On Derandomizing Yao's Weak-to-Strong OWF Construction

Chris Brzuska[1], Geoffroy Couteau[2], Pihla Karanko[1], and Felix Rohrbach[3]

[1] Aalto University, Finland, {chris.brzuska,pihla.karanko}@aalto.fi
[2] CNRS, IRIF, Université Paris Cité, France, geoffroy.couteau@ens.fr
[3] TU Darmstadt, Germany, felix.rohrbach@cryptoplexity.de

Abstract. The celebrated result of Yao (FOCS'82) shows that concatenating $n \cdot p(n)$ copies of a *weak* one-way function (OWF) f, which can be inverted with probability $1-\frac{1}{p(n)}$, yields a *strong* OWF g, showing that weak and strong OWFs are black-box equivalent. Yao's transformation is not *security-preserving*, i.e., the input to g needs to be much larger than the input to f. Understanding whether a larger input is inherent is a long-standing open question.

In this work, we explore necessary features of constructions which achieve short input length by proving the following: for any *direct product* construction of a strong OWF g from a weak OWF f, which can be inverted with probability $1-\frac{1}{p(n)}$, the input size of g must grow as $\Omega(p(n))$. Here, *direct product* refers to the following structure: the construction g executes some arbitrary pre-processing function (independent of f) on its input s, obtaining a vector $(x_1, \cdots, x_l)$, and outputs $f(x_1), \cdots, f(x_l)$. When setting the pre-processing to be the identity, one recovers thus Yao's construction.

Our result generalizes to functions g with post-processing, as long as the post-processing function is not too lossy. Thus, in essence, any weak-to-strong OWF hardness amplification must either (1) be very far from security-preserving, (2) use adaptivity, or (3) must be very far from a *direct-product* structure (in the sense that post-processing of the outputs of f is very lossy).

On a technical level, we use ideas from lower bounds for secret-sharing to prove the impossibility of derandomizing Yao in a black-box way. Our results are in line with Goldreich, Impagliazzo, Levin, Venkatesan, and Zuckerman (FOCS 1990) who derandomize Yao's construction for *regular* weak OWFs by evaluating the OWF along a random walk on an expander graph—the construction is adaptive, since it alternates steps on the expander graph with evaluations of the weak OWF.

1 Introduction

In this work, we continue the study of constructions of strong one-way functions (OWFs) from weak OWFs. The classical weak-to-strong hardness amplification technique, due to Yao [Yao82], uses direct product amplification which is not security preserving[4]. Our main result shows that the increase in the input size is inherent for *direct product* constructions. Namely, any direct product black-box construction of a strong OWF from a $(1-1/p(n))$-weak OWF must have input length at least $\Omega(p(n))$.

Weak and strong OWFs. An $\alpha(n)$-secure OWF $f : \{0,1\}^n \mapsto \{0,1\}^n$ is an efficiently computable function such that any probabilistic polynomial-time adversary $\mathcal{A}$ can invert f with probability at most $\alpha(n)$. When α is a negligible function, we say that f is a *strong* OWF; when $\alpha(n) = 1 - 1/p(n)$ for a polynomial p, we say that f is a *weak* OWF. The seminal work of Yao [Yao82] shows that weak OWFs imply strong OWFs, via a standard *direct product* hardness amplification: given a weak OWF f, define $g(x_1, ..., x_l) = f(x_1)||...||f(x_l)$. Then, Yao proved that g is a strong OWF for $l > |x_i|p(|x_i|)$.

Adaptive vs. non-adaptive construction. In this paper we study *non-adaptive* weak-to-strong OWF constructions, that is, constructions where the calls to the weak OWF can be made in parallel. I.e., a strong OWF construction g that makes calls to a weak OWF f is called *non-adaptive* if g's calls to f only depend on g's input, but not on the output of f on any of these inputs. Yao's construction is a simple, non-adaptive construction where each call to f uses an independent chunk of the input. In general, non-adaptive constructions can make correlated calls to f though.

[4] In a security-preserving construction, the input length of the strong OWF is linear in that of the weak OWF.

We say that a construction is *adaptive*, if the output of (at least) one call to f is used to determine the input to another f call. That is, adaptive constructions cannot compute all calls to f in parallel. For the toy constructions on the right, g_1 is non-adaptive (it does not matter whether g_1 computes $f(x)$ or $f(x+1)$ first) and g_2 is adaptive (g_2 must make the inner f call first).

$$g_1(x) := f(x)||f(x+1)$$
$$g_2(x) := f(f(x))$$

On the (in)efficiency of Yao's construction. The construction of Yao is generic: it turns an *arbitrary* weak OWF f into a strong OWF g and just depends on the hardness of f. In addition, g has an appealing simple direct-product structure. In turn, g is suboptimal w.r.t. its computational complexity:

1. g makes a *large number of calls* to the underlying weak OWF, and
2. g is *not security preserving*: g's input length is polynomially larger than the input length of f.

Many celebrated cryptographic reductions are similarly not security-preserving and have a high number of calls—the HILL construction of pseudorandom generator from any OWF being perhaps one of the most well-known examples [HILL99]. In beautiful works, a decade ago, Haitner, Reingold, Vadhan and Zheng [HRV10, VZ12] developed rich tools for computational entropy, and improved the original n^8 seed length by HILL to $\mathcal{O}(n^3)$, where n is the input length of the OWF—since further improvements seem extremely hard to obtain, it is natural to ask whether large lower bounds on the input size are inherent.

In a seminal result [IR89], Impagliazzo and Rudich formalize the notions of *black-box* constructions/reductions, and develop methods to establish their limitations. Informally, a (fully) black-box construction of a primitive C from a primitive P treats both P and any adversary $\mathcal{A}$ against P in a black-box way. Following this breakthrough result, a long line of work (see [KST99, GT00, GGK03, LTW05, Lu06, CRS+07, Wee07, Lu09]) has been devoted to proving limitations on the *efficiency* of black-box reductions. Our work continues this successful line of work.

To our knowledge, three previous works study black-box limitations on the efficiency of Yao's construction. Lin, Trevisan, and Wee [LTW05] address the first of the two limitations above: they show that any fully black-box construction of an $\varepsilon(n)$-secure OWF from a $(1-\delta(n))$-secure OWF f must make at least $q = \Omega((1/\delta)\cdot\log(1/\varepsilon))$ calls to f. They also show that fully black-box constructions cannot be perfectly security-preserving: if f has input size n, the input size of the strong OWF must be at least $n + \Omega(\log 1/\varepsilon) - O(\log q)$. Later, Lu [Lu09] extends the results of [LTW05] to the weakly black-box setting with bounded non-uniformity. Moreover, Lu [Lu06] shows that a *non-adaptive* fully black-box construction (i.e., a construction where all the calls to f are made in parallel) cannot amplify security beyond $\mathsf{poly}(n)$ if the algorithm implementing the reduction has constant depth, and its size is below $2^{\mathsf{poly}(n)}$.

