B.1 UOWHFs from Unknown (Almost-)Regular OWFs

Ames et al. [1] presented an elegant construction based on any almost-regular OWFs[11], where no knowledge is required about the regularity of the OWF. Furthermore, their construction enjoys output length $\Theta(n)$ and key length $O(n{\cdot}\log n)$ and makes $O(n/\log n)$ calls to the underlying OWF. To see this, we set $s = \Omega(\log n)$ in Theorem B.1 and thus get a construction of UOWHFs by making $\kappa = O(n/\log n)$ calls to any $(\alpha, \alpha \cdot \beta)$-almost regular (t, ε)-OWF, where α and β need not to be efficiently computable, and the construction tolerates regularity slackness for any $\beta = n^{O(1)}$ or even certain $\beta = (1/\varepsilon)^{O(1)}$. We note that the number of calls $O(n/\log n)$ is optimal (for black-box constructions) in general by matching the lower bound of [2].

[11] The authors of [1] mainly stated the neat case, i.e., for $\beta = 1$ and $s = 1$, and similar to Theorem 4.3 it (implicitly in their proof) generalizes to Theorem B.1, where almost regularity and logarithmic shrinkage are considered.

Definition B.1 (the generalized iterate [1]) *Let $n \in \mathbb{N}$, function $f : \{0,1\}^n \to \{0,1\}^n$, and let $\mathcal{H}$ be a family of pairwise-independent hash functions from $\{0,1\}^{n+s}$ to $\{0,1\}^n$. For $i \le \kappa \in \mathbb{N}$, $x_1 \in \{0,1\}^n$, $v_1, \ldots, v_\kappa \in \{0,1\}^s$ and vector $\vec{h}^\kappa = (h_1, \ldots, h_\kappa) \in \mathcal{H}^\kappa$, recursively define the i^{th} randomized iterate by:*

$$x_1 \xrightarrow{f} \begin{matrix} y_1 \\ v_1 \end{matrix} \xrightarrow{h_1} x_2 \xrightarrow{f} \begin{matrix} y_2 \\ v_2 \end{matrix} \xrightarrow{h_2} \ \cdots \ \ x_\kappa \xrightarrow{f} \begin{matrix} y_\kappa \\ v_\kappa \end{matrix} \xrightarrow{h_\kappa} x_{\kappa+1} \xrightarrow{f} y_{\kappa+1}$$

$$y_i = f(x_i), \ x_{i+1} = h_i(y_i \| v_i)$$

We denote the κ^{th} iterate by function g_f^κ, i.e., $y_{k+1} = g_f^\kappa(v_1\|\ldots\|v_\kappa, x_1, \vec{h}^\kappa)$, where $x_1 \xleftarrow{\$} \{0,1\}^n$, $v_1, \ldots, v_\kappa \xleftarrow{\$} \{0,1\}^s$, $\vec{h}^\kappa \leftarrow Shoup(U_{O(n \cdot \log n)})$ and $Shoup : \{0,1\}^{O(n \cdot \log n)} \to \mathcal{H}^\kappa$ is Shoup's generator [22].

Theorem B.1 (UOWHFs from unknown almost-regular OWFs [1]) *For security parameter $n \in \mathbb{N}$, any (not necessarily efficient) $\alpha = \alpha(n)$, $\beta = \beta(n) \ge 1$ and any efficiently computable $s = s(n)$, $\kappa = \kappa(n)$ such that $s(n) \cdot \kappa(n) \ge n + s(n)$, let $f : \{0,1\}^n \to \{0,1\}^n$ be any $(\alpha, \alpha \cdot \beta)$-almost regular (length-preserving) (t, ε)-one-way function, let g_f^κ, $\mathcal{H}$ and $Shoup : \{0,1\}^{q \in O(n \cdot \log n)} \to \mathcal{H}^\kappa$ be defined as in Definition B.1. Then, we have that*

