Simple Constructions from (Almost) Regular One-Way Functions

Noam Mazor[1]* and Jiapeng Zhang[2]

[1] Tel-Aviv University, Israel
noammaz@gmail.com
[2] University of Southern California, USA
jiapengz@usc.edu

Abstract. Two of the most useful cryptographic primitives that can be constructed from one-way functions are *pseudorandom generators* (PRGs) and *universal one-way hash functions* (UOWHFs). In order to implement them in practice, the efficiency of such constructions must be considered. The three major efficiency measures are: the *seed length*, the *call complexity* to the one-way function, and the *adaptivity* of these calls. Still, the optimal efficiency of these constructions is not yet fully understood: there exist gaps between the known upper bound and the known lower bound for black-box constructions.

A special class of one-way functions called *unknown-regular* one-way functions is much better understood. Haitner, Harnik and Reingold (CRYPTO 2006) presented a PRG construction with semi-linear seed length and linear number of calls based on a method called *randomized iterate*. Ames, Gennaro and Venkitasubramaniam (TCC 2012) then gave a construction of UOWHF with similar parameters and using similar ideas. On the other hand, Holenstein and Sinha (FOCS 2012) and Barhum and Holenstein (TCC 2013) showed an almost linear call-complexity lower bound for black-box constructions of PRGs and UOWHFs from one-way functions. Hence Haitner et al. and Ames et al. reached *tight* constructions (in terms of seed length and the number of calls) of PRGs and UOWHFs from regular one-way functions. These constructions, however, are adaptive.

In this work, we present non-adaptive constructions for both primitives which match the optimal call-complexity given by Holenstein and Sinha and Barhum and Holenstein. Our constructions, besides being simple and non-adaptive, are robust also for *almost-regular* one-way functions.

Keywords: pseudorandom generator· universal one-way hash function.

1 Introduction

A wide class of cryptographic primitives can be constructed from *one-way functions*, which is the minimal assumption for cryptography. Informally, a function

* Research supported by Israel Science Foundation grant 666/19 and the Blavatnik Interdisciplinary Cyber Research Center at Tel-Aviv University.

f is called a one-way function if it is easy to compute, but hard to invert by polynomial-time algorithms. Two important primitives that can be constructed from one-way functions are *pseudorandom generators* (PRGs) [22, 5] and *universal one-way hash functions* (UOWHFs) [19]. These two primitives are useful for constructing even more powerful primitives such as encryption, digital signatures and commitments. Thus, an improvement in the efficiency of constructions for PRGs and UWOHFs would have an effect on other primitives. Yet, the optimal efficiency of these two basic primitives is not fully understood.

There are several important efficiency measures to account for when considereing PRGs and UOWHFs. For PRG constructions, one aims to minimize the seed length and the number of calls to the one-way function f. For UOWHF constructions, there is a need to minimize the key length and the number of calls to f. Besides these two measurements, another important parameter is the *adaptivity* of the calls. That is, if the inputs for the one-way function are independent of the output of previous calls, then the construction can be implemented in parallel. By contrast, if the calls are adaptive, one must make them sequentially.

Constructions. Much progress was done since the notion of PRGs has been introduced. The first construction of pseudorandom generators was given by Blum and Micali [5] based on the assumption that a specific function is hard to invert. This construction was generalized by Yao [22] to work with any one-way permutation. Since then, many subsequent works made effort to construct PRGs based on arbitrary one-way functions. Notably, through introducing the *randomized iterate*[3] method, Goldreich, Krawczyk and Luby [8] gave a PRG construction from any *unknown-regular* one-way function. The notion of regular one-way function is a refinement of a one-way permutation: A one-way function f is called *regular* if for every n and x, x' with $|x| = |x'| = n$ it holds that $\left|f^{-1}(f(x))\right| = \left|f^{-1}(f(x'))\right|$. We say that the function is *unknown-regular* if the *regularity parameter*, $\left|f^{-1}(f(x))\right|$, may not be a computable function of n. More recently, the randomized iterate method was further studied by [11, 23], who reached a construction of PRGs from any unknown-regular one-way functions, while having $O(n \log n)$ seed length and making $O(n/\log n)$ calls to the one-way function. [25] improved the seed length up to $\omega(n)$ by using a transformation that converts any unknown-regular function into a function that is known-regular on its image.

For arbitrary one-way function, a seminal work by Håstad, Impagliazzo, Levin and Luby [15] gave the first PRG construction. Since then, the efficiency has been improved by many works ([10, 16, 13, 21]). Currently, the state-of-the-art construction of PRGs due to [21] uses $O(n^3)$ bits of random seed and $O(n^3)$ adaptive calls to the one-way function, or alternatively seed of size $O(n^4)$ with non-adaptive calls [13, 21]. [4]

[3] For a one-way function f and pairwise independent hash functions $h_1, \ldots, h_k$, the k-th randomized iteration of f is $f \circ h_k \circ \cdots \circ f \circ h_1 \circ f$.

[4] We ignore low order terms for this introduction.

The constructions of UOWHFs use similar ideas to the constructions of PRGs. Still, the best PRGs constructions from arbitrary one-way functions are more efficient than the best known UOWHFs constructions. Rompel [20] gave the first UOWHF construction from arbitrary one-way functions. The efficiency was improved by [12], who gave a construction of UOWHF using $O(n^6)$ adaptive calls with a key of size $O(n^7)$. Constructing a UOWHF using $O(n^3)$ calls to the one-way function is still an interesting open question.

The efficiency of UOWHF based on an unknown-regular one-way function is similar to the efficiency of the unknown-regular based PRGs. Interestingly, this was shown by [2] using the same method of randomized iterate, resulting in a construction that uses $\Theta(n)$ key length and $\Theta(n)$ calls. We stress that when the regularity of f is known (i.e., can be computed efficiently given n), there are much more efficient constructions for both PRGs and UOWHFs ([9, 7, 19, 23]).

Lower bounds. The lower bounds for black-box constructions are relatively far from the upper bounds. In this line of work, there are two incomparable types of results. The first type, due to [6] is stated with terms of the stretching and compression of the PRG and UOWHF, respectively. Specifically, [6] showed that any black-box PRG construction $G\colon \{0,1\}^m \to \{0,1\}^{m+s}$ from f must use $\Omega(s/\log n)$ calls to f. Similarly, any black box UWOHF construction with input size m and output size $m-s$ must use $\Omega(s/\log n)$ calls. In the second type of results [17] showed that any black-box PRG construction from f must use $\Omega(n/\log n)$ calls to f, even for 1-bit stretching. [3] showed similar results for 1-bit compressing UWOHF.

As mentioned, there is a substantial gap between the aforementioned lower and upper bounds. One explanation for that gap is that all of the above lower bounds hold even when the one-way function f is *unknown-regular*. For this case, these bounds are known to be tight with the mentioned above constructions, which are based on *randomized iterations*. These constructions, however, are adaptive.

1.1 Our Contribution

In this paper, we give *non-adaptive* constructions of tight call complexity for PRGs and UOWHFs from unknown-regular one-way functions. Both of our constructions are quite simple and are very similar to each other. Same as previous results, the security of our constructions holds also if f is only almost-regular ([23]), which means that for every $|x| = |x'|$, the ratio between $\left|f^{-1}(f(x))\right|$ and $\left|f^{-1}(f(x'))\right|$ is only bounded by a polynomial in $|x|$ (compared to a ratio of 1, in the case of regular functions).

The seed (or key) length in our construction for PRGs (or UOWHFs respectively) is $O(n^2)$, compared to $\tilde{O}(n)$ bits in the previous adaptive constructions. This seems unavoidable and raises an interesting open question.[5]

[5] By [17], $\Omega(n)$ calls are necessary for any black-box construction. Since for non-adaptive constructions the uniformly random calls seem the only reasonable way

Our constructions and results In this section, we present our constructions. The results here are stated for regular one-way functions but can be naturally expanded to almost-regular functions, as stated in Sections 3 and 4. The main crux of the construction is the following observation. For regular f and i.i.d uniform random variables X_1, X_2 over $\{0,1\}^n$, given any fixing of $f(X_1)$, both the entropy and min-entropy of the pair $X_1, f(X_2)$ are exactly n. To see the above, recall that for regular f with (unknown) regularity parameter r, it holds that there are exactly r possible values for X_1 given $f(X_1)$, and exactly $2^n/r$ possible values for $f(X_2)$. Thus, the regularity parameter r "cancels out" when considering the number of possible values (given $f(X_1)$) of the pair $X_1, f(X_2)$, which is $r \cdot 2^n/r = 2^n$. In the PRG construction, we exploit this fact by using a universal family of hash functions $\mathcal{H}$ (and the Goldreich-Levin therorem) in order to extract pseudo-uniform bits. In the UOWHF construction, we use similar ideas in order to compress the pair $X_1, f(X_2)$ without creating too many collisions. For both constructions, we need additional properties from the universal family $\mathcal{H}$ that we ignore for this introduction. See more details in Sections 3 and 4. We next present the constructions. The main ideas of the proofs for the following theorems are described in Section 1.2.

A simple construction of PRGs from regular one-way functions. We start with a description of our PRG construction. Let $\mathcal{H} = \left\{h : \{0,1\}^{2n} \to \{0,1\}^{n+\log n}\right\}$ be a family of 2-universal hash functions. For a regular one-way function $f\colon \{0,1\}^n \to \{0,1\}^n$ and an integer $t \in \mathbb{N}$,[6] the generator $G_t : \mathcal{H} \times \{0,1\}^{n(t+1)} \to \mathcal{H} \times \{0,1\}^{t\cdot(n+\log n)}$ is given by

$$G_t(h, x_1, \ldots, x_{t+1}) = (h, h(x_1, f(x_2)), \ldots, h(x_t, f(x_{t+1})))$$

We show that for every polynomial t, the distribution $G_t(\mathcal{H}, X_1, \ldots, X_t)$ is pseudorandom. Note that the input length of G_t is $|h| + n \cdot (t+1)$ and the output length is $|h| + t \cdot (n + \log n)$. By making $t = \Theta(n/\log n)$ calls, we show that G_t is indeed a pseudorandom generator.

Theorem 1.1. *[Main theorem for PRG, informal] Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be an unknown-regular one-way function and let $t(n) \geq n/\log n + 1$ be some polynomial. Then, G_t is a PRG with seed length $O(n^2 + n(t(n)+1))$. Furthermore, G_t makes $t(n)$ non-adaptive calls to f.*

A simple construction of UOWHFs from regular one-way functions. Now we introduce the construction of the UOWHFs. It is a well-known fact that in order to construct UWOHF, it is sufficient to construct a function for which it is hard to find a collision for a *random* input. Let f be a one-way function, let t

to use the one-way function, such construction needs at least $\Omega(n^2)$ input bits. We admit it is only a vague explanation.

[6] The assumption that f is length-preserving is made for simplicity, and is not crucial for our constructions.

be a parameter and let $\mathcal{H} = \left\{h : \{0,1\}^{2n} \to \{0,1\}^{n-\log n}\right\}$ be a family of hash functions. We define the function $C_t : \mathcal{H} \times \{0,1\}^{n\cdot t} \to \mathcal{H} \times \{0,1\}^{(t-1)\cdot(n-\log n)+2n}$ as

$$C_t\left(h, x_1, \ldots, x_t\right) = (h, f(x_1), h(x_1, f(x_2)), \ldots, h(x_{t-1}, f(x_t)), x_t)$$

The main difference of this construction from the PRG one is that h is now a shrinking function. In addition, we also output $f(x_1)$ and the very last input of C_t. As before, since the output length of UOWHFs has to be shorter than the input length, we have to make up for the additional output $(f(x_1), x_t)$ by taking t to be $\Theta(n/\log n)$.

The OUWHF can now be defined using C_t. Let $k = \log|\mathcal{H}| + n \cdot t$ and for a string $z \in \{0,1\}^k$, let C_z be the function defined by $C_z(w) = C_t(w \oplus z)$ for every $w \in \{0,1\}^k$. Our main theorem for this part is stated as follows.