1.1 On Security-Preserving Amplification of Weak OWFs

The above result leaves open one of the most intriguing limitations of Yao's construction: the fact that it causes a polynomial blowup in the input size. While [LTW05] shows that *some* blowup in the input size is unavoidable, it leaves a huge gap: starting with a $(1-1/p(n))$-secure OWF f with input length n, Yao's construction requires an input size $n^2 \cdot p(n)$ to build *any* strong OWF, while the result of [LTW05] only shows that to build an *extremely strong* OWF, say a $2^{-\mu\cdot n}$-secure OWF (for some constant μ), one needs input size at least $(1+\mu)\cdot n - \log p$.

In a sense, the proof of [LTW05] cannot do much better, because it rules out even *adaptive* fully black-box reductions. However, in this setting, it is actually known that we can do much better than Yao's construction and obtain an almost security-preserving construction, if we start from a *regular* (i.e. outputs have the same number of preimages) weak OWF, and use adaptivity. Indeed, the work of Goldreich, Impagliazzo, Levin, Venkatesan, and Zuckerman [GIL+90] provides precisely such a construction, using random walks on expander graphs. Following that, Haitner, Harnik and Reingold [HHR06] present another almost security-preserving adaptive construction using hash function calls instead of expander steps; and their construction is secure even when the regularity parameters is not known.

This leaves us in between two extremes: on the one hand, Yao's construction is non-adaptive (hence optimally parallelizable: if one starts with a parallelizable weak OWF, one ends up with a parallelizable

strong OWF), extremely simple (it has a straightforward direct product structure) and works for arbitrary OWFs; however, it is not security-preserving. On the other hand, the constructions [GIL+90] and [HHR06] are almost security-preserving, but are considerably more involved, require adaptive calls, and work only for regular OWFs. Improving this state of affairs is a long-standing and intriguing open problem.

1.2 Our Contribution

In this work, we make progress on understanding the *limits* of non-adaptive constructions. Specifically, we show that any *direct product* black-box construction of strong OWF from a $(1-1/p(n))$-secure OWF cannot be security preserving, in a strong sense: it requires an input length of at least $\Omega(p(n))$. While this still leaves a gap with respect to Yao's construction, which has input length $O(n^2 \cdot p(n))$, this gap vanishes asymptotically when p grows. By *direct product* construction, we mean a construction g of strong OWF with the following structure: on input s, $g(s)$ outputs $(f(x_1), \cdots, f(x_\ell))$, where f is the weak OWF, and $(x_1, \cdots, x_\ell)$ are computed from s arbitrarily, but without calling f (we call the mapping from s to $(x_1, \cdots, x_\ell)$ the *pre-processing*). This is a natural generalization of *Yao-style* constructions of strong OWFs (we recover Yao's construction by letting the pre-processing be the identity function). Furthermore, our result generalizes to the setting where some post-processing (independent of f) is applied to the outputs $(f(x_1), \cdots, f(x_\ell))$, whenever this post-processing is *not too lossy*: we prove that whenever each output of the post-processing has at most polynomially many preimages, the same $\Omega(p(n))$ input length bound holds. We summarize the results in the following theorem:

Theorem 1 (Informal). *Let f be a $(1-1/p(n))$-secure OWF (a weak OWF). Let g be any non-adaptive construction, with not-too-compressing post-processing, of input length $< cp(n)$ for any constant c. Then, it is impossible to prove, in a fully black-box way, that g is a strong OWF.*

Observe that if we could generalize our result to arbitrary (f-independent) post-processing functions, the above would capture all non-adaptive constructions. Hence, in essence, our result says the following: any (fully black-box) construction of strong OWF from a weak OWF must either (1) be very far from security preserving, or (2) use adaptivity, or (3) compute a highly non-injective function of the outputs of the non-adaptive calls (i.e., be very far from a "direct product" structure).

1.3 Relation to Correlated-Product and Correlated-Input Security

Usually, parallel concatenation of cryptographic primitives on *independent* inputs preserves security. For example, if f and g are one-way functions, then so is $(x_1, x_2) \mapsto (f(x_1), g(x_2))$. However, things might change significantly when x_1 and x_2 are *correlated*, e.g., sampled jointly from a high min-entropy source. Variants of this problem have been studied on many occasions in cryptography, and have profound connections to the feasibility of cryptography with weak sources of randomness, leakage-resilient cryptography, related-key attacks, or deterministic encryption (to name a few); see e.g. Wichs [Wic13] for discussions on cryptography with correlated sources. In addition, security for correlated inputs has proven to be a very useful assumption by itself: one-wayness under correlated product (i.e., one-wayness of $f(x_1), \cdots, f(x_k)$ for $(x_1, \cdots, x_k)$ sampled from a joint distribution) has been used to build CCA secure cryptosystems [RS09, HLO12], and correlated-input secure hash functions have found numerous applications such as OT extension [IKNP03], trapdoor hash function [DGI+19], constrained pseudorandom functions [AMN+18], password-based login [GOR11], and many more.

A general and natural question to ask is: which type of constructions *preserve* hardness, when the inputs are jointly sampled from a high min-entropy source, rather than being sampled independently? This is a fundamental question in itself, because this setting occurs in real-life use of standard cryptographic construction (when they are misused, when the source of randomness is imperfect, or when the adversary has access to some leakage on the inputs), but also due to the many applications outlined above.

It is well-known that not all constructions will preserve security under correlated inputs. For example, even though the map $x \mapsto x^e \bmod n$ is believed to be one-way when n is a product of two large safe primes (this is the RSA assumption), the extended euclidean algorithm provides an efficient inverter for the map $x \mapsto (x^{e_1}, x^{e_2}) \bmod n$ whenever $\gcd(e_1, e_2) = 1$ (this example is taken

from [HLO12]). Hence, there are specific functions f_i (here, $f_i : x \mapsto x^{e_i}$) and specific correlations of the inputs (here, the equality correlation: the same input x is used for all functions) such that correlated-product security breaks down. However, this leaves open the possibility that some specific input correlations preserve correlated-product security (for example, this is the case when the correlated-inputs are indistinguishable from random, e.g. when sampled as the output of a PRG), or that some specific functions maintain correlated-product security for general correlations.