$$\mathcal{G} = \{ g_f : \{0,1\}^{s \cdot \kappa} \to \{0,1\}^n \ , \ g_f(z) = g_f^\kappa(z, x_1, Shoup(u)), z \stackrel{def}{=} v_1\|\ldots\|v_\kappa \ \}$$

where each function g_f is described by $(x_1, u) \in \{0,1\}^n \times \{0,1\}^q$, is a family of $(t - n^{O(1)}, \mathsf{poly}(\beta, 2^s, \kappa) \cdot \varepsilon^{\Theta(1)})$-UOWHFs with key length $O(n \cdot \log n)$, output length n and at least s bits of shrinkage.
Notice that x_1 is not input to hash function g_f but the part (together with u) of the description of g_f.

B.2 Some Lemmas and Proofs Reproduced and Adapted from [24]

A technical Lemma. To prove Lemma 4.3 and Lemma 5.1, we will need the following lemma that was folklore in leakage-resilient cryptography and was explicitly stated in [24]. Informally, it states that "if any algorithm wins a one-sided game (e.g., inverting a OWF) on uniformly sampled challenges only with some negligible probability, then it cannot do much better (beyond a negligible advantage) in case that the challenges are sampled from any distribution of logarithmic Rényi entropy deficiency".

Lemma B.1 (one-sided game on imperfect randomness [24]) *For any $e \le m \in \mathbb{N}$, let $\mathcal{W}$ and $\mathcal{Z}$ be any sets with $|\mathcal{W}| = 2^m$, let $\mathsf{Adv} : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be any (deterministic) real-valued function, let (W, Z) be any random variable over set $\mathcal{W} \times \mathcal{Z}$ with $\mathsf{CP}(W|Z) \le 2^{e-m}$, we have*

$$\mathbb{E}[\mathsf{Adv}(W,Z)] \ \le \ \sqrt{2^{e+2} \cdot \mathbb{E}[\mathsf{Adv}(U_\mathcal{W}, Z)]} \tag{6}$$

where $U_\mathcal{W}$ denotes uniform distribution over $\mathcal{W}$ and independent of Z.

Definition B.2 (bounded-width layered branching program - LBP) *An (s,k,v)-LBP M is a finite directed acyclic graph whose nodes are partitioned into $k + 1$ layers indexed by $\{1, \ldots, k+1\}$. The first layer has a single node (the source), the last layer has two nodes (sinks) labeled with 0 and 1, and each of the intermediate layers has up to 2^s nodes. Each node in the $i \in [k]$ layer has exactly 2^v outgoing labeled edges to the $(i+1)^{th}$ layer, one for every possible string $h_i \in \{0,1\}^v$.*

Theorem B.2 (bounded-space generator [18, 15]) *Let $s = s(n), k = k(n), v = v(n) \in \mathbb{N}$ and $\varepsilon = \varepsilon(n) \in (0,1)$ be polynomial-time computable functions. Then, there exist a polynomial-time computable function $q = q(n) = \Theta(v + (s + \log(k/\varepsilon)) \cdot \log k)$ and a generator $BSG : \{0,1\}^q \to \{0,1\}^{k \cdot v}$ that runs in time $\mathsf{poly}(s,\ k,\ v,\ \log(1/\varepsilon))$, and ε-fools every (s,k,v)- LBP M, i.e.,*

$$| \ \Pr[M(U_{k \cdot v}) = 1] \ - \ \Pr[M(BSG(U_n)) = 1] \ | \ \le \ \varepsilon \ .$$

We state some inequalities in Lemma B.2 below along with their proofs (reproduced from [24]). It is not hard to see that (7), (9) and (11) hold for the randomized version. For example, we have by the pairwise independence of $\mathcal{H}$ that all $x_1, \ldots, x_k$ are i.i.d. to U_n so that (9) immediately follows and (11) follows by a Chernoff bound. Then, for every inequality (8), (10) and (12), we define an LBP (see Definition B.2) and argue that the advantage of the LBP on $\vec{H}^k$ and $BSG(U_q)$ is bounded by 2^{-2n} and thus (8), (10) and (12) follow from their respective counterparts (7), (9) and (11) by adding an additive term 2^{-2n}.