Theorem 1.2. *[Main theorem for UOWHF, informal] Let* $f\colon \{0,1\}^n \to \{0,1\}^n$ *be an unknown-regular one-way function and let* $t(n) \geq n/\log n{+}2$ *be some polynomial. Then,* $\{C_z\}_{z\in\{0,1\}^k}$ *is a family of universal one-way hash functions with key length* $k = O(n^2 + n \cdot t(n))$ *and output length* $O(n^2 + n \cdot t(n))$*. Furthermore, for every* $z \in \{0,1\}^k$*,* C_z *makes* t *non-adaptive calls to* f*.*

1.2 Proof Overview

Here we give a short overview of our proofs. For both constructions, the proof boils down to showing that each input pair x_i, x_{i+1} induces a weak version of the desired primitive. For PRG, the main part of the security proof is showing that given $f(x_1)$ and h, it is hard to distinguish between $h(x_1, f(x_2))$ and a uniform string. For UOWHF, we prove the security by showing that given h, x_1, x_2, it is hard to find a collision h, x'_1, x'_2 to the function $C(h, x_1, x_2) = h, f(x_1), h(x_1, f(x_2))$. Note that it may be easy to find $x'_2 \neq x_2$ with $f(x'_2) = f(x_2)$. To solve this, we further demand that $f(x'_2) \neq f(x_2)$.[7] To show that this is enough, we prove that any collision in our UOWHF must contain a collision in the above form, for at least one input pair. Below we give short descriptions of the main ideas in more details.

The PRG construction. We start by sketching the security proof for the PRG. Let X_1 and X_2 be uniform random variables over $\{0,1\}^n$, and let h be a hash function, uniformly sampled from a universal family of hash functions $\mathcal{H} = \left\{h : \{0,1\}^{2n} \to \{0,1\}^{n+\log n}\right\}$. Recall that we want to show that given h and $f(X_1)$, it holds that $h(X_1, f(X_2))$ is computationally indistinguishable from uniform $n{+}\log n$ bits. For simplicity, assume that we are only interested in proving that the distinguish advantage is at most n^{-c}, for some constant $c > 1$.

[7] For this reason we need to output the last input x_t in our UOWHF construction.

The main observation is that for regular f, given $f(X_1)$, the pair $X_1, f(X_2)$ has exactly n bits of min-entropy. Thus, by the leftover hash lemma, the $n - O(c\log n)$ first bits of $h(X_1, f(X_2))$ are $n^{-c}/2$ statistically close to uniform. To argue that the suffix of $h(X_1, f(X_2))$ looks uniform, we show that $g(x_1, y) = h, f(x_1), h(x_1, y)_{1,\dots,n-O(c\log n)}$ is a one-way function,[8] and thus we can use Goldreich-Levin in order to extract additional $O(c\log n)$ pseudorandom bits from $X_1, f(X_2)$.

The UOWHF construction. We now sketch the security proof for the UOWHF. Let $\mathcal{H}$ be a universal family of hash functions $\left\{h : \{0,1\}^{2n} \to \{0,1\}^{n-\log n}\right\}$. We show that given random h and uniformly sampled x_1 and x_2 from $\{0,1\}^n$, it is hard to find $(x'_1, x'_2) \neq (x_1, x_2)$ such that $f(x_1) = f(x'_1)$, $f(x_2) \neq f(x'_2)$ and yet $h(x_1, f(x_2)) = h(x'_1, f(x'_2))$. For $x_1, x_2 \in \{0,1\}^n$ and $h \in \mathcal{H}$ we define

$$\mathcal{G}_{h,x_1,x_2} := \{(x'_1, y) \colon h(x_1, f(x_2)) = h(x'_1, y) \ \wedge \ f(x_1) = f(x'_1) \ \wedge \ y \in Im(f)\}.$$

That is, the set $\mathcal{G}_{h,x_1,x_2}$ contains all the pairs $(x'_1, f(x'_2))$ for which h, x'_1, x'_2 collides with h, x_1, x_2. The main observation here is that, since h outputs $n - \log n$ bits, and there are exactly 2^n pairs (x'_1, y) such that $y \in Im(f)$ and $f(x'_1) = f(x_1)$, the expected size of $\mathcal{G}_{h,x_1,x_2}$ is at most $2^n/2^{n-\log n} = n$. Thus, we can use an algorithm A that finds a collision in the above function in order to invert f: Given input y, we choose random $x_1, x_2 \in \{0,1\}^n$ and *plant* y in $\mathcal{G}_{h,x_1,x_2}$. That is, we choose a random h conditioned on the event that $h(x_1, f(x_2)) = h(x'_1, y)$ for some $x'_1 \in f^{-1}(f(x_1))$. Since there are about n such pairs, we can hope that the planted pair (x'_1, y) will be output by A with good probability.

However, we need to find x'_1 for which the pair (x'_1, y) has a good probability to be output by A. To do that, we also use A in order to find a pre-image x'_1 of $f(x_1)$, and then show that x'_1 has a good probability to be output again by A.[9] For more details, see Section 4.

1.3 Additional Related Work

Arbitrary one-way functions. In [12], the notion of inaccessible entropy (introduced in [14]) was used in order to construct UOWHF. Similar techniques were later used in [10] to construct PRG, where the notion of inaccessible entropy was replaced with next-block pseudoentropy. This construction was later simplified by [21], who also improved the seed length with the cost of adaptivity. Lately, [1] pointed out that the notions of accessible entropy and next-block pseudoentropy are deeply related to each other.

[8] Actually, we need to show that the function g is hard to invert on outputs sampled from a specific distribution. This is sufficient for applying the Goldreich-Levin theorem, see Lemma 2.5.

[9] Such a "collision based" argument was also used in [2].

Regular one-way functions. As mentioned above, the construction from regular one-way functions are more efficient. Beside almost-regular, a few refinements of regularity were considered in past works. [4] showed a construction for UOWHF that uses $O(ns^6(n))$ key-length under the assumption that $f^{-1}(f(x))$ is concentrated in an interval of size $2^{s(n)}$. [24] considered unknown-weakly-regular functions. The last are functions for which the set of inputs with maximal number of siblings is of fraction at least n^{-c} for some constant c. For such functions, [24] presented PRG with $O(n \log n)$ seed-length and $O(n^{2c+1})$ calls. [23] considered known-almost-regular and unknown-weakly-regular functions. For the last, [23] showed a tight construction of UOWHF based on the randomized iterate method.

1.4 Paper Organisation

Formal definitions are given in Section 2. The PRG construction and proof of Theorem 1.1 are in Section 3. The UOWHF construction and proof of Theorem 1.2 are in Section 4.

2 Preliminaries

2.1 Notations

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values and functions. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Given a vector $s \in \{0,1\}^n$, let s_i denote its i-th entry, and $s_{1,\ldots,i}$ denote its first i entries. For $s, w \in \{0,1\}^*$ we use $s \circ w$ to denote their concatenation and for $s, w \in \{0,1\}^n$, we use $s \oplus w \in \{0,1\}^n$ to denote their bit-wise XOR.

The support of a distribution P over a finite set $\mathcal{S}$ is defined by $\text{Supp}(P) := \{x \in \mathcal{S} : P(x) > 0\}$. For a (discrete) distribution D let $d \leftarrow D$ denote that d was sampled according to D. Similarly, for a set $\mathcal{S}$, let $s \leftarrow \mathcal{S}$ denote that s is drawn uniformly from $\mathcal{S}$. For a function $f\colon \{0,1\}^n \to \{0,1\}^n$, let $y \leftarrow f(\{0,1\}^n)$ denote that y sampled from the following distribution: sample x uniformly from $\{0,1\}^n$, and let $y = f(x)$. Let $\mathsf{Im}(f) := \{f(x)\colon x \in \{0,1\}^n\}$ be the image of f. The statistical distance (also known as, variation distance) of two distributions P and Q over a discrete domain $\mathcal{X}$ is defined by $\text{SD}(P,Q) := \max_{\mathcal{S} \subseteq \mathcal{X}} |P(\mathcal{S}) - Q(\mathcal{S})| = \frac{1}{2}\sum_{x \in \mathcal{S}} |P(x) - Q(x)|$. The min-entropy of a distribution X, denoted by $\text{H}_\infty(X)$ is defined by $\text{H}_\infty(X) := -\log(\max_{x \in \text{Supp}(X)} \{\Pr[X = x]\})$.

Let poly denote the set of all polynomials, and let PPT stand for probabilistic polynomial time. A function $\nu\colon \mathbb{N} \to [0,1]$ is negligible, denoted $\nu(n) = neg(n)$, if $\nu(n) < 1/p(n)$ for every $p \in$ poly and large enough n. Lastly, we identify a matrix $M \in \{0,1\}^{n \times m}$ with a function $M\colon \{0,1\}^n \to \{0,1\}^m$ by $M(x) := x \cdot M$, thinking of $x \in \{0,1\}^n$ as a vector with dimension n.

2.2 One-Way Functions

We now formally define basic cryptographic primitives. We start with the definition of one-way function.

Definition 2.1 (One-way function) *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* one-way function *if for every probabilistic polynomial time algorithm* A*, there is a negligible function $\nu : \mathbb{N} \to [0,1]$ such that for every $n \in \mathbb{N}$*

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(f(x)) \in f^{-1}(f(x))\right] \leq \nu(n)$$

For simplicity we assume that the one-way function f is length-preserving. That is, $|f(x)| = |x|$ for every $x \in \{0,1\}^*$. This can be assumed without loss of generality, and is not crucial for our constructions.

In this paper we focus on almost-regular one-way functions, formally defined below.

Definition 2.2 (Almost-regular function) *A function family $f = \{f_n \colon \{0,1\}^n \to \{0,1\}^n\}$ is* β-almost-regular *for $\beta \geq 0$ if for every $n \in \mathbb{N}$ and $x \in \{0,1\}^n$ it holds that*

$$\frac{2^n}{|\mathsf{Im}(f)\ |} \cdot n^{-\beta} \leq \left|f^{-1}(f(x))\right| \leq \frac{2^n}{|\mathsf{Im}(f)\ |} \cdot n^{\beta}.$$

f is almost-regular *if there exists $\beta \geq 0$ such that f is β-almost-regular, and* regular *if it is 0-almost-regular.*

Note that we do not assume that the regularity of f can be computed efficiently. That is, we only assume that f is unknow-(almost)-regular.

Immediately from the definition of a one-way function, we get the following simple observation.

Claim 2.3 *For every one-way function $f \colon \{0,1\}^n \to \{0,1\}^n$ there exists a negligible function $\nu(n)$ such that for every input $x \in \{0,1\}^n$ it holds that $\left|f^{-1}(f(x))\right| \leq 2^n \cdot \nu(n)$.*

2.3 Pseudorandom Generators

In Section 3 we use one-way functions in order to construct PRGs. The later are formally defined below.

Definition 2.4 (Pseudorandom generator) *Let n be a security parameter. A polynomial-time computable function $G : \{0,1\}^n \to \{0,1\}^{m(n)}$ is called a* pseudorandom generator *if for every $n > 0$ it holds that $m(n) > n$ and, for every probabilistic polynomial time algorithm* D*, there is a negligible function $\nu : \mathbb{N} \to [0,1]$ such that for every $n > 0$,*

$$\left|\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{D}(G(x)) = 1] - \Pr_{x \leftarrow \{0,1\}^{m(n)}}[\mathsf{D}(x) = 1]\right| \leq \nu(n).$$

A key ingredient in the construction of PRG from one-way function is the Goldreich-Levin hardcore predicate. The following lemma follows almost directly from [9].