Our results can be cast in the context of correlated-product security: we show that even though Yao's construction of OWF, which is a very natural and seminal construction, is provably secure (with a black-box proof) when used with random and independent inputs, it breaks down for *any possible correlated source*, whenever the entropy of the source is below $p(n)$. This provides a natural example of a construction, from a weak OWF f, where correlated-product security cannot be generically shown to hold (in a black-box way) for *arbitrary* sources, unless they contain enough entropy such that all of the correlated inputs can have independent entropy. In contrast, [RS09] shows that when f is instantiated as a *lossy trapdoor function*, then $f(x_1), \cdots, f(x_k)$ is one-way for correlated inputs $(x_1, \cdots, x_k)$, and [HLO12] shows that assuming OWFs, there *exists* a correlated-product secure function. Our results provide a partial complementary perspective to this line of work.

Comparison to [Wic13]. Wichs [Wic13] also studies, among other questions, the one-wayness of constructions of the form $(f(x_1), \cdots, f(x_k))$ for inputs $(x_1, \cdots, x_k)$ sampled from a correlated source. Our results are incomparable: we show that for a *generic weak OWF* f, and for any *fixed* distribution over the inputs $(x_1, \cdots, x_k)$ with $o(k)$ bits of entropy, the one-wayness of $f(x_1), \cdots, f(x_k)$ does not follow from that of f in a black-box way. In contrast, [Wic13] shows that for an *arbitrary function* f, there is no black-box reduction (to any standard hardness assumption) of one-wayness of $(f(x_1), \cdots, f(x_k))$ when the x_i can come from *arbitrarily correlated distributions*, even with high per-input entropy. That is, [Wic13] handles a considerably larger class of constructions and reductions to many possible assumptions, but only rules out a much more stringent security notion (where one-wayness must hold even when the input distributions are not fixed a priori and can be correlated arbitrarily). To state the difference using quantifiers (we omit the hardness parameter of the weak OWF for simplicity), we show that

$$\exists \text{ weakOWF } f \, \forall \text{ pre-proc: } g(x) := f(\text{pre-proc}(x)_1), ..., f(\text{pre-proc}(x)_l) \text{ is } \textbf{not} \text{ a strong OWF}$$

while Wichs [Wic13] shows that

$$\exists \text{ corr-srce} \, \forall f : \; g(x) := f(\text{corr-srce}(x)_1), ..., f(\text{corr-srce}(x)_l) \text{ is } \textbf{not} \text{ a strong OWF}$$

that is, the quantification is in different order, as is highlighted in pink. To put it differently, we *rule out* fully black-box constructions of the form

$$\forall \text{ weakOWF } f \, \exists \text{ pre-proc: } g(x) := f(\text{pre-proc}(x)_1), ..., f(\text{pre-proc}(x)_l) \text{ is a strong OWF}$$

and Wichs rules out fully black-box constructions of the form

$$\forall \text{ corr-srce} \, \exists f : \; g(x) := f(\text{corr-srce}(x)_1), ..., f(\text{corr-srce}(x)_l) \text{ is a strong OWF.}$$

1.4 Related Work

We already discussed several works on bounding the efficiency of black-box reductions [KST99, GT00, GGK03, LTW05, Lu06, CRS+07, Wee07, Lu09], including some specifically targeting hardness amplifications of one-way functions, and related work on correlated-product security. Besides, our black-box separations use some established tools (in addition to key new technical insights, which we cover afterwards) such as the two-oracle technique of [Sim98, HR04] where one oracle implements the base primitive and the second oracle breaks all constructions built from this primitive. We use the Borel-Cantelli style technique from [MMN+16] to extract a single oracle from a distribution of random oracles analogously to the seminal work on black-box separations by Impagliazzo and Rudich [IR89].

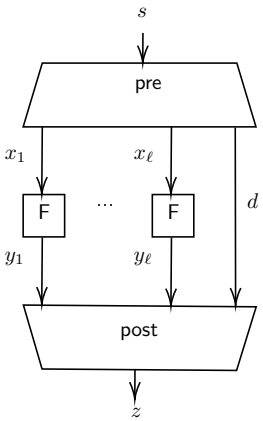
Hardness amplification of functions, via direct products and related constructions, have a rich history, which goes well beyond one-way functions and is too vast to be covered here. In particular, amplifying the hardness of *computing* boolean functions (rather than *inverting* functions) using direct product constructions is at the heart of major lines of work on worst-case to average-case reductions, constructions of non-cryptographic pseudorandom generators, circuit lower bounds, and many more – see e.g. [Lip91, BFL91, BFNW93, Imp95, GNW95, IW97, STV01, Tre03, HVV04, Tre05, Vio05, SV08] and references therein.

1.5 Technical Overview

To prove our black-box separations, we exhibit an oracle relative to which there is a weak one-way function, yet all strong one-way functions with an appropriate structure can be inverted efficiently with constant probability. The standard method to do so is to design oracles relative to which the starting primitive (here, the weak one-way function) clearly exists and is the *only possible source of hardness*. For example, in the seminal work by Impagliazzo and Rudich (IR) on the separation of key exchange from OWFs [IR90], IR introduce a random oracle, which is a strong OWF with high probability, as well as assuming $\mathsf{P} = \mathsf{NP}$, thereby ruling out most other (stronger) cryptographic primitives. In our setting, we instantiate this intuition by choosing three oracles:

(1) A PSPACE oracle, which *destroys* all possible sources of hardness,
(2) a random oracle F, which instantiates the weak OWF, and
(3) an inverter INV, which inverts F on a (roughly) $1 - 1/p$ fraction of its inputs, effectively turning it into a weak OWF. Note that a random oracle F alone would already be a strong OWF, if we did not weaken it by adding INV.

In this oracle world, we consider *non-adaptive* constructions of strong OWFs g from the weak OWF F. Since we wish to rule out (relativizing) *fully black-box* reductions (as defined by Reingold, Trevisan and Vadhan [RTV04]), we do not give g access to INV. In fact, this is inherent in our setting: observe that given access to INV, it is not too hard to build a strong OWF (e.g. the strong OWF can perform a random walk starting from the input x, until it lands on a hard input y – which can be tested using INV – and outputs $\mathsf{F}(y)$). In general, whenever one can efficiently test which inputs are hard, constructing a security-preserving OWF becomes feasible – and it is precisely the lack of any such tester that makes it highly nontrivial to improve over Yao's seminal construction. Since we rule out fully black-box reductions, we do not let g access INV and thus, g does not know where the easy inputs are.



$g(\mathrm{s})$

$x_1, ..., x_\ell, d \leftarrow \mathsf{pre}(s)$
for $i = 1..\ell$
  $y_i \leftarrow \mathsf{F}(x_i)$
$z \leftarrow \mathsf{post}(y_1, ..., y_\ell, d)$
return z

Fig. 1: (n, m)-non-adaptive construction. F is the weak OWF. Length of d can be arbitrary, $|x_i| = |y_i| = m$ and $|s| = n$.