Lemma B.2 (some inequalities from [24]) *For any $n, k \in \mathbb{N}$, it holds that*

$$\mathsf{CP}(\ Y_k \mid \vec{H}^k) \ \le\ k{\cdot}2^{\max -n+1}\ , \tag{7}$$

$$\mathsf{CP}(\ Y'_k \mid U_q) \ \le\ k{\cdot}2^{\max -n+1} + 2^{-2n}, \tag{8}$$

$$\forall j \in [k]: \ \ \Pr[\mathcal{E}_j] \ \ge\ n^{-c}\ \ , \tag{9}$$

$$\forall j \in [k]: \ \ \Pr[\mathcal{E}'_j] \ \ge\ n^{-c} - 2^{-2n}\ , \tag{10}$$

$$\Pr[\mathcal{E}_1 \vee \mathcal{E}_2 \vee \ldots \vee \mathcal{E}_k] \ \ge\ 1 - 2^{-k/n^{2c}}\ , \tag{11}$$

$$\Pr[\mathcal{E}'_1 \vee \mathcal{E}'_2 \vee \ldots \vee \mathcal{E}'_k] \ \ge\ 1 - 2^{-k/n^{2c}} - 2^{-2n}\ , \tag{12}$$

where $Y_k = f^k(X_1, \vec{H}^k)$ and $Y'_k = f^k(X_1, BSG(U_q))$.

Proof of (7),(9) and (11). We have that x_1, $x_2 = h_1(y_1)$, $\ldots$, $x_k = h_{k-1}(y_{k-1})$ are all i.i.d. to U_n due to the universality of $\mathcal{H}$, which implies that $\mathcal{E}_1, \ldots$ and $\mathcal{E}_k$ are i.i.d. events with probability at least n^{-c}. For every $j \in [k]$, define $\zeta_j = 1$ iff $\mathcal{E}_j$ occurs (and $\zeta_j = 0$ otherwise). It follows by a Chernoff-Hoeffding bound that

$$\Pr[\ (\neg\mathcal{E}_1) \wedge \ldots \wedge (\neg\mathcal{E}_k)\ ] \ =\ \Pr[\ \sum_{j=1}^{k} \zeta_j = 0\ ] \ \le\ 2^{-k/n^{2c}}$$

which yields (11) by taking a negation. Finally, Regarding (7), consider two instances of the random iterate seeded with independent x_1 and x'_1 and a common random $\vec{h}^k$, the collision probability is upper bounded by the sum of events that the first collision occurs on points $y_1, y_2, \ldots, y_k \in \mathcal{Y}_{[\max]}$ respectively. We thus have by the pairwise independence of $\mathcal{H}$ that

$$\begin{aligned}
& \mathsf{CP}(\ Y_k\ \mid \vec{H}^k) \\
\le\ & \Pr_{x_1, x'_1 \xleftarrow{\$} \{0,1\}^n} [f(x_1) = f(x'_1)] \ +\ \sum_{j=2}^{k} \left(\Pr_{y_{j-1} \ne y'_{j-1},\ h_{j-1} \xleftarrow{\$} \mathcal{H}} [f(x_j) = f(x'_j)] \right) \\
\le\ & k \cdot \mathsf{CP}(f(U_n)) \\
\le\ & k \sum_{i=1}^{\max} \sum_{y \in \mathcal{Y}_i} \Pr[f(U_n) = y]{\cdot}2^{i-n} \\
=\ & k \sum_{i=1}^{\max}\ \Pr[f(U_n) \in \mathcal{Y}_i]{\cdot}2^{i-n} \\
\le\ & k{\cdot}2^{\max -n}(1 + 2^{-1} + \ldots + 2^{-(\max -1)}) \ \le\ k{\cdot}2^{\max -n+1}\ .
\end{aligned}$$