Lemma 2.5 *Let n be a security parameter. Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a function, and D a distribution on $\{0,1\}^n$, such that for every* PPT A

$$\Pr_{x\leftarrow D}\left[\mathsf{A}(f(x)) \in f^{-1}(f(x))\right] = neg(n).$$

Then for every PPT P*,*

$$\Pr_{x\leftarrow D, r\leftarrow\{0,1\}^n}\left[\mathsf{P}(f(x),r) = \mathrm{GL}(x,r)\right] \le 1/2 + neg(n)$$

where $\mathrm{GL}(x,r) := \langle x,r\rangle$ *is the Goldreich-Levin predicate.*

Proof. By the proof of Goldreich-Levin [9], for every $p \in$ poly there is an oracle-aided PPT algorithm A such that for every algorithm P and x with

$$\Pr_{r\leftarrow\{0,1\}^n}\left[\mathsf{P}(f(x),r) = \mathrm{GL}(x,r)\right] \ge 1/2 + 1/p(n)$$

it holds that

$$\Pr\left[\mathsf{A}^{\mathsf{P}}(f(x)) = x\right] \ge 1/p^2(n).$$

Thus, it holds for every $p \in$ poly that

$$\Pr_{x\leftarrow D}\left[\Pr_{r\leftarrow\{0,1\}^n}\left[\mathsf{P}(f(x),r) = \mathrm{GL}(x,r)\right] \ge 1/2 + 1/p(n)\right] = neg(n)$$

which implies that

$$\Pr_{x\leftarrow D, r\leftarrow\{0,1\}^n}\left[\mathsf{P}(f(x),r) = \mathrm{GL}(x,r)\right] \le 1/2 + 1/p(n) + neg(n)$$

for every $p \in$ poly.

The next lemma, stated in [22], is useful for showing that a sequence of bits is pseudorandom. The proof of the lemma is given in Appendix A.

Lemma 2.6 (Distinguishability to prediction) *There exists an oracle-aided* PPT *algorithm* P *such that the following holds. Let Q be a distribution over $\{0,1\}^* \times \{0,1\}^n$, let* D *be an algorithm and $\alpha \in [0,1]$ such that,*

$$\Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}^n}\left[\mathsf{D}(x,z) = 1\right] - \Pr_{(x,y)\leftarrow Q}\left[\mathsf{D}(x,y) = 1\right] \ge \alpha.$$

Then there exists $i \in [n]$ such that

$$\Pr_{(x,y)\leftarrow Q}\left[\mathsf{P}^{\mathsf{D}}(x, y_{1,\dots,i-1}) = y_i\right] \ge 1/2 + \alpha/n.$$

2.4 Universal One Way Hash Function

Lastly, we formally define UOWHF.

Definition 2.7 (Universal one-way hash function) *Let k be a security parameter. A family of functions* $\mathcal{F} = \left\{ f_z \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^k}$ *is a family of* universal one-way hash functions *(UOWHFs) if it satisfies:*

1. *Efficiency: Given $z \in \{0,1\}^k$ and $x \in \{0,1\}^{n(k)}$, $f_z(x)$ can be evaluated in time* $\mathrm{poly}(n(k), k)$*.*
2. *Shrinking: $m(k) < n(k)$.*
3. *Target Collision Resistance: For every probabilistic polynomial-time adversary A, the probability that A succeeds in the following game is negligible in k:*
   (a) *Let $(x, state) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$.*
   (b) *Choose $z \leftarrow \{0,1\}^k$.*
   (c) *Let $x' \leftarrow A(state, z) \in \{0,1\}^{n(k)}$.*
   (d) *A succeeds if $x \neq x'$ and $f_z(x) = f_z(x')$.*

A relaxation of the target collision resistance property can be done by requiring the function to be collision resistant only on random inputs.

Definition 2.8 (Collision resistance on random inputs) *Let n be a security parameter. A function $f \colon \{0,1\}^n \to \{0,1\}^{m(n)}$ is* collision resistant on random inputs *if for every probabilistic polynomial-time adversary A, the probability that A succeeds in the following game is negligible in n:*

1. *Choose $x \leftarrow \{0,1\}^n$.*
2. *Let $x' \leftarrow A(x) \in \{0,1\}^n$.*
3. *A succeeds if $x \neq x'$ and $f(x) = f(x')$.*

The following lemma states that it is enough to construct a function that is collision resistant on random inputs, in order to get UOWHF.

Lemma 2.9 (From random inputs to targets, folklore) *Let n be a security parameter. Let $F \colon \{0,1\}^n \to \{0,1\}^{m(n)}$ be a length-decreasing function. Suppose F is collision-resistant on random inputs.*

Then $\{F_y \colon \{0,1\}^n \to \{0,1\}^m\}_{y \in \{0,1\}^n}$, for $F_y(x) := F(y \oplus x)$, is a family of target collision-resistant hash functions.

2.5 2-Universal Hash Families

2-universal families are an important ingredient in our constructions. In this section, we formally define this notion, together with some useful properties of such families.

Definition 2.10 (2-universal family) *A family of function $\mathcal{F} = \left\{ f \colon \{0,1\}^n \to \{0,1\}^\ell \right\}$ is* 2-universal *if for every $x \neq x' \in \{0,1\}^n$ it holds that $\Pr_{f \leftarrow \mathcal{F}}\left[f(x) = f(x')\right] = 2^{-\ell}$.*

A universal a family is explicit *if given a description of a function $f \in \mathcal{F}$ and $x \in \{0,1\}^n$, $f(x)$ can be computed in polynomial time (in n, ℓ). Such family is* constructible *if it is explicit and there is a* PPT *algorithm that given $x, x' \in \{0,1\}^n$ outputs a uniform $f \in \mathcal{F}$, such that $f(x) = f(x')$.*

An important property of 2-universal families is that they can be used to construct a strong extractor. This is stated in the leftover hash lemma:

Lemma 2.11 (Leftover hash lemma [18]) *Let $n \in \mathbb{N}$, $\epsilon \in [0,1]$, and let X be a random variable over $\{0,1\}^n$. Let $\mathcal{H} = \left\{ h \colon \{0,1\}^n \to \{0,1\}^\ell \right\}$ be a 2-universal hash family with $\ell \leq \mathrm{H}_\infty(X) - 2\log 1/\epsilon$. Then,*

$$SD((H, H(X)), (H, U_\ell)) \leq \epsilon$$

for U_ℓ being the uniform distribution over $\{0,1\}^\ell$ and H being the uniform distribution over $\mathcal{H}$.

The family of all binary matrices of size $n \times \ell$, $\left\{ m \colon m \in \{0,1\}^{n \times \ell} \right\}$, is a constructible 2-universal family. This family has an additional property that is useful in the proof. This property is defined below.

Definition 2.12 (Approximately flat family) *A family of functions $\mathcal{H} = \left\{ h \colon \{0,1\}^{2n} \to \{0,1\}^\ell \right\}$ is* approximately-flat *if for every set $\mathcal{Y} \subseteq \{0,1\}^n$, $x_1, x_2 \in \{0,1\}^n$ and $y_1 \in \mathcal{Y}$ it holds that,*

$$\Pr_{h \leftarrow \mathcal{H}}\left[\exists y_2 \in \mathcal{Y} \text{ s.t. } h(x_1, y_1) = h(x_2, y_2)\right] \geq 2^{-10} \cdot \min\left\{ |\mathcal{Y}| \cdot 2^{-\ell}, 1 \right\}.$$

The proof of the next lemma is in Appendix A.

Lemma 2.13 *For every $\ell, n \in \mathbb{N}$ such that $\ell \leq n$, the family $\left\{ m \colon m \in \{0,1\}^{n \times \ell} \right\}$ is approximately-flat.*

2.6 Useful Inequalities

The following well-known inequalities will be useful later on.

Lemma 2.14 (Jensen Inequality) *Let X be a distribution over $\mathbb{R}$ and let $f \colon \mathbb{R} \to \mathbb{R}$ be a convex function. It holds that*

$$f(\mathrm{E}\left[X\right]) \leq \mathrm{E}\left[f(X)\right]$$

Lemma 2.15 (Cauchy–Schwarz inequality) *Let $n \in \mathbb{N}$ and $a_1, \ldots, a_n \in \mathbb{R}$ be numbers. Then,*

$$(\sum_{i \in [n]} a_i)^2 \leq n \cdot \sum_{i \in [n]} a_i^2$$

Lastly, the following lemma will be useful in the security proof of the UOWHF. Let A be an algorithm such that for every x, the output of $\mathsf{A}(x)$ is in some small set $\mathcal{S}_x$. Then the lemma roughly states the event of two executions of A returning the same value is not too rare.

Lemma 2.16 *Let $\Omega \subseteq \{0,1\}^n$ and $\mathcal{X}$ be some set, let X be a distribution over $\mathcal{X}$, and let $S\colon \mathcal{X} \to P(\Omega)$ be a function that maps elements in $\mathcal{X}$ to subsets of Ω. Let A be an algorithm, such that for every $x \in \mathcal{X}$, $\mathsf{A}(x) \in S(x) \cup \{\perp\}$. Assume that for every $u \in \Omega$, it holds that $0 < \Pr_{x \leftarrow X}[u \in S(x)] \leq \ell/|\Omega|$, and that $\Pr_{x \leftarrow X}[\mathsf{A}(x) \in S(x)] \geq p$. Then*

$$\sum_{u \in \Omega} \Pr_{x \leftarrow X}[\mathsf{A}(x) = u] \Pr_{x \leftarrow X}[\mathsf{A}(x) = u \mid u \in S(x)] \geq p^2/\ell.$$

.

Proof. Using Cauchy–Schwarz inequality, it holds that:

$$\begin{aligned}
&\sum_{u \in \Omega} \Pr_{x \leftarrow X}[\mathsf{A}(x) = u] \Pr_{x \leftarrow X}[\mathsf{A}(x) = u \mid u \in S(x)] \\
&= \sum_{u \in \Omega} \Pr_{x \leftarrow X}[\mathsf{A}(x) = u]^2 / \Pr_{x \leftarrow X}[u \in S(x)] \\
&\geq \sum_{u \in \Omega} \Pr_{x \leftarrow X}[\mathsf{A}(x) = u]^2 \cdot |\Omega| / \ell \\
&\geq \left(\sum_{u \in \Omega} \Pr_{x \leftarrow X}[\mathsf{A}(x) = u] \right)^2 / \ell \\
&\geq p^2/\ell.
\end{aligned}$$

3 The PRG Construction

In this section we prove the security of our PRG construction. We start with a description of the construction. Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be an almost-regular one-way function, let t be a parameter and let $\mathcal{H} = \left\{m\colon m \in \{0,1\}^{2n \times (n+\log n)}\right\}$ be the 2-universal family induced by the set of matrices of size $2n \times (n + \log n)$.[10] The generator $G : \mathcal{H} \times \{0,1\}^{n(t+1)} \to \mathcal{H} \times \{0,1\}^{t\cdot(n+\log n)}$ is given by

$$G\big(h, x_1, \ldots, x_{t+1}\big) = (h, h(x_1, f(x_2)), \ldots, h(x_t, f(x_{t+1}))) .$$

The main theorem of this part is as follows.

[10] By taking $\mathcal{H} = \left\{h_m\colon m \in \{0,1\}^{2n\times(\log^2 n + \log n)}, h \in \mathcal{G}\right\}$ where $\mathcal{G} = \left\{g\colon \{0,1\}^{2n} \to \{0,1\}^{n-\log^2 n}\right\}$ is arbitrary 2-universal family, and $h_m(z) := h(z) \circ m(z)$, the seed of length can be reduced up to $O(n \cdot t)$.

Theorem 3.1. *[Main theorem for PRG] Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be an almost-regular one-way function and let $t(n) \geq n/\log n + 1$ be some polynomial. Then G is a PRG with seed length $O(n^2 + n(t+1))$. Furthermore, G uses t non-adaptive calls to f.*

Note that the stretch of G is $t \cdot \log n - n$, which is tight with [6] for large values of t. We now prove Theorem 3.1. Our main lemma states that given h and $f(x_1)$, the hash $h(x_1, f(x_2))$ looks uniform for a computationally bounded algorithm.