Modeling non-adaptive constructions. A non-adaptive construction can be thought of as a circuit which first has a pre-processing layer, followed by a layer of parallel calls to a weak OWFs and then some post-processing, see Figure 1. When the construction omits the post-processing layer, as in Yao's construction, this corresponds to a *direct product* construction. The input size n of the construction might be different from the input size m of the weak OWF. As a starting point, we consider what happens when the construction does not use any post-processing, as is the case in Yao's construction. When there is no post-processing, the additional data d in Figure 1 only reduces the input domain and does not add security. Thus, w.l.o.g., we assume that there is no d.

Inverting direct product constructions. Considering the simple case with no post-processing and no d, the first observation is that g must make more than $p(m)$ calls to the weak OWF, since otherwise *all* the calls will be easy to invert with constant probability. In that case the adversary could simply invert all the weak OWF calls and then use PSPACE to invert the pre-processing layer, thus inverting g with constant probability.

Now that g makes at least $p(m)$ calls to the weak OWF, we can make the main observation of the paper: if we can invert a $1 - 1/p(m)$ fraction of the weak OWF calls and n is about the same as $p(m)$, then the remaining entropy of the input s cannot be very high, on the average. This is formalized in Lemma 19. This is because the number of calls to the weak OWF is at least the same order of magnitude as the length of the input to the strong OWF. Hence, there is not enough entropy in the strong OWF input to distribute among all the weak OWF calls, so most of the calls will end up having very little entropy of their own, i.e. entropy that is not shared with other calls.

Now the probability that an adversary can indeed invert a $1 - 1/p(m)$ fraction of the weak OWF calls is high, since that is the expected fraction of easy calls. Since the entropy of the input s is low, given the easy calls, and the adversary has the PSPACE oracle, the adversary can guess s with high probability. Note that low entropy alone is not enough to guess s, since inverting pre-processing might be inefficient, hence we also need PSPACE.

To summarize, we know that there must be many calls to the underlying weak one-way function—and since we can also show that each of them must have a non-trivial amount of entropy (i.e., information about the input)—we can show that we can invert all non-adaptive constructions without post-processing, unless n is larger than any constant times $p(m)$, establishing the first lower bound on the randomness efficiency of non-adaptive constructions. Note that Yao's construction consumes $n = m^2 p(m)$ many bits.

On strong OWFs with injectiveish post-processing. We sketched above why constructions without post-processing (direct product constructions) cannot be strongly one-way. It is relatively easy to extend the above argument to constructions with *not too lossy* post-processing, i.e., constructions where any output of the post-processing has at most polynomially many preimages: the inverter chooses a uniformly random value amongst the (polynomial size) list of all possible preimages of the post-processing, and applies the previous inversion attack on the candidate. It then succeeds with probability $\frac{1}{\mathsf{poly}}$ times the success probability of the previous attack.

Relation to Threshold Secret Sharing. The pre-processing pre in Figure 1 is somewhat analogous to a threshold secret sharing scheme, where the participants' shares correspond to the values x_i and the secret together with the dealer's randomness corresponds to the strong OWF input s. On average, we learn the 'shares' of all but a $\frac{1}{p(m)}$ fraction of the 'participants'.

The analogy is somewhat weak though, since the OWF inverter needs to find one (whole) preimage s, while a secret sharing scheme is broken if the adversary finds the secret which is a (known) function of s (even if they cannot recover the dealer's randomness, that is, all of s). Also, computational security suffices for OWFs while secret sharing schemes usually aim for information-theoretic security.

Interestingly, in the proof of our negative result, the adversary has a PSPACE oracle, and hence, in the proof we care about the *information theoretic* security. This is because PSPACE can invert computationally hard pre-processing, as long as it is not information theoretically impossible, i.e. the input does not have too high entropy conditioned on the adversary's knowledge. Hence, even though the object we study is not formally related to secret sharing, our proof is inspired by previous work on secret sharing by Blundo, Santis, and Vaccaro (BSV [BSV96]).

BSV discuss the minimum amount of randomness needed by an information-theoretically secure secret sharing scheme. BSV prove that if the secret length is m and there are l participants, then the dealer needs to use $l \cdot m$ bits of randomness (to choose both the secret and the participants' shares). The length of this randomness corresponds to the analogous number in Yao's weak to strong OWF construction (when number of weak OWF calls is $l > mp(m)$, we use lm input length) and it is close to the analogous number that we get in this paper (input length to strong OWF needs to be $\mathcal{O}(p(m))$, i.e. there is an m^2 gap between our lower bound and Yao's upper bound).

It is intuitive that some gap should exist between the information theoretically secure secret sharing scheme and our more relaxed "mostly secure secret sharing scheme", where the adversary is allowed to learn a function of the input as long as they cannot learn the entire input. However, the OWF construction and secret sharing schemes are not formally related and a better lower bound than ours might be possible to obtain.

2 Preliminaries

We use $x \leftarrow\!\$\, S$ for sampling x uniformly from set S, and $y \leftarrow\!\$\, \mathcal{A}(x)$ for running randomized algorithm $\mathcal{A}$ on x with uniformly random coins and assigning the result to y. We write $\Pr_{\mathcal{A}}[1 = \mathcal{A}(x)]$ for the probability over the randomness of $\mathcal{A}$ that $\mathcal{A}$, on input x, returns 1. We now introduce weak and strong one-way functions, oracle algorithms, relativization, black-box reductions and entropy.

Definition 2 (One-Way Functions). *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is called a* (strong) *one-way function (OWF), if for every probabilistic polynomial-time*

algorithm $\mathcal{A}$ there exists a negligible function $\epsilon : \mathbb{N} \to [0,1]$ such that for every n,

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

Further, f is called a weak *one-way function, if there exists a polynomial $p(n)$ such that for every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a $N_0 \in \mathbb{N}$ such that for all $n \geq N_0$:*

$$\Pr_{\mathcal{A},x\leftarrow\{0,1\}^n}\left[\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \leq 1 - \frac{1}{p(n)}.$$

*In this case we sometimes say that f is a p-*weak *OWF.*

Definition 3 (Oracle Algorithms). *The complexity of an oracle algorithm (e.g., Turing Machine) is the number of steps it makes, where an oracle query is counted as a single step.*

In particular, a probabilistic polynomial-time (PPT) oracle algorithm makes at most a polynomial number of queries. Since our oracle algorithms have access to a PSPACE oracle, our runtime discussions only consider the number of oracle calls the algorithm makes.

Definition 4 (Relativizing Statements). *We say that a statement about algorithms* relativizes *if it also holds whenever all algorithms are given oracle access to an arbitrary (deterministic) function* O.