□

Proof of (8). For any $k \in \mathbb{N}$, consider the following $(2n, k, \log|\mathcal{H}|)$-LBP M_1: on source node input $(y_1 = f(x_1), y'_1 = f(x'_1))$. For $1 \le i \le k$, at each i^{th} layer M_1 computes $y_i := f(h_{i-1}(y_{i-1}))$ and $y'_i := f(h_{i-1}(y'_{i-1}))$. Finally, at the $(k+1)^{th}$ layer M_1 outputs 1 iff $y_k = y'_k \in \mathcal{Y}_{\max}$. Imagine running two iterates with random x_1, x_1' and seeded by a common hash function from distribution either $\vec{H}^k$ or $BSG(U_q)$, we have

$$\mathsf{CP}(\ Y_k \mid \vec{H}^k\ )\ =\ \Pr_{(x_1,x_1')\leftarrow U_{2n},\ \vec{h}^k\leftarrow\vec{H}^k}[M_1(x_1,x_1',\vec{h}^k)=1]$$

$$\mathsf{CP}(\ Y_k' \mid BSG(U_q)\ )\ =\ \Pr_{(x_1,x_1')\leftarrow U_{2n},\ \vec{h}^k\leftarrow BSG(U_q)}[M_1(x_1,x_1',\vec{h}^k)=1]$$

and thus

$$\begin{aligned}
&\mid \mathsf{CP}(Y_k \mid \vec{H}^k\ )\ -\ \mathsf{CP}(\ Y_k' \mid BSG(U_q)\ )\mid\\
\leq\ & \mathbb{E}_{(x_1,x_1')\leftarrow U_{2n}}\left[\ \mid \Pr[M_1(x_1,x_1',\vec{H}^k)=1]\ -\ \Pr[M_1(x_1,x_1',BSG(U_q))=1]\mid\ \right]\\
\leq\ & 2^{-2n}\quad .
\end{aligned}$$

It follows by (7) that

$$\mathsf{CP}(Y_k' \mid BSG(U_q)\ )\ \leq\ \mathsf{CP}(\ Y_k \mid \vec{H}^k\ )\ +\ 2^{-2n}\ \leq\ k{\cdot}2^{\max-n+1}+2^{-2n}\quad .$$

Note that for any $\vec{h}^k$ and any $u_1,u_2{\in}BSG^{-1}(\vec{h}^k)$,

$$\mathsf{CP}(\ Y_k' \mid U_q=u_1\ )\ =\ \mathsf{CP}(\ Y_k' \mid U_q=u_2)\ =\ \mathsf{CP}(\ Y_k' \mid BSG(U_q)=\vec{h}^k\ )\ .$$

Therefore,

$$\mathsf{CP}(Y_k' \mid U_q\ )\ =\ \mathsf{CP}(Y_k' \mid BSG(U_q)\ )\ \leq\ k{\cdot}2^{\max-n+1}+2^{-2n}\quad .$$

□

Proof of (10). Similar to that of (8), we define another $(n+1,k,\log|\mathcal{H}|)$-LBP M_2 that on source node input $(x_1,\mathsf{tag}_1=0)$, it computes $y_i:=f(x_i)$, $x_{i+1}:=h_i(y_i)$, for every $i\leq k$ and sets $\mathsf{tag}_i=1$ if $i=j$ and $y_i\in\mathcal{Y}_{\max}$ (or otherwise $\mathsf{tag}_i:=\mathsf{tag}_{i-1}$). Finally, it outputs tag_k. Thus,

$$\Pr[\mathcal{E}_j']\geq\Pr[\mathcal{E}_j]-2^{-2n}\geq n^{-c}-2^{-2n}\ .$$

□

Proof of (12). Consider the following $(n+1,k,\log|\mathcal{H}|)$-LBP M_3: on source node input (x_1,tag_1) and layered input vector $\vec{h}^k$, it computes $y_i:=f(x_i)$, $x_{i+1}:=h_i(y_i)$, at each i^{th} layer, and sets $\mathsf{tag}_i=1$ iff either $\mathsf{tag}_{i-1}=1$ or $y_i\in\mathcal{Y}_{\max}$. Finally, M_3 outputs tag_k. By the bounded space generator we have