Lemma 3.2 *Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be an almost-regular one-way function. For any PPT algorithm D, it holds that*

$$\left| \Pr_{\substack{x_1 \leftarrow \{0,1\}^n, \\ h \leftarrow \mathcal{H}, \\ u \leftarrow \{0,1\}^{n+\log n}}} [\mathsf{D}(h, f(x_1), u) = 1] - \Pr_{\substack{x_1, x_2 \leftarrow \{0,1\}^n, \\ h \leftarrow \mathcal{H}}} [\mathsf{D}(h, f(x_1), h(x_1, f(x_2))) = 1] \right|$$

is a negligible function of n.

We prove Lemma 3.2 below, but first we use it in order to give the proof of Theorem 3.1, which is straight-forward.

Proof (Proof of Theorem 3.1). Let f and t be as in Theorem 3.1. By construction G makes t calls to f. Additionally, $t(n + \log n) > n(t+1)$ when $t \geq n/\log n + 1$. We are left to show that the output of G is indistinguishable from uniform. The proof is by a hybrid argument. Let H be a uniform random variable over $\mathcal{H}$, and $X_1, \ldots, X_{t+1}$ be i.i.d. uniform random variables over $\{0,1\}^n$. Assume toward a contradiction that there is a PPT algorithm $\widehat{\mathsf{D}}$ that can distinguish $G(H, X_1, \ldots, X_{t+1})$ from uniform. Then we show that the following algorithm D contradicts Lemma 3.2.

Algorithm 3.3 (The distinguisher D)

```
Input: $h \in \mathcal{H}, y \in \{0,1\}^n, z \in \{0,1\}^{n+\log n}$.
Operation:
  1. Sample $\ell \leftarrow [t]$.
  2. Sample $x_1, \ldots x_{\ell-1} \leftarrow (\{0,1\}^n)^{\ell-1}$ and $u \leftarrow \{0,1\}^{(t-\ell)n\log n}$.
  3. Compute $w := h, h(x_1, f(x_2)), \ldots, h(x_{\ell-2}, f(x_{\ell-1})), h(x_{\ell-1}, y), z, u$.
  4. Execute $\widehat{\mathsf{D}}(w)$ and output its output.
```

For each $\ell \in [t+1]$, let the distribution Hyb_ℓ be defined as

$$Hyb_\ell := \big(H, H(X_1, f(X_2)), \ldots, H(X_{\ell-1}, f(X_\ell)), U_{(t+1-\ell)n \cdot \log n}\big)$$

where $U_{(t+1-\ell)n \cdot \log n}$ is the uniform distribution over $\{0,1\}^{(t+1-\ell)n \cdot \log n}$. That is, Hyb_ℓ is equal to $G(H, X_1, \ldots, X_{t+1})$ on the first $\ell - 1$ blocks, and uniform on the rest. Observe that for every fixing of ℓ in the algorithm, the distribution of w for input $h \leftarrow \mathcal{H}, y \leftarrow f(U_n), z \leftarrow \{0,1\}^{n+\log n}$ is exactly as the distribution Hyb_ℓ. Similarly, the distribution of w for input $h \leftarrow \mathcal{H}, y \leftarrow f(U_n)$ and $z = h(X', Y')$ for $X' \leftarrow f^{-1}(y)$ and $Y' \leftarrow f(\{0,1\}^n)$ is exactly as the distribution $Hyb_{\ell+1}$. Thus, it holds that,

$$\left| \Pr_{\substack{x_1 \leftarrow \{0,1\}^n, \\ h \leftarrow \mathcal{H}, \\ u \leftarrow \{0,1\}^{n+\log n}}} [\mathsf{D}(h, f(x_1), u) = 1] - \Pr_{\substack{x_1, x_2 \leftarrow \{0,1\}^n, \\ h \leftarrow \mathcal{H}}} [\mathsf{D}(h, f(x_1), h(x_1, f(x_2))) = 1] \right| \tag{1}$$

$$= \left| 1/t \cdot \sum_{\ell=1}^{t} \left(\Pr_{w \leftarrow Hyb_\ell} \left[\widehat{\mathsf{D}}(w) = 1 \right] - \Pr_{w \leftarrow Hyb_{\ell+1}} \left[\widehat{\mathsf{D}}(w) = 1 \right] \right) \right|$$

$$= 1/t \cdot \left| \Pr_{w \leftarrow Hyb_1} \left[\widehat{\mathsf{D}}(w) = 1 \right] - \Pr_{w \leftarrow Hyb_{t+1}} \left[\widehat{\mathsf{D}}(w) = 1 \right] \right|$$

$$= 1/t \cdot \left| \Pr_{w \leftarrow \{0,1\}^{\log|\mathcal{H}| + (n+\log n)\cdot t}} \left[\widehat{\mathsf{D}}(w) = 1 \right] - \Pr_{w \leftarrow G(H, X_1, \ldots, X_{t+1})} \left[\widehat{\mathsf{D}}(w) = 1 \right] \right|.$$

Where the last equality holds since $Hyb_{t+1} \equiv G(H, X_1, \ldots, X_{t+1})$ and Hyb_1 is the uniform distribution. We conclude by Lemma 3.2 that the advantage probability of $\widehat{\mathsf{D}}$ is negligible.

3.1 Proving Lemma 3.2

In the rest of this section we prove Lemma 3.2. Fix $\beta \geq 0$, any β-almost-regular one-way function $f : \{0,1\}^n \to \{0,1\}^n$ and $n \in \mathbb{N}$. Recall that we want to show that $h(x_1, f(x_2))$ looks uniform to computationally bounded algorithms, given h and $f(x_1)$. By the leftover hash lemma, every prefix $p(x_1, x_2)$ of the above hash $h(x_1, f(x_2))$ is somewhat close to uniform. In order to show that the suffix looks uniform as well, we prove that the concatenation of $h, f(x_1)$ and $p(x_1, x_2)$ is a one-way function, and then use Goldreich-Levin. The next claim states that the described function is indeed one-way on part of its domain.

Claim 3.4 *For every $i \in [n+\log n]$, let $g_i \colon \mathcal{H} \times \{0,1\}^n \times \{0,1\}^n \to \mathcal{H} \times \{0,1\}^n \times \{0,1\}^{i-1}$ be the following function*

$$g_i(h, x_1, y) := (h, f(x_1), h(x_1, y)_{1,\ldots,i-1}) \,.$$

Then it holds that for every PPT A and every function $i = i(n)$

$$\Pr_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n \\ z = (h, x_1, f(x_2))}} \left[\mathsf{A}(g_i(z)) \in g_i^{-1}(g_i(z)) \right] = neg(n). \tag{2}$$

Proof. Assume toward contradiction that the claim does not hold. That is, there exists PPT algorithm A, a function $i(n)$ and a constant $d \in \mathbb{N}$ such that

$$\Pr_{\substack{h\leftarrow\mathcal{H},x_1,x_2\leftarrow\{0,1\}^n\\ z=(h,x_1,f(x_2))}}\left[\mathsf{A}(g_i(z)) \in g_i^{-1}(g_i(z))\right] \geq n^{-d} \tag{3}$$

for infinitely many $n \in \mathbb{N}$. Fix such n and consider the following algorithm $\widehat{\mathsf{A}}$. In the following we show $\widehat{\mathsf{A}}$ can be used to invert f.

Algorithm 3.5 (The inverter $\widehat{\mathsf{A}}$)

Input: $h \in \mathcal{H}, y \in \{0,1\}^n$, $z \in \{0,1\}^{n-(4d+2\beta)\log n}$.

Operation:

```
1. For every $w \in \{0,1\}^{(4d+2\beta+1)\log n}$ and $j \in [n+\log n]$:
   (a) Let $(h, x, y')$ be the output of $\mathsf{A}(h, y, (z \circ w)_{1,\ldots,j-1})$.
   (b) If $f(x) = y$, output $x$.
```

That is, $\widehat{\mathsf{A}}$ tries to invert y using A and only a prefix of $h(x_1, f(x_2))$. It does so by iterating over all the possible values of the missing input bits

$$h(f^{-1}(y), f(x_2))_{n-(4d+2\beta)\log n+1,\ldots,n+\log n}$$

and every possible index $j \in [n + \log n]$. Clearly $\widehat{\mathsf{A}}$ runs in a polynomial time. Let x_1 be some preimage of y and let x_2 be some element in $\{0,1\}^n$. Note that when the guess w is equal to $h(x_1, f(x_2))_{n-(4d+2\beta)\log n+1,\ldots,n+\log n}$, and when the index j is equal to i, the value of $h, y, (z \circ w)_{1,\ldots,j-1}$ computed by the algorithm is equal to the output of $g_i(h, x_1, f(x_2))$. Thus, by definition it is clear that the success probability of $\widehat{\mathsf{A}}$ is better than A's. Formally, we get that,

$$\begin{aligned}
&\Pr_{h\leftarrow\mathcal{H},x_1,x_2\leftarrow\{0,1\}^n}\left[\widehat{\mathsf{A}}(h, f(x_1), h(x_1, f(x_2))_{1,\ldots,n-(4d+2\beta)\log n}) \in f^{-1}(f(x_1))\right] \\
&\geq \Pr_{x_1,x_2\leftarrow\{0,1\}^n}\left[\mathsf{A}(g_i(h, x_1, f(x_2))) \in g_i^{-1}(g_i(h, x_1, f(x_2)))\right] \\
&\geq n^{-d}.
\end{aligned} \tag{4}$$

Next, we show that $\widehat{\mathsf{A}}$ can guess the value of $h(x_1, f(x_2))_{1,\ldots,n-(4d+2\beta)\log n}$. Indeed, recall that by the β-almost-regularity of f, given any fixing of $f(x_1)$, the min-entropy of $x_1, f(x_2)$ is at least $n - 2\beta\log n$. Thus, by the left-over hash lemma, $h(x_1, f(x_2))_{1,\ldots,n-(4d+2\beta)\log n}$ is $n^{-d}/2$ close to uniform given h

and $f(x_1)$. Let $k = n - (4d + 2\beta)\log n$. Combining the above with Equation (4),

$$\begin{aligned}
&\Pr_{h \leftarrow \mathcal{H}, x_1 \leftarrow \{0,1\}^n, u \leftarrow \{0,1\}^k}\left[\widehat{\mathsf{A}}(h, f(x_1), u) \in f^{-1}(f(x_1))\right] \qquad (5)\\
&= \mathop{\mathrm{E}}_{y \leftarrow f(\{0,1\}^n)}\left[\Pr_{\substack{h \leftarrow \mathcal{H}, x_1 \leftarrow f^{-1}(y),\\ u \leftarrow \{0,1\}^k}}\left[\widehat{\mathsf{A}}(h, y, u) \in f^{-1}(f(x_1))\right]\right]\\
&\geq \mathop{\mathrm{E}}_{y}\left[\Pr_{\substack{h \leftarrow \mathcal{H}, x_1 \leftarrow f^{-1}(y),\\ x_2 \leftarrow \{0,1\}^n}}\left[\widehat{\mathsf{A}}(h, y, h(x_1, f(x_2))_{1,\ldots,k}) \in f^{-1}(f(x_1))\right] - n^{-d}/2\right]\\
&= \Pr_{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n}\left[\widehat{\mathsf{A}}(h, f(x_1), h(x_1, f(x_2))_{1,\ldots,k}) \in f^{-1}(f(x_1))\right] - n^{-d}/2\\
&\geq n^{-d}/2.
\end{aligned}$$

Finally, let Inv be the algorithm that given $f(x_1)$ samples $h \leftarrow \mathcal{H}$ and $u \leftarrow \{0,1\}^{n-(4d+2\beta)\log n}$, and executes $\widehat{\mathsf{A}}$. By Equation (5) Inv inverts $f(x_1)$ successfully with probability at least $n^{-d}/2$ for uniformly sampled $x_1 \in \{0,1\}^n$, for infinitely many $n \in \mathbb{N}$, which is a contradiction.