To rule out a relativizing statement, we first argue about *distributions* of oracles and then show the existence of a single oracle using the following Borel-Cantelli style theorem.

Theorem 5 ([MMN+16], Lemma 2.9). *Let $(E_1, E_2, ...)$ be a sequence of events such that $\exists c \forall m \in \mathbb{N} : \Pr[E_m] \geq c$, where constant c is $0 < c < 1$. Then,*

$$\Pr\left[\bigwedge_{k=1}^{\infty}\bigvee_{m>k} E_m\right] \geq c \tag{1}$$

Intuitively, Theorem 5 says that if an event happens with constant probability for all m, then, with constant probability, the event happens infinitely often.

Entropy. Throughout this paper, the term *entropy* refers to *Shannon entropy* which satisfies a *chain rule.*

Definition 6 (Shannon Entropy). *Let X be a random variable and let $\mathsf{dom}(X)$ be its domain, then*

$$H(X) := -\sum_{z\in\mathsf{dom}(X)} \Pr[X = z] \cdot \log_2(\Pr[X = z]),$$

is the Shannon entropy *of X.*

Lemma 7 (Chain Rule for Entropy). *Let $X_1, \ldots, X_n$ be random variables. Then the following holds*

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2|X_1) + \cdots + H(X_n|X_1, \ldots, X_{n-1}).$$

We use also other simple but useful properties of entropy. In particular, Definition 6 implies that entropy is non-negative. Also, the entropy $H(X)$ of a random variable X is always more or equal to the entropy $H(f(X))$ of the random variable $f(X)$ for any deterministic function f—if f is injective, the entropy is preserved, if f is not injective, it decreases. Finally, for any three random variables X, Y, Z, we have that $H(X|Y) \geq H(X|Y, Z)$, i.e., conditioning on additional information maintains or decreases the entropy of a random variable.

3 Main Results

In this section, we introduce different types of constructions of strong OWF from weak OWF which we study in this paper (Section 3.1) and state our main theorems (Section 3.2). In particular, we introduce non-adaptive constructions, non-adaptive constructions without post-processing and non-adaptive constructions with injective*ish* post-processing.

3.1 Black-box constructions and reductions

Definition 8 (Non-adaptive). *A construction $g = (\mathsf{pre}, \mathsf{post})$ from a weak one-way function F is non-adaptive, if it computes its output as $\mathsf{post}(\mathsf{F}(\mathsf{pre}(s)))$ (see Fig 1). The number of queries ℓ is induced by pre. (n,m)-NA denotes a non-adaptive construction with input length n based on a weak OWF F whose input length is m.*

Definition 9 (Non-adaptive, no post-processing construction). *We say that a construction $g = (\mathsf{pre}, \mathsf{post})$ is a (n,m)-NANPP, if it is (n,m)-NA and the post-processing function is the identity function, i.e., $\mathsf{post}(y_1, ..., y_\ell, d) := y_1||..||y_\ell||d$.*

Definition 10 (Non-adaptive, injective*ish* post-processing constr.). *We say that a construction $g = (\mathsf{pre}, \mathsf{post})$ is a (n,m)-NAIPP, if it is (n,m)-NA and the post-processing function is almost injective, that is, every image of post has at most a polynomial (in n) number of preimages.*

Note that the identity function is injective and thus, in particular, is injective*ish*. Therefore, every NANPP is also a NAIPP, but the converse does not hold. Likewise, both NANPP and NAIPP are NA constructions, but the converse does not hold. Since we are interested in ruling out constructions, whenever we rule out NAIPP, we also rule out NANPP.

We formalized the kind of constructions our negative results capture, and now specify which type of reduction proofs our theorems rule out. Namely, our results concern BBB-style proofs following the notation of [BBF13] or fully black-box proofs following the notation of [RTV04]. Since we consider parametrized definitions, we here state a customized version of fully black-box security which precisely captures the quantifiers our negative results capture.

Definition 11 (Fully Black-Box Proof). *We say that a proof that weak OWF implies strong OWF is* fully black-box *if it establishes a* relativizing *statement of the following type:*

$$\forall \textit{poly } p, \exists \textit{ poly-time computable } g_p, \forall \textit{poly } q, \exists \textit{PPT } \mathcal{R}_q \, \forall p\textit{-weak OWF } \mathsf{F}, \mathcal{A}:$$

$$\left(\Pr_{x \leftarrow \$ \{0,1\}^n} \left[g_p^{\mathsf{F}}(\mathcal{A}(1^n, g_p^{\mathsf{F}}(x))) = g_p^{\mathsf{F}}(x) \right] > \tfrac{1}{q(n)} \textit{ for infinitely many } n \in \mathbb{N} \right) \tag{2}$$

$$\Rightarrow \left(\Pr_{x \leftarrow \$ \{0,1\}^n} \left[\mathsf{F}(\mathcal{R}_q^{\mathcal{A},\mathsf{F}}(1^n, \mathsf{F}(x))) = \mathsf{F}(x) \right] > 1 - \tfrac{1}{p(n)} \textit{ for inf. many } n \right) \tag{3}$$

In this case, we also refer to the construction g *as fully black-box.*

Intuitively, line (2) says that an adversary $\mathcal{A}$ breaks the strong OWF g^{F} and line (3) says that the reduction $\mathcal{R}^{\mathcal{A}}$ breaks the weak-OWF F.

Remark. Typically, in the definition of fully black-box, the pink parts are omitted. That is, the polynomial p is considered part of the definition of F and the polynomial q is considered as part of the definition of the adversary $\mathcal{A}$, namely its success probability. We allow the construction g to depend on the polynomial p and the reduction $\mathcal{R}$ to depend on q, since we seek to cover a larger and meaningful class of proofs. In particular, Yao's original proof building strong OWFs from weak OWFs is fully black-box in the sense of Definition 11, but would not be covered if the construction were not allowed to depend on p or if the reduction was not allowed to depend on q.

3.2 Theorems

We now state our main theorems, all of which rely on the two-oracle technique. Namely, we construct a distribution over oracles $(\mathcal{O}_1, \mathcal{O}_2)$ such that $\mathcal{O}_1$ is a *weak* one-way function and $\mathcal{O}_2$ helps to invert the *strong* one-way function and is part of the adversary. Since we rule out black-box reductions rather than provide an oracle separation, only the reduction has access to the (adversary) oracle $\mathcal{O}_2$ while the construction does not (cf. Section 1.5). Corollary 14 extracts a single oracle from the oracle distribution, using the Borel-Cantelli style argument Theorem 5. However, we prefer to state our theorem in terms of oracle distributions since this matches the technical core arguments of our separation results more closely.