$$\begin{aligned}
&\mid\ \Pr[M_3(X_1,\vec{H}^k)=1]\ -\ \Pr[M_3(X_1,BSG(U_q)=1]\mid\\
=\ &\mid\Pr[\ \bigvee_{i=1}^{k}\mathcal{E}_i\ ]-\Pr[\ \bigvee_{i=1}^{k}\mathcal{E}_i'\ ]\mid\ \leq\ 2^{-2n}\ ,
\end{aligned}$$

and thus by (11)

$$\Pr[\ \bigvee_{i=1}^{k}\mathcal{E}_i'\ ]\ \geq\ \Pr[\ \bigvee_{i=1}^{k}\mathcal{E}_i\ ]\ -\ 2^{-2n}\ \geq\ 1-2^{-k/n^{2c}}-2^{-2n}\ .$$

□

B.3 $\mathcal{F}$ Is a Family of One-way Functions

Proof of Lemma 5.1. Assume for contradiction that there exists A (of running time $t - n^{O(1)}$) that inverts f_u with some non-negligible ε_{A}, i.e.,

$$\Pr_{u \leftarrow U_q,\ x \xleftarrow{\$} \{0,1\}^n} [\ \mathsf{A}(u, f_u(x)) \in f_u^{-1}(\ f_u(x)\ )\ ] \ \geq \varepsilon_{\mathsf{A}} \quad .$$

We use shorthand $\mathcal{C}$ for the event that A inverts f_u, i.e.,

$$\mathcal{C} \ \stackrel{\mathsf{def}}{=} \ \Big((X_1, U_q) \in \{\ (x,u) : \mathsf{A}(u, f_u(x)) \in f_u^{-1}(f_u(x))\ \} \Big)$$

and thus

$$\begin{aligned}
\varepsilon_{\mathsf{A}} \quad &\leq \quad \Pr[\mathcal{C}] \\
&= \quad \Pr[\mathcal{C} \wedge (\mathcal{E}'_1 \vee \mathcal{E}'_2 \vee \ldots \vee \mathcal{E}'_k)] + \Pr[\mathcal{C} \wedge \neg(\mathcal{E}'_1 \vee \mathcal{E}'_2 \vee \ldots \vee \mathcal{E}'_k)] \\
&\leq \quad \Pr[\bigvee_{j=1}^{k} (\mathcal{C} \wedge \mathcal{E}'_j)] \ + \ \Pr[\neg(\mathcal{E}'_1 \vee \mathcal{E}'_2 \vee \ldots \vee \mathcal{E}'_k)] \\
&\leq \quad \sum_{j=1}^{k} \Pr[\mathcal{C} \wedge \mathcal{E}'_j] \ + \ (2^{-k/n^{2c}} + 2^{-2n})
\end{aligned}$$

where the third inequality follows from the union bound and (12). We have by an averaging argument that there exists $j^* \in [k]$ such that $\Pr[\mathcal{C} \wedge \mathcal{E}'_{j^*}] \geq (\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})/k$. That is, conditioned on event $\mathcal{E}'_{j^*}$, algorithm A inverts $f_u(x){=}f^k(x, BSG(u))$ to produce $x' \in {f_u}^{-1}(f_u(x))$ with probability

$$\Pr[\ \mathcal{C} \mid \mathcal{E}'_{j^*}\ ] \ = \ \frac{\Pr[\mathcal{C} \wedge \mathcal{E}'_{j^*}]}{\Pr[\ \mathcal{E}'_{j^*}\ ]} \ \geq \ \Pr[\mathcal{C} \wedge \mathcal{E}'_{j^*}] \ \geq \ (\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})/k \ \ .$$

The above implies an algorithm M^{A} (as given in Algorithm 4) that inverts $y_{j^*} = f^{j^*}(x, BSG(u))$ with respect to f to get $x_{j^*} \in f^{-1}(y_{j^*})$ with almost the same probability. Loosely speaking, on input y_{j^*}, the algorithm M^{A} evaluates the iterate to obtain y_k, invokes A on y_k to get x_1, and produces x_{j^*} (again by evaluating the iterate on x_1) as a candidate preimage of y_{j^*} under function f. The only issue is that j^* is unknown, so it simply makes a random guess $j \xleftarrow{\$} [k]$, which hits j^* with probability $1/k$. Therefore,

---

Algorithm 4 M^{A}.