We are now ready to prove Lemma 3.2. The proof is straight-forward from Claim 3.4 together with Lemma 2.5 and Lemma 2.6.

Proof (Proof of Lemma 3.2.). Assume toward a contradiction that Lemma 3.2 does not hold. That is, there exists PPT algorithm D and a constant $c \in \mathbb{N}$ such that

$$\left|\Pr_{\substack{x_1 \leftarrow \{0,1\}^n,\\ h \leftarrow \mathcal{H},\\ u \leftarrow \{0,1\}^{n+\log n}}}[\mathsf{D}(h, f(x_1), u) = 1] - \Pr_{\substack{x_1, x_2 \leftarrow \{0,1\}^n,\\ h \leftarrow \mathcal{H}}}[\mathsf{D}(h, f(x_1), h(x_1, f(x_2))) = 1]\right| \qquad (6)$$

$$\geq n^{-c}$$

for infinitely many $n \in \mathbb{N}$. We assume without loss of generality that for infinitely many $n \in \mathbb{N}$ it holds that

$$\Pr_{\substack{x_1 \leftarrow \{0,1\}^n,\\ h \leftarrow \mathcal{H},\\ u \leftarrow \{0,1\}^{n+\log n}}}[\mathsf{D}(h, f(x_1), u) = 1] - \Pr_{\substack{x_1, x_2 \leftarrow \{0,1\}^{2n},\\ h \leftarrow \mathcal{H}}}[\mathsf{D}(h, f(x_1), h(x_1, f(x_2))) = 1] \qquad (7)$$

$$\geq n^{-c}$$

as otherwise we can flip the output of D. By Lemma 2.6 there is a oracle-aided PPT algorithm P such that for infinitely many $n \in \mathbb{N}$ and $i = i(n)$ it holds that

$$\Pr_{\substack{x_1, x_2 \leftarrow \{0,1\}^{2n},\\ h \leftarrow \mathcal{H}}}\left[\mathsf{P}^{\mathsf{D}}(h, f(x_1), h(x_1, f(x_2))_{1,\ldots,i-1}) = h(x_1, f(x_2))_i\right] \geq 1/2 + n^{-c-4}.$$

Recall that, by definition, $h, f(x_1), h(x_1, f(x_2))_{1,\ldots,i-1} = g_i(x_1, f(x_2))$. Additionally, by our choice of the family $\mathcal{H}$, $h(x_1, f(x_2)))_i$ is the GL predicate of the function $g_i(x_1, f(x_2))$.[11] Thus, the above contradicts Claim 3.4 and lemma 2.5.

4 The UOWHF Construction

In this section we prove the security of our UOWHF construction. We start with a full description of the construction. Let $f\colon \{0,1\}^n \rightarrow \{0,1\}^n$ be an almost-regular one-way function, let t be a parameter and let $\mathcal{H} = \left\{m\colon m \in \{0,1\}^{2n\times(n-\log n)}\right\}$ be the 2-universal family induced by the set of matrices of size $2n \times (n - \log n)$.[12]

The function $C : \mathcal{H} \times \{0,1\}^{n\cdot t} \rightarrow \mathcal{H} \times \{0,1\}^{(t-1)\cdot(n-\log n)+2n}$ is given by

$$C\big(h, x_1, \ldots, x_t\big) = h, f(x_1), h(x_1, f(x_2)), \ldots, h(x_{t-1}, f(x_t)), x_t.$$

Let $k = \log|\mathcal{H}| + n\cdot t$. For a string $z \in \{0,1\}^k$, let $C_z(w) := C(w \oplus z)$. Our main theorem for this part is stated as follows.

Theorem 4.1. *[Main theorem for UOWHF] Let $f = f\colon \{0,1\}^n \rightarrow \{0,1\}^n$ be an almost-regular one-way function and let $t(n) \geq n/\log n + 2$ be some polynomial. Then $\mathcal{F}_k = \{C_z\}_{z\in\{0,1\}^k}$ is a family of universal one-way hash functions with key length $k = O(n^2 + n\cdot t(n))$ and output length $O(n^2 + n\cdot t(n))$. Furthermore, for every $z \in \{0,1\}^k$, C_z uses t non-adaptive calls to f.*

In the rest of this section we prove Theorem 4.1. Note that by Lemma 2.9 in order to prove Theorem 4.1, it is enough to show that it is hard to find a collision of C for a *random* input. The main lemma of this part is the following one, which essentially states that no efficient algorithm can find a collision in a simpler function, $\widehat{C}(h, x_1, x_2) = h, f(x_1), h(x_1, f(x_2))$. Note that $\widehat{C}$ is not UOWHF, as it is not shrinking, and, as we are only interested in collisions (h, x_1', x_2') in which $f(x_2) \neq f(x_2')$.

Lemma 4.2 *Let $f\colon \{0,1\}^n \rightarrow \{0,1\}^n$ be an almost-regular one-way function. For every PPT algortihm A, it holds that,*

$$\Pr_{\substack{h\leftarrow\mathcal{H}, x_1, x_2\leftarrow\{0,1\}^n,\\ (x_1', x_2')\leftarrow\mathsf{A}(h, x_1, x_2)}}\left[f(x_1) = f(x_1') \wedge f(x_2) \neq f(x_2') \wedge h(x_1, f(x_2)) = h(x_1', f(x_2'))\right]$$

is a negligible function of n.

[11] Note that if $i \leq n - \omega(\log n)$ there is no need in GL. Indeed, by the leftover hash lemma, the first bits of h are statisticly close to uniform.

[12] Any approximately-flat, constructible, and 2-universal hash family will suffice. Such a family with a smaller size, if exists, can be used in order to reduce the key length up to $O(n \cdot t)$.

We prove Lemma 4.2 below, but first let us prove the security of C using Lemma 4.2. The proof is by reduction, stated in the next claim. Informally, we show that an algorithm that breaks the security of C can be used in order to find a collision in the function $\widehat{C}$ defined above.

Claim 4.3 *There exists an oracle-aided* PPT *algorithm* A *such that the following holds. Let f be a one-way function, $t \in$ poly and C be the function described above. Let $n \in \mathbb{N}$, $\alpha \in [0,1]$ and let* $\mathsf{ColFinder}$ *be an algorithm such that*

$$\Pr_{w \leftarrow \mathcal{H} \times (\{0,1\}^n)^t, w' \leftarrow \mathsf{ColFinder}(w)} [w' \neq w \wedge C(w) = C(w')] = \alpha.$$

Then,

$$\Pr_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n, \\ (x_1', x_2') \leftarrow \mathsf{A}^{\mathsf{ColFinder}}(h, x_1, x_2)}} \left[\substack{f(x_1) = f(x_1') \\ \wedge f(x_2) \neq f(x_2') \wedge h(x_1, f(x_2)) = h(x_1', f(x_2'))} \right] \geq (\alpha - \nu(n))/t,$$

where ν is a negligible function, depending only on f and t.

The proof of Theorem 4.1 is now immediate.

Proof (Proof of Theorem 4.1.). Let f, t and C_z be as in Theorem 4.1. It is clear that C_z is efficiently computable for every $z \in \{0,1\}^k$, and that C is shrinking since $\log |H| + n \cdot t > \log |H| + (t-1) \cdot (n - \log n) + 2n$ for $t \geq n / \log n + 2$.

Next, we show that it is collision-resistant for random input. Assume toward contradiction that there exists a PPT $\mathsf{ColFinder}$ and $p \in$ poly such that

$$\Pr_{\substack{w \leftarrow \mathcal{H} \times (\{0,1\}^n)^t, \\ w' \leftarrow \mathsf{ColFinder}(w)}} [w' \neq w \wedge C(w) = C(w')] \geq 1/p(n)$$

for infinitely many $n \in \mathbb{N}$. Then, by Claim 4.3, for infinitely many $n \in \mathbb{N}$ it holds that

$$\Pr_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n, \\ (x_1', x_2') \leftarrow \mathsf{A}^{\mathsf{ColFinder}}(h, x_1, x_2)}} \left[\substack{f(x_1) = f(x_1') \wedge \\ f(x_2) \neq f(x_2') \wedge h(x_1, f(x_2)) = h(x_1', f(x_2'))} \right] \geq 1/(2t \cdot p(n)).$$

Note that by the choice of t, $1/(2t \cdot p(n))$ is not negligible, and that since both A and $\mathsf{ColFinder}$ are efficent, $\mathsf{A}^{\mathsf{ColFinder}}(\cdot)$ can be efficiently implemented. Thus, the above contradicts Lemma 4.2.

4.1 Proving Claim 4.3

We next prove Claim 4.3. The next simple claim will be useful in the proof, as it states that given $(h, x_1, \ldots, x_t)$, with high probability there is no collision $(h, x_1', \ldots, x_t')$ of C in which for some $j \in [t]$ it holds that $x_j \neq x_j'$ while $f(x_j) = f(x_j')$ and $f(x_{j+1}) = f(x_{j+1}')$.

Claim 4.4 *For every one-way function f and polynomial t, there exists a negligible function ν such that the following holds. For every $x_1, \ldots, x_t \in \{0,1\}^n$,*

$$\Pr_{h \leftarrow \mathcal{H}}\left[\substack{\forall j \in [t-1],\ \forall x'_j \in f^{-1}(f(x_j)) \setminus \{x_j\}\ \textit{it holds that} \\ h(x'_j, f(x_{j+1})) \neq h(x_j, f(x_{j+1}))}\right] \geq 1 - \nu(n).$$

Proof. Fix $x_1, \ldots, x_t \in \{0,1\}^n$, $j \in [t-1]$ and $x'_j \in f^{-1}(f(x_j)) \setminus \{x_j\}$. Since $\mathcal{H}$ is 2-universal, it holds that

$$\Pr_{h \leftarrow \mathcal{H}}\left[h(x'_j, f(x_{j+1})) = h(x_j, f(x_{j+1}))\right] = n/2^n.$$

By the union bound,

$$\begin{aligned}
&\Pr_{h \leftarrow \mathcal{H}}\left[\substack{\exists j \in [t-1], x'_j \in f^{-1}(f(x_j)) \setminus \{x_j\}\text{ s.t.} \\ h(x'_j, f(x_{j+1})) = h(x_j, f(x_{j+1}))}\right] \\
&\leq \sum_{j \in [t-1]} \sum_{x'_j \in f^{-1}(f(x_j)) \setminus \{x_j\}} \Pr_{h \leftarrow \mathcal{H}}\left[h(x'_j, f(x_{j+1})) = h(x_j, f(x_{j+1}))\right] \\
&\leq t(n) \cdot |f^{-1}(f(x_j))| \cdot n/2^n.
\end{aligned}$$

Since f is a one-way function, by Claim 2.3 it holds that $|f^{-1}(f(x_k))| \leq 2^n \cdot neg(n)$, and thus the claim follows.

Proof (Proof of Claim 4.3.). Let f, t n, α and $\mathsf{ColFinder}$ as in Claim 4.3. Let A be the following algorithm.

```
Algorithm 4.5 (The reduction A)
Input: $h \in \mathcal{H}$, $x_1, x_2 \in \{0,1\}$.
Oracle: ColFinder.
Operation:

  1. Sample $i \leftarrow [t-1]$, $z_1, \ldots, z_{i-1}, z_{i+2}, \ldots, z_t \leftarrow \{0,1\}^n$ and set
     $z_i = x_1, z_{i+1} = x_2$.
  2. Apply $\mathsf{ColFinder}(h, z_1, \ldots, z_t)$ to get $(h', z'_1, \ldots, z'_t)$.
  3. Output $z'_i, z'_{i+1}$.
```

We next show that with all but negligible probability over the choice of $w = (h, x_1, \ldots, x_t)$, the following must hold. For every $w' = (h', x'_1, \ldots, x'_t)$ with $w \neq w'$ and $C(w) = C(w')$, there exists some $i \in [t-1]$ such that $f(x_i) = f(x'_i)$ and $f(x_{i+1}) \neq f(x'_{i+1})$. The lemma then follows easily.