Theorem 12 (NANPP Impossibility). *$\forall$ constant $\mathbf{d}, \forall$poly p, $\forall(n,m)$-NANPP g with input length $n \leq \mathbf{d} \cdot p(m)$, $\exists$ poly-query $\mathcal{A}$, $\exists$poly $q(n) = n^c, c \in \mathbb{N}_+$, $\forall$PPT $\mathcal{R}$, $\exists$ distribution $\mathcal{D}$ over pairs of oracles $(\mathcal{O}_1, \mathcal{O}_2)$:*

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] = \text{constant} < 1$$

where $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ is an indicator variable that is 1 iff at least one of the following is true:

1. *Weak OWF breaks:*
   $\Pr_{x\leftarrow\$\{0,1\}^m,\mathcal{R}}\left[\mathcal{R}^{\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2},\mathcal{O}_1,\mathcal{O}_2}(1^m, \mathcal{O}_1(x)) \in \mathcal{O}_1^{-1}(\mathcal{O}_1(x))\right] \geq 1 - \frac{1}{p(m)}$.
2. *Strong OWF is secure-ish:*
   $\Pr_{s\leftarrow\$\{0,1\}^n,\mathcal{A}}\left[\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}(1^n, g^{\mathcal{O}_1}(s)) \in (g^{\mathcal{O}_1})^{-1}(g^{\mathcal{O}_1}(s))\right] \leq \frac{1}{q(n)}$.

Remark. In the definition of the bad event, the *oracles* are *fixed* and the randomness is taken only over the sampling of x as well as the internal randomness of $\mathcal{A}$ and $\mathcal{R}$, respectively. Thus, $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ is indeed a well-defined event once the oracles $(\mathcal{O}_1, \mathcal{O}_2)$ have been sampled from $\mathcal{D}$.

Theorem 13 (NAIPP Impossibility). *$\forall$ constant $\mathbf{d}, \forall$poly p, $\forall(n,m)$-NAIPP g with input length $n \leq \mathbf{d} \cdot p(m)$, $\exists$ poly-query $\mathcal{A}$, $\exists$poly $q(n) = n^c, c \in \mathbb{N}_+$, $\forall$PPT $\mathcal{R}$, $\exists$ distribution $\mathcal{D}$ over pairs of oracles $(\mathcal{O}_1, \mathcal{O}_2)$:*

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] = \text{constant} < 1$$

where $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ is the same indicator variable as in Theorem 12.

We use the same oracle distribution for Theorem 13 and Theorem 12, see Section 4 for its definition. Theorem 13 implies Theorem 12, so it would suffice to prove Theorem 13. However, we found the presentation to be easier to follow when presenting the proof of the weaker Theorem 12 first (Section 5.2) and then discussing the generalization to the proof of Theorem 13 (Section 6). For both theorems, we prove that relative to $\mathcal{O}_1, \mathcal{O}_2$, oracle $\mathcal{O}_1$ is a weak OWF. Before proving the theorems for oracle distributions, we now extract a single oracle from the distribution where the bad event happens with *constant* probability. Since the standard Borel-Cantelli lemma requires the probability to be less than $1/m^2$, we here use the strengthened version by Mahmoody, Mohammed, Nematihaji, Pass and Shelat [MMN+16].

Corollary 14 (Main). *Let $\mathbf{d}$ be any constant. There is no fully black-box (n,m)-NAIPP construction of a OWF from a $p(m)$-weak OWF with $n \leq \mathbf{d} \cdot p(m)$.*

Proof. Recall that a black-box proof means the following:

$$\begin{aligned}
&\forall\text{poly } p, \exists \text{ poly-time computable } g, \forall\text{poly } q, \exists\text{PPT } \mathcal{R}\, \forall p\text{-weak OWF } \mathsf{F}, \mathcal{A}: \\
&(\mathcal{A} \text{ inverts } g) \Rightarrow (\mathcal{R}^{\mathcal{A}} \text{ inverts } \mathsf{F}) \text{ Formally:} \\
&\quad \left(\Pr_{x\leftarrow\$\{0,1\}^n}\left[g^{\mathsf{F}}(\mathcal{A}(1^n, g^{\mathsf{F}}(x))) = g^{\mathsf{F}}(x)\right] > \tfrac{1}{q(n)} \text{ for infinitely many } n \in \mathbb{N}\right) \\
&\Rightarrow \left(\Pr_{x\leftarrow\$\{0,1\}^n}\left[\mathsf{F}(\mathcal{R}^{\mathcal{A},\mathsf{F}}(1^n, \mathsf{F}(x))) = \mathsf{F}(x)\right] > 1 - \tfrac{1}{p}(n) \text{ for infi. many } n \in \mathbb{N}\right)
\end{aligned}$$

In order to rule out a black-box proof, we thus define an oracle $\mathcal{O}_1$ (and an oracle $\mathcal{O}_2$ helping the adversary) such that the following holds:

$$\begin{aligned}
&\forall\text{poly } p, \forall \text{ poly-time } g^{\mathcal{O}_1}, \exists\text{poly } q, \forall\text{PPT } \mathcal{R}^{\mathcal{O}_1,\mathcal{O}_2}\, \exists\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}, \exists\mathcal{O}_1, \mathcal{O}_2: \\
&\mathcal{A} \text{ breaks } g_1^{\mathcal{O}}, \text{ but } \mathcal{R} \text{ does not } p\text{-invert } \mathcal{O}_1. \text{ Formally:} \\
&(1)\ \Pr_{x\leftarrow\$\{0,1\}^n}\left[g^{\mathcal{O}_1}(\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}(1^n, g^{\mathcal{O}_1}(x))) = g^{\mathcal{O}_1}(x)\right] > \tfrac{1}{q(n)} \text{ for inf. many } n \in \mathbb{N} \\
&(2)\ \Pr_{x\leftarrow\$\{0,1\}^n}\left[\mathcal{O}_1(\mathcal{R}^{\mathcal{A},\mathcal{O}_1,\mathcal{O}_2}(1^n, \mathcal{O}_1(x))) = \mathcal{O}_1(x)\right] < 1 - \tfrac{1}{p}(n) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for all but finitely many } n \in \mathbb{N}
\end{aligned}$$

Strictly speaking, we only need to prove the above for *some* polynomial p, but since our proof establishes (1) and (2) for *all* polynomial p anyways, we prefer to state this stronger statement here.

Now, let us fix a polynomial p, a candidate NAIPP g, a polynomial q (s.t. it satisfies Theorem 13) and a candidate reduction $\mathcal{R}$ and show the existence of an adversary and a p-weak OWF F.