Input: $u \in \{0,1\}^q$, $y \in \{0,1\}^n$

Sample $j \xleftarrow{\$} [k]$;
$\big(\vec{h}^k = (h_1, \ldots, h_k)\big) := BSG(u)$;
Let $\tilde{y}_j := y$;
FOR $i = j+1$ TO k
  Compute $\tilde{x}_i := h_{i-1}(\tilde{y}_{i-1})$, $\tilde{y}_i := f(\tilde{x}_i)$;
$\tilde{x}_1 \leftarrow \mathsf{A}(u, \tilde{y}_k)$;
FOR $i = 1$ TO $j-1$
  Compute $\tilde{y}_i := f(\tilde{x}_i)$, $\tilde{x}_{i+1} := h_i(\tilde{y}_i)$;

Output: $\tilde{x}_j$

---

it holds that

$$\Pr[\ \mathsf{M}^{\mathsf{A}}(U_q, Y'_{j^*};\ j\ ) \in f^{-1}(Y'_{j^*}) \ \mid \ j = j^* \ \wedge \ \mathcal{E}'_{j^*}\ ] \ \ \geq \ (\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})/k \ , \qquad (13)$$

where we recall that $Y'_{j^*} = f^{j^*}(X_1, BSG(U_q))$. We state in Claim 2 that replacing the above Y'_{j^*} (which is correlated to U_q) with $f(U_n)$ (which is independent of U_q) the inverting probability weakens only by a $1/\mathsf{poly}(n)$ factor and thus makes M^{A} an inverter for f.

$$\begin{aligned}\varepsilon &\geq \Pr[\ \mathsf{M}^{\mathsf{A}}(\ U_q, f(U_n);\ j\ ) \in f^{-1}(f(U_n))\ ] \\ &= \Pr[j = j^*] \cdot \Pr[f(U_n) \in \mathcal{Y}_{\max}\ ] \cdot \Pr[\ \mathsf{M}^{\mathsf{A}}(\ U_q, f(U_n);\ j\ ) \in f^{-1}(f(U_n))\ \mid\ \mathcal{E}_{hit}\ ] \\ &\geq \frac{(\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})^2}{2^8 \cdot k^4 \cdot n^{3c}}\ ,\end{aligned}$$

where $\mathcal{E}_{hit} \stackrel{\text{def}}{=} (j = j^*\ \wedge\ f(U_n) \in \mathcal{Y}_{\max})$, the first inequality is due to the one-way-ness of f. This yields an upper bound on ε_{A} (by taking a square root) as desired and thus completes the proof. □

THE RATIONALE FOR CLAIM 2. By Lemma B.1, the collision probability of (U_q, Y'_{j^*}) conditioned on $\mathcal{E}'_{j^*}$ is small enough and close to that of the uniform distribution of $(U_q, U_{\mathcal{Y}_{\max}})$. Thus, any algorithm that inverts the former (i.e., Y'_{j^*} given U_q) with a non-negligible probability will invert the the latter (i.e., $U_{\mathcal{Y}_{\max}}$ given uncorrelated U_q) with also a non-negligible probability.