Indeed, fix such w and w'. First note that since $C(w) = C(w')$, it holds that $h = h'$. Let j be the first index for which $x_j \neq x'_j$, and observe that by the definition of C, $j \in [t-1]$. We split into cases:

- If $f(x_j) \neq f(x'_j)$, then $j > 1$ (since $C(w) = C(w')$ implies that $f(x_1) = f(x'_1)$) and for $i = j-1$ it holds that $f(x_i) = f(x'_i)$ and $f(x_{i+1}) \neq f(x'_{i+1})$.

- For the other case, assume that $f(x_j) = f(x'_j)$. By Claim 4.4, with probability all but negligible over the choice of w, it holds that, $h(x_j, f(x_{j+1})) \neq h(x'_j, f(x_{j+1}))$, and thus it must hold that $f(x_{j+1}) \neq f(x'_{j+1})$. We get that for $i = j$, it holds that $f(x_i) = f(x'_i)$ and $f(x_{i+1}) \neq f(x'_{i+1})$.

Since i is chosen uniformly in Theorem 4.5, and since the distribution of $h, z_1, \ldots, z_t$ in Theorem 4.5 is uniform for every $i \in [t-1]$ and uniformly chosen input h, x_1, x_2, we conclude that the success probability of $\mathsf{A}^{\mathsf{ColFinder}}$ is at least $(\alpha - neg(n))/t$.

4.2 Proving Lemma 4.2

We now prove Lemma 4.2. For the rest of this section, fix $\beta \geq 0$, and a β-almost-regular one-way function f. In order to prove the lemma, we show how to invert the one-way function f using an algorithm that contradicts the lemma. Formally,

Claim 4.6 *There exists* PPT *oracle-aided algorithm* Inv *such that the following holds. Let $n \in \mathbb{N}$, $\alpha \in [0,1]$ and let* A *be an algorithm such that*

$$\Pr_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n, \\ (x'_1, x'_2) \leftarrow \mathsf{A}(h, x_1, x_2)}} \left[\substack{f(x_1) = f(x'_1) \wedge \\ f(x_2) \neq f(x'_2) \wedge h(x_1, f(x_2)) = h(x'_1, f(x'_2))} \right] = \alpha.$$

Then,

$$\Pr_{x \leftarrow \{0,1\}} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x)) \right] \geq \alpha^2 \cdot n^{-2\beta - 2} \cdot 2^{-12}.$$

The proof of Lemma 4.2 is immediate from Claim 4.6, as $\Pr_{x \leftarrow \{0,1\}} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x)) \right]$ must be negligible.

Proof (Proof of Lemma 4.2.). Assume toward contradiction that there exists a PPT algorithm A and $p \in$ poly such that

$$\Pr_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n, \\ (x'_1, x'_2) \leftarrow \mathsf{A}(h, x_1, x_2)}} \left[\substack{f(x_1) = f(x'_1) \wedge \\ f(x_2) \neq f(x'_2) \wedge h(x_1, f(x_2)) = h(x'_1, f(x'_2))} \right] \geq 1/p(n)$$

for infinitely many $n \in \mathbb{N}$. Then, by Claim 4.6 it holds that

$$\Pr_{x \leftarrow \{0,1\}} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x)) \right] \geq 1/p(n)^2 \cdot n^{-2\beta - 2} \cdot 2^{-10}$$

for infinitely many $n \in \mathbb{N}$, which is a contradiction to f being a one-way function.

The rest of this part is dedicated for proving Claim 4.6. Let n, α and A be as in Claim 4.6. In the following we assume that A outputs a valid pair (x'_1, x'_2) with $(f(x_1) = f(x'_1) \wedge f(x_2) \neq f(x'_2) \wedge h(x_1, f(x_2)) = h(x'_1, f(x'_2)))$ or $(\perp, \perp)$. For x_1, x_2 and h, we define,

$$\mathcal{G}_{h, x_1, x_2} := \left\{ (x'_1, y) \in f^{-1}(f(x_1)) \times \mathsf{Im}(f) \, : h(x_1, f(x_2)) = h(x'_1, y) \right\}.$$

For ease of notation, we say that $x \in \mathcal{G}_{h,x_1,x_2}$ if there exists $y \in \mathsf{Im}(f)$ such that $(x, y) \in \mathcal{G}_{h,x_1,x_2}$. Let Inv be the following algorithm. Note that Inv can be implemented efficiently, by the constructibility of $\mathcal{H}$.

```
Algorithm 4.7 (The inverter Inv)
Input: $y \in \mathsf{Im}(f)$ .
Oracle: $\mathsf{A}$.
Operation:

  1. Sample $x_1, x_2 \leftarrow \{0,1\}^n$ and $h \leftarrow \mathcal{H}$.
  2. Apply $\mathsf{A}(h, x_1, x_2)$ to get $(x_1', x_2')$. If $\mathsf{A}$ outputs $(\perp, \perp)$, output $\perp$.
  3. Sample $h' \leftarrow \mathcal{H}$ such that $h'(x_1, f(x_2)) = h'(x_1', y)$.
  4. Apply $\mathsf{A}(h', x_1, x_2)$ to get $(x_1'', x)$. Output $x$.
```

That is, in order to invert its input y, Inv samples x_1, x_2 and h. It then uses A in order to find x_1' with $f(x_1') = f(x_1)$. Lastly, it samples h' with $h'(x_1, f(x_2)) = h'(x_1', y)$ and uses A in order to find a collision to h', x_1, x_2. By the choice of h', a possible collision is $(h', x_1', f^{-1}(y))$. We observe that if A finds such a collision, Inv successfully inverted y.

For $x_1, x_2 \in \{0,1\}^n$, $x_1' \in f^{-1}(f(x))$ and $y \in \mathsf{Im}(f)$, let

$$\begin{aligned} & p_{\mathsf{A}}(x_1, x_2, x_1', y) \\ & := \Pr_{h' \leftarrow \mathcal{H}} \left[\mathsf{A}(h', x_1, x_2) \in \{x_1'\} \times f^{-1}(y) \mid h'(x_1, f(x_2)) = h'(x_1', y)\right] \\ & = \Pr_{h' \leftarrow \mathcal{H}} \left[\mathsf{A}(h', x_1, x_2) \in \{x_1'\} \times f^{-1}(y) \mid (x_1', y) \in \mathcal{G}_{h', x_1, x_2}\right] \end{aligned}$$

and define $p_{\mathsf{A}}(x_1, x_2, \perp, y) = 0$. By the above observation, it holds that

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x))\right] \geq \mathop{\mathrm{E}}_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n \\ y \leftarrow f(\{0,1\}^n) \\ (x_1', x_2') \leftarrow A(h, x_1, x_2)}} \left[p_{\mathsf{A}}(x_1, x_2, x_1', y)\right] \qquad (8)$$

and thus it is enough to bound the latter. We bound it using the following two claims. The first shows that it is enough to bound the probability that A outputs $(x_1', \cdot)$. The second claim bounds the last probability.

Claim 4.8 *For every $x_1, x_2 \in \{0,1\}^n$ and $x' \in f^{-1}(f(x_1))$ the following holds.*

$$\begin{aligned} & \mathop{\mathrm{E}}_{y \leftarrow f(\{0,1\}^n)} \left[p_{\mathsf{A}}(x_1, x_2, x', y)\right] \\ & \geq \Pr_{h' \leftarrow \mathcal{H}} \left[\mathsf{A}(h', x_1, x_2) = (x', \cdot) \mid x' \in \mathcal{G}_{h', x_1, x_2}\right] \cdot n^{-\beta - 1} \cdot 2^{-10}. \end{aligned}$$

Proof. Fix $x_1, x_2 \in \{0,1\}^n$ and $x' \in f^{-1}(f(x_1))$, and for every $h \in \mathcal{H}$, let $\mathsf{A}(h) := \mathsf{A}(h, x_1, x_2)$ and $\mathcal{G}_h := \mathcal{G}_{h,x_1,x_2}$. Then, by the definition of p_{A}, it holds that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[p_{\mathsf{A}}(x_1,x_2,x',y)\right]\\
&=\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid (x',y)\in\mathcal{G}_{h'}\right]\right]\\
&=\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[\frac{\Pr_{h'\leftarrow\mathcal{H}}\left[(x',y)\in\mathcal{G}_{h'}\wedge\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]}{\Pr_{h'\leftarrow\mathcal{H}}\left[(x',y)\in\mathcal{G}_{h'}\mid x'\in\mathcal{G}_{h'}\right]}\right]\\
&=\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[\frac{\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]}{\Pr_{h'\leftarrow\mathcal{H}}\left[(x',y)\in\mathcal{G}_{h'}\mid x'\in\mathcal{G}_{h'}\right]}\right]\\
&=\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]\cdot\frac{\Pr_{h'\leftarrow\mathcal{H}}\left[x'\in\mathcal{G}_{h'}\right]}{\Pr_{h'\leftarrow\mathcal{H}}\left[(x',y)\in\mathcal{G}_{h'}\right]}\right]
\end{aligned}$$

Since by our assumption on A, for every (x',y) with $\Pr\left[\mathsf{A}(h)\in\{x'\}\times f^{-1}(y)\right]>0$ it holds that $(x',y)\neq(x_1,f(x_2))$, we get that for every such pair $\Pr_{h'\leftarrow\mathcal{H}}\left[(x',y)\in\mathcal{G}_{h'}\right]=n/2^n$. Continue,

$$\begin{aligned}
&\mathop{\mathrm{E}}_{y\leftarrow f(\{0,1\}^n)}\left[p_{\mathsf{A}}(x_1,x_2,x',y)\right]\\
&=\sum_{y\in\mathsf{Im}(f)}\Pr_{x\leftarrow\{0,1\}^n}\left[f(x)=y\right]\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]\\
&\qquad\cdot\frac{2^n}{n}\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[x'\in\mathcal{G}_{h'}\right]\\
&\geq\sum_{y\in\mathsf{Im}(f)}\frac{1}{|\mathsf{Im}(f)\,|\cdot n^{\beta}}\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]\\
&\qquad\cdot\frac{2^n}{n}\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[x'\in\mathcal{G}_{h'}\right]\\
&=\frac{1}{|\mathsf{Im}(f)\,|\cdot n^{\beta}}\cdot\frac{2^n}{n}\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[x'\in\mathcal{G}_{h'}\right]\\
&\qquad\cdot\sum_{y\in\mathsf{Im}(f)}\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')\in\{x'\}\times f^{-1}(y)\mid x'\in\mathcal{G}_{h'}\right]\\
&=\frac{2^n}{|\mathsf{Im}(f)\,|\cdot n^{\beta+1}}\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[x'\in\mathcal{G}_{h'}\right]\cdot\Pr_{h'\leftarrow\mathcal{H}}\left[\mathsf{A}(h')=(x',\cdot)\mid x'\in\mathcal{G}_{h'}\right]
\end{aligned}$$

where the inequality holds since f is β-almost-regular. Recall that the family $\mathcal{H}$ is approximately-flat. That is,

$$\begin{aligned}
&\Pr_{h'\leftarrow\mathcal{H}}\left[\exists y\in\mathsf{Im}(f)\ \text{ s.t. } h'(x_1,f(x_2))=h'(x',y)\right]\\
&\geq 2^{-10}\cdot\min\left\{|\mathsf{Im}(f)\,|\cdot 2^{-(n-\log n)},1\right\}.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\frac{2^n}{|\mathsf{Im}(f)\ | \cdot n^{\beta+1}} \cdot \Pr_{h'\leftarrow\mathcal{H}}[x' \in \mathcal{G}_{h'}] \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h') = (x',\cdot) \mid x' \in \mathcal{G}_{h'}]\\
&\geq \frac{2^n}{|\mathsf{Im}(f)\ | \cdot n^{\beta+1}} \cdot 2^{-10} \cdot \min\left\{|\mathsf{Im}(f)\ | \cdot 2^{-(n-\log n)}, 1\right\}\\
&\qquad\qquad \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h') = (x',\cdot) \mid x' \in \mathcal{G}_{h'}]\\
&\geq n^{-\beta-1} \cdot 2^{-10} \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h') = (x',\cdot) \mid x' \in \mathcal{G}_{h'}]
\end{aligned}$$

and the claim holds.