By Theorem 13, there is an oracle distribution over pairs $(\mathcal{O}_1, \mathcal{O}_2)$, and an adversary $\mathcal{A}$ such that the probability of the bad event $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ is constant in m. We show that there exists a *fixed* oracle pair $(\mathcal{O}_1, \mathcal{O}_2)$ for which the bad event $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ in Theorem 13 happens only for finitely many m. From that it follows that there is a fixed oracle pair for which $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}$ breaks the candidate strong OWF $g^{\mathcal{O}_1}$ infinitely many often, but the reduction $\mathcal{R}^{\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}}$ inverts the weak OWF $\mathcal{O}_1$ well enough at most on finitely many m. Thus, it suffices to show via Theorem 5, that Theorem 13 implies that there is an oracle relative to which $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ happens only for finitely many m.

By Theorem 13, we have

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] = \mathsf{constant} < 1.$$

Hence, the constant probability version of Borel-Cantelli (Theorem 5) yields

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\bigwedge_{m=1}^{\infty}\bigvee_{m>k}\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] = \mathsf{constant} < 1,$$

which means that, with constant probability, there is a k for which no $m > k$ satisfies $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$. Taking such an oracle pair $(\mathcal{O}_1, \mathcal{O}_2)$ concludes the proof of Corollary 14.

□

4 Oracle Distributions

In this section, we define the oracle (distribution)s we rely on. Firstly, a PSPACE creates a world where no one-way functions exist. Then, we add an oracle (distribution) F in order to create a world where weak one-way functions exist, and finally, we add an oracle (distribution) $\mathcal{O}_2$ which breaks NANPP and NAIPP constructions. The adversary will have access to $\mathcal{O}_2$, PSPACE and F while the candidate strong OWF construction only has access to PSPACE and F, but not to $\mathcal{O}_2$. We recall from Section 1.5 that it is *necessary* to not give the construction access to the information which parts of F are easy and which parts are hard, and not giving the construction access to $\mathcal{O}_2$ is related to this necessary restriction, since the adversary (modeled by $\mathcal{O}_2$) uses the information of which parts are easy. On a technical-conceptual level, it is meaningful to not give the construction access to the adversary (modeled by $\mathcal{O}_2$), since the adversary is *inefficient*, while the construction is efficient (in this (oracle) world where all algorithms have access to PSPACE and F). We consider an inefficient adversary since we rule out black-box reduction which work for *any* black-box adversary that breaks the strong OWF, including inefficient ones.

As mentioned before, we denote our adversary by $\mathcal{O}_2$. We encode the pair of oracles PSPACE and F into a single oracle $\mathcal{O}_1$ so that we are aligned with the terminology of a two-oracle separation result (and this is also convenient notation in the proof).

Definition 15 (Oracle Distributions). *Let* $\mathbf{p}$ *be any fixed polynomial. The oracle distribution* $D_{\mathbf{p}}$ *over oracles* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *samples permutations* Π_m *of the elements in* $\{0,1\}^m$ *for every* $m \in \mathbb{N}$ *and a random subset* $\mathsf{EASY}_{\mathsf{in}}^m$ *of* $\{0,1\}^m$ *s.t.* $|\mathsf{EASY}_{\mathsf{in}}^m| = \lceil(1 - 1/\mathbf{p}(m))2^m\rceil$. *We define*

$$\mathcal{O}_1 := (\mathsf{PSPACE}, \mathsf{F}) \text{ and } \mathcal{O}_2 := \mathsf{INV},$$

where F *and* INV *behave as follows:*

$\mathsf{F}(x)$	$\mathsf{INV}(y)$
$m \leftarrow \lvert x\rvert$	$m \leftarrow \lvert y\rvert$
$y \leftarrow \Pi_m(x)$	**if** $y \in \mathsf{EASY}_{\mathsf{out}}^m$
return y	**return** $\mathsf{F}^{-1}(y)$
	else return $\perp$

Here, we use $\mathsf{EASY}_{\mathsf{out}}^m := \Pi_m(\mathsf{EASY}_{\mathsf{in}}^m)$.

Remark. Throughout this paper we treat $(1 - 1/\mathbf{p}(m))2^m$ as an integer, omitting the ceil function since the difference does not affect our proofs.

5 Proof of Theorem 12

We split the proof of Theorem 12 into two parts. We first show that the probability of Case 1 (weak OWF breaks) of the bad event introduced in Theorem 12 is smaller than *any* constant (Section 5.1), and then we show that the probability of Case 2 (strong OWF is secure-ish) of the bad event introduced in Theorem 12 is a small constant (Section 5.2). Recall that both probabilities are (only) over the sampling of the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$.

5.1 $\mathcal{R}^{\mathcal{A}}$ is not a successful weak OWF inverter

In this section, we show that the probability (over the oracle distributions) that F is not a $2\mathbf{p}(m)$-weak OWF is small.

Theorem 16 (F is Weak OWF). *For all constants* $\mathbf{c}$*, for all polynomials* $\mathbf{p}$*, for all poly-query* $\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}$*, for all adversaries* $\mathcal{R}$ *making polynomially many (in* m*) queries to the oracles* F*,* PSPACE*,* INV*,* $\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}$*,*

$$\Pr_{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}\leftarrow\$D_{\mathbf{p}}}\left[\mathsf{SuccInv}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}_{\mathcal{A},\mathcal{R}} \geq 1 - \frac{1}{2\mathbf{p}(m)}\right] \leq 1/\mathbf{c}$$

where $\mathsf{SuccInv}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}_{\mathcal{A},\mathcal{R}}$ *is defined as*

$$\Pr_{x\leftarrow\$\{0,1\}^m,\mathcal{R}}\left[\mathcal{R}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV},\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}}(1^m,\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))\right].$$

When we define $\mathbf{p}(m) := \frac{1}{2}p(m)$*, the above is equivalent to*

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\textit{Case 1 of } \mathsf{Bad}^{\mathcal{R},\mathcal{A},g}_m\right] \leq 1/\mathbf{c},$$

where $\mathcal{D} := D_{\mathbf{p}}$*,* $\mathcal{O}_1 := \mathsf{F},\mathsf{PSPACE}$*,* $\mathcal{O}_2 := \mathsf{INV}$ *and* $\mathsf{Bad}^{\mathcal{R},\mathcal{A},g}_m$ *is defined as in Theorem 12.*

The proof of Theorem 16 uses standard techniques for arguing one-wayness of a random oracle. We include the proof in Appendix B for completeness.

5.2 $\mathcal{A}$ is a successful strong OWF inverter

We prove that an adversary with access to the oracles F, INV and PSPACE (cf. Section 4), can break all short input NANPP constructions which have access to F and PSPACE only.