Claim 2 *With shorthand* $\mathcal{E}_{hit} \stackrel{def}{=} (j = j^*\ \wedge\ f(U_n) \in \mathcal{Y}_{\max})$ *we have that*

$$\Pr[\ \mathsf{M}^{\mathsf{A}}(\ U_q, f(U_n);\ j\ ) \in f^{-1}(f(U_n))\ \mid\ \mathcal{E}_{hit}\ ]\ \geq\ \frac{(\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})^2}{2^8 \cdot k^3 \cdot n^{2c}}\ . \tag{14}$$

Proof of Claim 2. To apply Lemma B.1, let $\mathcal{W} = \{0,1\}^q \times \mathcal{Y}_{\max}$, let Z be empty set, W be the distribution of (U_q, Y'_{j^*}) conditioned on $\mathcal{E}'_{j^*}$ (i.e., $Y'_{j^*} \in \mathcal{Y}_{\max}$), and define

$$\mathsf{Adv}(u, y)\ \stackrel{\text{def}}{=}\ \Pr[\ \mathsf{M}^{\mathsf{A}}(u, y;\ j\ ) \in f^{-1}(y) \mid j = j^*\ ]\ .$$

where the probability is taken over the internal coins of M^{A}. Thus, we have

$$\mathsf{Adv}(W)\ =\ \Pr[\ \mathsf{M}^{\mathsf{A}}(U_q, Y'_{j^*};\ j\ ) \in f^{-1}(Y'_{j^*})\ \mid\ j = j^*\ \wedge\ \mathcal{E}'_{j^*}\ ]\ .$$

and

$$\begin{aligned}\mathsf{CP}(W) &= \mathsf{CP}(\ (U_q, Y'_{j^*}\ ) \mid \mathcal{E}'_{j^*}\ ) = \frac{\mathsf{CP}(\ (U_q, Y'_{j^*}\ )\ \wedge\ \mathcal{E}'_{j^*})}{\Pr[\ \mathcal{E}'_{j^*}\ ]^2} \\ &\leq \frac{\mathsf{CP}(\ (U_q, Y'_{j^*})\ )}{\Pr[\ \mathcal{E}'_{j^*}\ ]^2} \\ &\leq \frac{2^{-q} \cdot \mathsf{CP}(\ Y'_{j^*}\ |U_q\ )}{(n^{-c} - 2^{-2n})^2} \\ &\leq \frac{j^* \cdot 2^{\max - n + 1} + 2^{-2n}}{(n^{-2c}/4) \cdot 2^q}\ \leq\ \frac{16k \cdot n^{2c}}{2^{n - \max + q}}\ \leq\ \underbrace{(32k \cdot n^{2c})}_{2^e} \cdot 2^{-m}\ ,\end{aligned}$$

where the fourth inequality is due to $2^{-2n} \leq j^* \cdot 2^{\max - n + 1}$ and $j^* \leq k$ and the fifth inequality is by

$2^{m-q} = |\mathcal{Y}_{\max}| \le 1/2^{\max-1-n}$. We thus have

$$\begin{aligned}
& \Pr[\ \mathsf{M}^{\mathsf{A}}(\ U_q, f(U_n);\ j\ ) \in f^{-1}(f(U_n))\ \mid\ j=j^*\ \wedge\ f(U_n)\in\mathcal{Y}_{\max}\ ] \\
= & \sum_{(u,y)\in\{0,1\}^q\times\mathcal{Y}_{\max}} 2^{-q}\cdot \Pr[f(U_n)=y \mid f(U_n)\in\mathcal{Y}_{\max}]\cdot \mathsf{Adv}(u,y) \\
\ge & \sum_{(u,y)\in\{0,1\}^q\times\mathcal{Y}_{\max}} 2^{-q}\cdot \frac{1}{2|\mathcal{Y}_{\max}|}\cdot \mathsf{Adv}(u,y) \\
= & \frac{\mathbb{E}[\mathsf{Adv}(U_{\mathcal{W}})]}{2} \ \ge\ \frac{\mathbb{E}[\mathsf{Adv}(W)]^2}{2^{e+3}} \\
\ge & \frac{(\varepsilon_{\mathsf{A}} - 2^{-k/n^{2c}} - 2^{-2n})^2}{2^8\cdot k^3\cdot n^{2c}}\ ,
\end{aligned}$$