The next claim uses Lemma 2.16 in order to show that in a random execution of Inv, A has a good probability to output the same element x'_1 in Items 2 and 4.

Claim 4.9 *For every $x_1, x_2 \in \{0,1\}$ the following holds. Let $\alpha_{x_1,x_2} := \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) \neq \perp]$. Then,*

$$\begin{aligned}
&\sum_{x'_1\in f^{-1}(f(x_1))} \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x'_1,\cdot)]\\
&\qquad\qquad \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h',x_1,x_2) = (x'_1,\cdot) \mid x'_1 \in \mathcal{G}_{h',x_1,x_2}]\\
&\geq \alpha^2_{x_1,x_2} \cdot n^{-\beta-1}/4.
\end{aligned}$$

Proof. Fix $x_1, x_2 \in \{0,1\}^n$, and let α_{x_1,x_2} be as in Claim 4.9. Let $\alpha_1 := \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x_1,\cdot)]$ and let $\alpha_2 := \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) \notin \{(x_1,\cdot),\perp\}]$. Notice that $\alpha_{x_1,x_2} = \alpha_1 + \alpha_2$.

Define $\widetilde{\mathsf{A}}(h)$ to be the algorithm that outputs the first coordinate of A's output $(\mathsf{A}(h,x_1,x_2)_1)$ if it is different from x_1, or $\perp$ otherwise. Let $\mathcal{G}_h := \mathcal{G}_{h,x_1,x_2}$. Note that by the assumption on A, $\widetilde{\mathsf{A}}$ always outputs elements in $S(h) = \{x \in \mathcal{G}_{h,x_1,x_2} \colon x \neq x_1\}$. We get that $\alpha_2 := \Pr_{h\leftarrow\mathcal{H}}\left[\widetilde{\mathsf{A}}(h) \neq \perp\right]$. Let $\Omega = f^{-1}(f(x_1)) \setminus \{x_1\}$. It holds that,

$$\begin{aligned}
&\sum_{x'_1\in f^{-1}(f(x_1))} \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x'_1,\cdot)]\\
&\qquad \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h',x_1,x_2) = (x'_1,\cdot) \mid x'_1 \in \mathcal{G}_{h',x_1,x_2}]\\
&= \sum_{x'_1\in\Omega} \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x'_1,\cdot)] \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h',x_1,x_2) = (x'_1,\cdot) \mid x'_1 \in \mathcal{G}_{h',x_1,x_2}]\\
&\qquad + \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x_1,\cdot)] \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h',x_1,x_2) = (x_1,\cdot) \mid x_1 \in \mathcal{G}_{h',x_1,x_2}]\\
&= \sum_{x'_1\in\Omega} \Pr_{h\leftarrow\mathcal{H}}\left[\widetilde{\mathsf{A}}(h) = x'_1\right] \cdot \Pr_{h'\leftarrow\mathcal{H}}\left[\widetilde{\mathsf{A}}(h) = x'_1 \mid x'_1 \in \mathcal{G}_{h',x_1,x_2}\right]\\
&\qquad + \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}(h,x_1,x_2) = (x_1,\cdot)] \cdot \Pr_{h'\leftarrow\mathcal{H}}[\mathsf{A}(h',x_1,x_2) = (x_1,\cdot)]\\
&= \sum_{x'_1\in\Omega} \Pr_{h\leftarrow\mathcal{H}}\left[\widetilde{\mathsf{A}}(h) = x'_1\right] \cdot \Pr_{h'\leftarrow\mathcal{H}}\left[\widetilde{\mathsf{A}}(h) = x'_1 \mid x'_1 \in S(h')\right] + \alpha_1^2,
\end{aligned}$$

where the second equality holds by definition of $\widetilde{\mathsf{A}}$ and since x_1 is always a member in $\mathcal{G}_{h,x_1,x_2}$. We next show that

$$\sum_{x_1' \in \Omega} \Pr_{h \leftarrow \mathcal{H}} \left[\widetilde{\mathsf{A}}(h) = x_1' \right] \cdot \Pr_{h' \leftarrow \mathcal{H}} \left[\widetilde{\mathsf{A}}(h) = x_1' \mid x_1' \in S(h') \right] \geq \alpha_2^2 \cdot n^{-\beta-1}. \qquad (9)$$

Indeed, assume that Ω is not empty, as otherwise the above holds trivially. We observe that for every $x \in \Omega$,

$$0 < \Pr_{h' \leftarrow \mathcal{H}} [x \in S(h')] \leq |\mathsf{Im}(f)\ | \cdot n/2^n \leq n^{\beta+1} / \left| f^{-1}(f(x)) \right| \leq n^{\beta+1} / |\Omega|. \quad (10)$$

Thus we can use Lemma 2.16, with $\mathcal{X} = \mathcal{H}$ in order to get Equation (9).

Combining the above, we conclude that

$$\begin{aligned} &\sum_{x_1' \in f^{-1}(f(x_1))} \Pr_{h \leftarrow \mathcal{H}} [\mathsf{A}(h, x_1, x_2) = (x_1', \cdot)] \\ &\qquad\qquad \cdot \Pr_{h' \leftarrow \mathcal{H}} [\mathsf{A}(h', x_1, x_2) = (x_1', \cdot) \mid x_1' \in \mathcal{G}_{h', x_1, x_2}] \\ &\geq \alpha_2^2 \cdot n^{-\beta-1} + \alpha_1^2. \end{aligned}$$

The claim follows since either α_1 or α_2 is at least $\alpha_{x_1,x_2}/2$.

We are now ready to prove Claim 4.6.

Proof (Proof of Claim 4.6). For fixed x_1 and x_2 let α_{x_1,x_2} be as in Claim 4.9. We start by showing that

$$\Pr_{x \leftarrow \{0,1\}} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x)) \right] \geq \mathop{\mathrm{E}}_{x_1, x_2 \leftarrow \{0,1\}^n} \left[\alpha_{x_1,x_2}^2 \right] \cdot n^{-2\beta-2} \cdot 2^{-12}. \quad (11)$$

Indeed, by Equation (8),

$$\begin{aligned} \Pr_{x \leftarrow \{0,1\}} \left[\mathsf{Inv}^{\mathsf{A}}(f(x)) \in f^{-1}(f(x)) \right] &\geq \mathop{\mathrm{E}}_{\substack{h \leftarrow \mathcal{H}, x_1, x_2 \leftarrow \{0,1\}^n \\ y \leftarrow f(\{0,1\}^n) \\ (x_1', x_2') \leftarrow A(h, x_1, x_2)}} [p_{\mathsf{A}}(x_1, x_2, x_1', y)] \\ &= \mathop{\mathrm{E}}_{x_1, x_2 \leftarrow \{0,1\}^n} \left[\mathop{\mathrm{E}}_{\substack{h \leftarrow \mathcal{H}, y \leftarrow f(\{0,1\}^n), \\ (x_1', x_2') \leftarrow A(h, x_1, x_2)}} [p_{\mathsf{A}}(x_1, x_2, x_1', y)] \right], \end{aligned}$$

and thus it is enough to show that for every fixed $x_1, x_2 \in \{0,1\}^n$,

$$\mathop{\mathrm{E}}_{\substack{h \leftarrow \mathcal{H}, y \leftarrow f(\{0,1\}^n), \\ (x_1', x_2') \leftarrow A(h, x_1, x_2)}} [p_{\mathsf{A}}(x_1, x_2, x_1', y)] \geq \alpha_{x_1,x_2}^2 \cdot n^{-2\beta-2} \cdot 2^{-12}.$$

Indeed, recall that by definition, $p_{\mathsf{A}}(x_1, x_2, \perp, y) = 0$. Therefore,

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\substack{h \leftarrow \mathcal{H}, y \leftarrow f(\{0,1\}^n),\\ (x_1', x_2') \leftarrow A(h, x_1, x_2)}} [p_{\mathsf{A}}(x_1, x_2, x_1', y)] \\
&= \sum_{x_1' \in f^{-1}(f(x_1))} \Pr_{h \leftarrow \mathcal{H}} [A(h, x_1, x_2) = (x_1', \cdot)] \cdot \mathop{\mathrm{E}}_{y \leftarrow f(\{0,1\}^n)} [p_{\mathsf{A}}(x_1, x_2, x_1', y)] \\
&\geq \sum_{x_1' \in f^{-1}(f(x_1))} \Pr_{h \leftarrow \mathcal{H}} [A(h, x_1, x_2) = (x_1', \cdot)] \\
&\qquad \cdot \Pr_{h' \leftarrow \mathcal{H}} [A(h', x_1, x_2) = (x_1', \cdot) \mid x_1' \in \mathcal{G}_{h', x_1, x_2}] \cdot n^{-\beta - 1} \cdot 2^{-10} \\
&\geq \alpha_{x_1, x_2}^2 \cdot n^{-2\beta - 2} \cdot 2^{-12}.
\end{aligned}$$

Where the equality holds by the assumption that A always outputs a valid collision, or $\perp$. The first inequality holds by Claim 4.8 and the second by Claim 4.9.

We are now left to bound $\mathrm{E}_{x_1, x_2 \leftarrow \{0,1\}^n} \left[\alpha_{x_1, x_2}^2\right] \cdot n^{-2\beta-2} \cdot 2^{-12}$. Observe that by definition $\mathrm{E}_{x_1, x_2 \leftarrow \{0,1\}^n} [\alpha_{x_1, x_2}] = \alpha$, and thus by the Jensen inequality, it holds that $\mathrm{E}_{x_1, x_2 \leftarrow \{0,1\}^n} \left[\alpha_{x_1, x_2}^2\right] \geq \alpha^2$, which concludes the proof.

Acknowledgement

We are thankful to Iftach Haitner and Salil Vadhan for very useful discussions. We also thank the anonymous reviewers for their comments.

References

1. Agrawal, R., Chen, Y.H., Horel, T., Vadhan, S.: Unifying computational entropies via kullback–leibler divergence. In: Annual International Cryptology Conference. pp. 831–858. Springer (2019)
2. Ames, S., Gennaro, R., Venkitasubramaniam, M.: The generalized randomized iterate and its application to new efficient constructions of uowhfs from regular one-way functions. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 154–171. Springer (2012)
3. Barhum, K., Holenstein, T.: A cookbook for black-box separations and a recipe for uowhfs. In: Theory of Cryptography Conference. pp. 662–679. Springer (2013)
4. Barhum, K., Maurer, U.: Uowhfs from owfs: Trading regularity for efficiency. In: International Conference on Cryptology and Information Security in Latin America. pp. 234–253. Springer (2012)
5. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM journal on Computing **13**(4), 850–864 (1984)
6. Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM journal on Computing **35**(1), 217–246 (2005)
7. Goldreich, O., Impagliazzo, R., Levin, L., Venkatesan, R., Zuckerman, D.: Security preserving amplification of hardness. In: Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science. pp. 318–326. IEEE (1990)
8. Goldreich, O., Krawczyk, H., Luby, M.: On the existence of pseudorandom generators. SIAM Journal on Computing **22**(6), 1163–1175 (1993)

9. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing. pp. 25–32 (1989)
10. Haitner, I., Harnik, D., Reingold, O.: Efficient pseudorandom generators from exponentially hard one-way functions. In: International Colloquium on Automata, Languages, and Programming. pp. 228–239. Springer (2006)
11. Haitner, I., Harnik, D., Reingold, O.: On the power of the randomized iterate. In: Annual International Cryptology Conference. pp. 22–40. Springer (2006)
12. Haitner, I., Holenstein, T., Reingold, O., Vadhan, S., Wee, H.: Universal one-way hash functions via inaccessible entropy. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 616–637. Springer (2010)
13. Haitner, I., Reingold, O., Vadhan, S.: Efficiency improvements in constructing pseudorandom generators from one-way functions. SIAM Journal on Computing **42**(3), 1405–1430 (2013)
14. Haitner, I., Reingold, O., Vadhan, S., Wee, H.: Inaccessible entropy. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 611–620 (2009)
15. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
16. Holenstein, T.: Pseudorandom generators from one-way functions: A simple construction for any hardness. In: Theory of Cryptography Conference. pp. 443–461. Springer (2006)
17. Holenstein, T., Sinha, M.: Constructing a pseudorandom generator requires an almost linear number of calls. In: 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science. pp. 698–707. IEEE (2012)
18. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing. pp. 12–24 (1989)
19. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing. pp. 33–43 (1989)
20. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. pp. 387–394 (1990)
21. Vadhan, S., Zheng, C.J.: Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In: Proceedings of the forty-fourth annual ACM symposium on Theory of computing. pp. 817–836 (2012)
22. Yao, A.C.: Theory and application of trapdoor functions. In: 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982). pp. 80–91. IEEE (1982)
23. Yu, Y., Gu, D., Li, X., Weng, J.: (almost) optimal constructions of uowhfs from 1-to-1, regular one-way functions and beyond. In: Annual Cryptology Conference. pp. 209–229. Springer (2015)
24. Yu, Y., Gu, D., Li, X., Weng, J.: The randomized iterate, revisited-almost linear seed length prgs from a broader class of one-way functions. In: Theory of Cryptography Conference. pp. 7–35. Springer (2015)
25. Yu, Y., Li, X., Weng, J.: Pseudorandom generators from regular one-way functions: New constructions with improved parameters. Theoretical Computer Science **569**, 58–69 (2015)

A Missing Proofs

A.1 Pseudorandom Generator

Lemma A.1 (Lemma 2.6, restated) *There exists a* PPT *algorithm* P *such that the following holds. Let* Q *be a distribution over* $\{0,1\}^* \times \{0,1\}^n$*, and let* D *be an algorithm and* $\alpha \in [0,1]$ *such that,*

$$\Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}^n}[\mathsf{D}(x,z)=1] - \Pr_{(x,y)\leftarrow Q}[\mathsf{D}(x,y)=1] \geq \alpha.$$

Then there exists $i \in [n]$ *such that*

$$\Pr_{(x,y)\leftarrow Q}\left[\mathsf{P}^{\mathsf{D}}(x, y_{1,\ldots,i-1}) = y_i\right] \geq 1/2 + \alpha/n.$$

Proof (Proof of Lemma 2.6.). Let Q, D and α be as in Lemma 2.6. We start by showing that D can be used in order to distinguish y_i from uniform bit given $x, y_{1,\ldots,i-1}$ for some index $i \in [n]$. Later we use this fact in order to predict y_i. Indeed, it holds that

$$\begin{aligned}\alpha &\leq \Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}^n}[\mathsf{D}(x,z)=1] - \Pr_{(x,y)\leftarrow Q}[\mathsf{D}(x,y)=1] \\ &\leq \sum_{i=1}^{n}\Big(\Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}^n}[\mathsf{D}(x, y_{1,\ldots,i-1}, z_{i,\ldots,n})=1] \\ &\qquad - \Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}^n}[\mathsf{D}(x, y_{1,\ldots,i}, z_{i+1,\ldots,n})=1]\Big),\end{aligned}$$

and thus there exists $i \in [n]$ such that

$$\begin{aligned}\epsilon := \Pr_{\substack{(x,y)\leftarrow Q,\\ b\leftarrow\{0,1\}\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x, y_{1,\ldots,i-1}, b, z)=1] - \Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x, y_{1,\ldots,i-1}, y_i, z)=1] \\ \geq \alpha/n,\end{aligned} \tag{12}$$

as we wanted to show. We now describe the predictor P. Consider the following algortihm.

Algorithm A.2 (The predictor P)
Input: $x \in \{0,1\}^*, y_{1,\ldots,i-1} \in \{0,1\}^{i-1}$.
Oracle: A distinguisher D.
Operation:

1. *Sample* $b \leftarrow \{0,1\}$, $z \leftarrow \{0,1\}^{n-i}$ *and execute* $\mathsf{D}(x, y_{1,\ldots,i-1}, b, z)$.
2. *If* D *output* 1*, output* $1-b$*. Otherwise, output* b.

We next show that the probability that P outputs y_i is at least $1/2+\alpha/n$.

Let $p := \Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},y_i,z)=1]$. It holds that

$$\begin{aligned}
p+\epsilon &= \Pr_{\substack{(x,y)\leftarrow Q, b\leftarrow\{0,1\}\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},b,z)=1]\\
&= 1/2\cdot\Big(\Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},y_i,z)=1]\\
&\qquad + \Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},1-y_i,z)=1]\Big)\\
&= 1/2\cdot\Big(p+\Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},1-y_i,z)=1]\Big)).
\end{aligned}$$

Thus, $\Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},1-y_i,z)=1]=p+2\epsilon$. Continue, the probability that P outputs y_i is given by

$$\begin{aligned}
&\Pr_{b\leftarrow\{0,1\}^n}[b=y_i]\cdot(1-p)+\Pr_{b\leftarrow\{0,1\}^n}[b=1-y_i]\\
&\qquad\cdot\Pr_{\substack{(x,y)\leftarrow Q,\\ z\leftarrow\{0,1\}^{n-i}}}[\mathsf{D}(x,y_{1,\dots,i-1},1-y_i,z)=1]\\
&= 1/2\cdot(1-p)+1/2\cdot(p+2\epsilon)\\
&= 1/2+\epsilon\\
&\geq 1/2+\alpha/n
\end{aligned}$$

as needed.

A.2 Universal Hash Families

Lemma A.3 (Lemma 2.13, restated) *For every $\ell,n\in\mathbb{N}$ such that $\ell\leq n$, the family $\left\{m\colon m\in\{0,1\}^{n\times\ell}\right\}$ is approximately-flat.*

Proof (Proof of Lemma 2.13). Fix $\mathcal{Y},x_1,x_2$ and y_1 as in Definition 2.12. We want to show that

$$\Pr_{M\leftarrow\{0,1\}^{2n\times\ell}}[\exists y_2\in\mathcal{Y}\text{ s.t. }M(x_1,y_1)=M(x_2,y_2)]\geq 2^{-10}\cdot\min\left\{|\mathcal{Y}|\cdot 2^{-\ell},1\right\}.$$

We first assume that $x_1\neq x_2$, as otherwise the lemma holds trivially. Next, we observe that M can be written as $M_{\mathcal{X}}\in\{0,1\}^{n\times\ell}$ and $M_{\mathcal{Y}}\in\{0,1\}^{n\times\ell}$, such that for every vectors $x,y\in\{0,1\}^n$ it holds that

$$M(x,y)=(x\cdot M_{\mathcal{X}})\oplus(y\cdot M_{\mathcal{Y}}). \tag{13}$$

We want to bound the probability that there exists $y_2 \in \mathcal{Y}$ such that $M(x_1, y_1) = M(x_2, y_2)$, or equivalently,

$$(x_1 \oplus x_2) \cdot M_{\mathcal{X}} = (y_2 \oplus y_1) \cdot M_{\mathcal{Y}}. \tag{14}$$

Since $x_1 \neq x_2$, it holds that $(x_1 \oplus x_2) \cdot M_{\mathcal{X}}$ is a uniform element in $\{0,1\}^{\ell}$. Thus, we are interested in lower bounding the probability

$$\begin{aligned}&\Pr_{M_{\mathcal{Y}} \leftarrow \{0,1\}^{n \times \ell}, z' \leftarrow \{0,1\}^{\ell}} [\exists y_2 \in \mathcal{Y} \text{ s.t. } z' = (y_2 \oplus y_1) \cdot M_{\mathcal{Y}}] \\ &= \Pr_{M_{\mathcal{Y}} \leftarrow \{0,1\}^{n \times \ell}, z \leftarrow \{0,1\}^{\ell}} [\exists y_2 \in \mathcal{Y} \text{ s.t. } z = y_2 \cdot M_{\mathcal{Y}}]\end{aligned}$$

where the equality holds since $z := z' \oplus y_1 \cdot M_{\mathcal{Y}}$ is a uniform element in $\{0,1\}^{\ell}$ which is independent from $M_{\mathcal{Y}}$. In the following we show that with probability at least $1/2$ over the choice of $M_{\mathcal{Y}}$, the size of the set $\mathcal{Y} \cdot M_{\mathcal{Y}} = \{y \cdot M_{\mathcal{Y}} \colon y \in \mathcal{Y}\}$ is at least $\min\left\{|\mathcal{Y}|/2, 2^{\ell}/32\right\}$, from which the lemma follows.

To see the above, first notice that for every vector $v \in \{0,1\}^n$ with $v \neq 0$, it holds that $\Pr_{M_{\mathcal{Y}}}[v \cdot M_{\mathcal{Y}} = 0] = 2^{-\ell}$, and thus,

$$\begin{aligned}&\mathop{\mathrm{E}}_{M_{\mathcal{Y}}}[|\{y_1 \neq y_2 \in \mathcal{Y} \colon y_1 \cdot M_{\mathcal{Y}} = y_2 \cdot M_{\mathcal{Y}}\}|] \\ &= \mathop{\mathrm{E}}_{M_{\mathcal{Y}}}[|\{y_1 \neq y_2 \in \mathcal{Y} \colon (y_1 \oplus y_2) \cdot M_{\mathcal{Y}} = 0\}|] \\ &\leq |\mathcal{Y}|^2 \cdot 2^{-\ell}.\end{aligned}$$

By Markov inequality, we get that with probability at least $1/2$ over the choice of $M_{\mathcal{Y}}$, it holds that

$$|\{y_1 \neq y_2 \in \mathcal{Y} \colon y_1 \cdot M_{\mathcal{Y}} = y_2 \cdot M_{\mathcal{Y}}\}| \leq 2\,|\mathcal{Y}|^2 \cdot 2^{-\ell}. \tag{15}$$

In the following we show that for every matrix $M_{\mathcal{Y}}$ for which Equation (15) holds, it holds that $\mathcal{Y} \cdot M_{\mathcal{Y}} \geq \min\left\{|\mathcal{Y}|/2, 2^{\ell}/32\right\}$.

Indeed, consider a graph $\mathcal{G}$, in which the set of vertices is $\mathcal{Y}$, and the set of edges E is the set $\{y_1 \neq y_2 \in \mathcal{Y} \colon y_1 \cdot M_{\mathcal{Y}} = y_2 \cdot M_{\mathcal{Y}}\}$. By assumption, $|E| \leq 2\,|\mathcal{Y}|^2 \cdot 2^{-\ell}$. Furthermore, it is not hard to see that $\mathcal{G}$ is composed of disjoint cliques, and that the number of connected components in $\mathcal{G}$ is exactly the size of $\mathcal{Y} \cdot M_{\mathcal{Y}}$. To bound the number of connected components of $\mathcal{G}$, we first assume that $\mathcal{G}$ has no more than $|\mathcal{Y}|/2$ isolated vertices, as otherwise the bound trivially follows. We start with removing the isolated vertices from $\mathcal{G}$, to get a graph with at least $|\mathcal{Y}|/2$ vertices and at most $2\,|\mathcal{Y}|^2 \cdot 2^{-\ell}$ edges. Let k be the number of connected components in the graph, and let $c_1, \ldots, c_k$ be the number of vertices in each component. Since $c_i > 1$ for every i, the number of edges in the i-th component is larger than $c_i^2/4$. By Cauchy–Schwarz inequality,

$$(|\mathcal{Y}|/2)^2 \leq (\sum_{i \in [k]} c_i)^2 \leq k \cdot \sum_{i \in [k]} c_i^2 \leq 4k\,|E| \leq 8k\,|\mathcal{Y}|^2 \cdot 2^{-\ell},$$

which implies that $k \geq 2^{\ell}/32$, and the lemma follows.