where the first inequality is because for any $y\in\mathcal{Y}_{\max}$ we have

$$\begin{aligned}
\Pr[f(U_n)=y \mid f(U_n)\in\mathcal{Y}_{\max}] \ &=\ \frac{\Pr[f(U_n)=y]}{\sum_{y^*\in\mathcal{Y}_{\max}}\Pr[f(U_n)=y^*]} \\
=\ \frac{1}{\sum_{y^*\in\mathcal{Y}_{\max}}\frac{\Pr[f(U_n)=y^*]}{\Pr[f(U_n)=y]}} \ &\ge\ \frac{1}{2|\mathcal{Y}_{\max}|}\ ,
\end{aligned}$$

the second inequality follows from (6) and the third is due to (13) and $2^e = 32k\cdot n^{2c}$. □

B.4 Putting Things Together

Construction 1 (UOWHF construction #4) *For constant c and $k = n^{2c}\cdot\log n\cdot\omega(1)$, let f be as defined in Definition 5.1, let f^k and $BSG(\cdot)$ be as defined in Definition 5.2, let $\mathcal{F}=\{f_u\}$ be defined as in (2), and further define $\mathcal{G}$ as in Theorem B.1, i.e.,*

$$\mathcal{G} = \{g_{f_u} : \{0,1\}^{\kappa\cdot s}\to\{0,1\}^n\ ,\ g_{f_u}(z) = g^{\kappa}_{f_u}(z, x_1, Shoup(u'))\ \}$$

where g_{f_u} is described by string $(x_1,u,u')\in\{0,1\}^n\times\{0,1\}^{O(n\cdot\log n)}\times\{0,1\}^{O(n\cdot\log n)}$, $g^{\kappa}_{f_u}$ and $Shoup(\cdot)$ are as defined in Definition B.1 (use f_u in place of f), and $\kappa\cdot s\ge n+s$ (e.g., set $s=\log n$, $\kappa=\Omega(n/\log n)$).

Why a family of OWFs? Note that the UOWHF g_{f_u} operates on input z and enjoys output length $\Theta(n)$, and it is described by key $(x_1,u,u')\in\{0,1\}^{O(n\cdot\log n)}$. An alternative is to view f_u as a single one-way function rather than a family of OWFs, i.e., $\tilde{f}(x,u)\stackrel{\text{def}}{=}(f_u(x),u)$ and plug $\tilde{f}$ into the UOWHF construction as in Theorem B.1. However, in this case, the output and key lengths of the UOWHF are $O(n\cdot\log n)$ and $O(n\cdot\log^2 n)$ respectively since $\tilde{f}$ now has input and output length $O(n\cdot\log n)$.

Proof sketch of Theorem 5.1. Consider a (t,ε)-OWF f as defined in Definition 5.1. Although f is far from regular, iterating it (as defined in Construction 1) sufficiently many, say $k=n^{2c}\cdot\log n\cdot\omega(1)$, times yields a family of one-way functions $\mathcal{F}$ with description size $O(n\cdot\log n)$, as stated in Lemma 5.1. Furthermore, Lemma 5.2 states that, for $\alpha = 2^{\max-a-1}$ and any $\beta = 2^{2a+2}\ge 4$, a random function $f_u \xleftarrow{\$} \mathcal{F}$ is $(\alpha,\ \alpha\cdot\beta)$-almost regular except for a $(O(k/\sqrt{\beta})+\mathsf{negl}(n))$-fraction. Therefore, plug f_u into Theorem B.1 and set $s=\log n$, $\beta=(1/\varepsilon^{O(1)})$ for some small enough constant $O(1)$ so that $\mathsf{poly}(\beta,2^s,\kappa)\cdot\varepsilon^{\Theta(1)}$ remains negligible, we obtain a family of UOWHFs with output length $\Theta(n)$ and key length $O(n\cdot \log n)$. In total, it makes $\kappa=O(n/\log n)$ calls to f^k for $k=n^{2c}\cdot\log n\cdot\omega(1)$ and thus $O(n^{2c+1}\cdot\omega(1))$ calls to f. □