Theorem 17 (Inverting OWF Candidate). $\forall$ *constant* $\mathbf{d}, \forall$ *poly* $\mathbf{p}$*,* $\forall(n,m)$*-NANPP* g *with input length* $n \leq \mathbf{dp}(m)$*,* $\exists$ *poly-query* $\mathcal{A}^{\mathsf{F},\mathsf{INV},\mathsf{PSPACE}}$*,* $\exists$*constant* $c > 0$ *s.t.*

$$\Pr_{(\mathsf{F},\mathsf{INV})\leftarrow\$D_{\mathbf{p}}}\left[\Pr_{s,\mathcal{A}}\left[\mathcal{A}^{\mathsf{F},\mathsf{INV},\mathsf{PSPACE}} \textit{ inverts } g(s)\right] \leq c\right] = \mathsf{constant} < 1$$

This implies that

$$\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\textit{Case 2 of } \mathsf{Bad}^{\mathcal{R},\mathcal{A},g}_m\right] = \mathsf{constant} < 1$$

where $\mathcal{D} := D_{\mathbf{p}}$*,* $\mathcal{O}_1 := \mathsf{F},\mathsf{PSPACE}$*,* $\mathcal{O}_2 := \mathsf{INV}$ *and* $\mathsf{Bad}^{\mathcal{R},\mathcal{A},g}_m$ *is defined as in Theorem 12.*

Interestingly, Theorem 17 does not depend on the number of calls to F in the strong OWF construction g. That is, if the input length of the construction g is too short, then no number of calls to F can make it a strong OWF. Now, let $\mathbf{p}(m)$ be a fixed polynomial. We now start by proving Theorem 17 for constructions which make few F-queries, and later, we prove the theorem for larger number of queries. More precisely, we show that no matter what the input length to g is, for every constant c, g must make at least $l > c \cdot \mathbf{p}(m)$ calls to F, otherwise all the F calls are easy with constant probability, which makes inverting g trivial.

Proposition 18 (Easy inversion if few F-Calls). *Consider a NANPP* $g = (\mathsf{pre}, \mathsf{post})$ *with* $\mathsf{post}(y_1, .., y_l, d) = y_1||..||y_l||d$. *For all constants* c, *if* $\mathsf{pre}(s) = (x_1, .., x_l, d)$ *induces at most* $l \leq c\mathbf{p}(m)$ *(parallel) calls to* F, *then all* $y_i := \mathsf{F}(x_i)$ *are in* $\mathsf{EASY}^m_{\mathsf{out}}$ *with constant probability, more precisely,* $\exists$*constant* > 0 *s.t.*

$$\Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[\Pr_s[\forall y_i \in g(s) : y_i \in \mathsf{EASY}^m_{\mathsf{out}}] > \textit{constant}] > \textit{constant} > 0 \tag{4}$$

In particular, with constant probability over the choice of the oracle F, g *can be inverted with non-negligible (constant) probability by a poly-query adversary.*

Proof. Suppose there are $l \leq c\mathbf{p}(m)$ parallel calls to F. Denote by $y_1, ..., y_l$ the outputs of the parallel calls to F. Now, when considering the randomness of choosing $\mathsf{EASY}^m_{\mathsf{in}}$, we have

$$\begin{aligned}
&\Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}},s}[y_1, ..., y_l \in \mathsf{EASY}^m_{\mathsf{out}}] \\
&\geq \underbrace{\sum_s 2^{-|s|}}_{=1} \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[y_1 \in \mathsf{EASY}^m_{\mathsf{out}} \mid s] \cdot ... \cdot \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[y_l \in \mathsf{EASY}^m_{\mathsf{out}} \mid s] \\
&= \left(1 - \frac{1}{\mathbf{p}(m)}\right)^l \geq \left(1 - \frac{1}{\mathbf{p}(m)}\right)^{c\mathbf{p}(m)} \geq \left(\frac{1}{4}\right)^c \ \forall\, \mathbf{p}(m) > 2.
\end{aligned}$$

where the first inequality is an equality iff $y_i \neq y_j \forall i \neq j$ and the second inequality follows since $(1 - \frac{1}{x})^x$ converges monotonously to $\frac{1}{e}$ and is greater than $\frac{1}{4}$ whenever $x \geq 2$. Now since $\left(\frac{1}{4}\right)^c$ is constant, we can use a simple averaging argument to prove (4): namely, we show

$$\Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}\left[\Pr_s[y_1, ..., y_l \in \mathsf{EASY}^m_{\mathsf{out}}] > 1/4^{c+1}\right] > 1/4^c - 1/4^{c+1} \tag{5}$$

as follows:

$$\begin{aligned}
1/4^c &\leq \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}},s}[y_1, ..., y_l \in \mathsf{EASY}^m_{\mathsf{out}}] \\
&= \sum_{\mathsf{F}} \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[\mathsf{F}] \underbrace{\Pr_s[y_1, ..., y_l \in \mathsf{EASY}^m_{\mathsf{out}} | \mathsf{F}]}_{:=a_{\mathsf{F}}} \\
&\leq \sum_{\mathsf{F} \text{ s.t. } a_{\mathsf{F}} > 1/4^{c+1}} \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[\mathsf{F}] + \sum_{\mathsf{F} \text{ s.t. } a_{\mathsf{F}} \leq 1/4^{c+1}} \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}[\mathsf{F}] 1/4^{c+1} \\
&\leq \Pr_{\mathsf{F}\leftarrow\$D_{\mathbf{p}}}\left[\mathsf{F} \text{ s.t. } a_{\mathsf{F}} > 1/4^{c+1}\right] + 1/4^{c+1}
\end{aligned}$$

which proves (5).

In the case where all $y_1, ..., y_l$ are all easy, $\mathcal{A}$ can invert $y_1, ..., y_l$ using INV oracle. Note that there is only a single pre-image x_i per y_i and thus, given the list $x_1, ..., x_l$, $\mathcal{A}$ can use the PSPACE oracle to find an s such that $\mathsf{pre}(s) = x_1, ..., x_l$.

□

Due to Proposition 18, for the remainder of this section, we can focus on constructions where pre makes more than $c \cdot \mathbf{p}(m)$ calls. Also in the case where g makes many queries, we can always invert the easy fraction of $(y_1, .., y_l)$. However, if many queries are made, then (with high probability) some y_i will also be hard. Of course, if pre-processing $\mathsf{pre}(s) = (x_1, .., x_l)$ distributes the entropy well, then knowing some of the x_i might suffice to restrict the set of suitable candidate values s to a polynomial-sized set, and once a polynomial-sized set of candidates is obtained, a random candidate s is a suitable pre-image with high enough probability. How well does this strategy work when considering *arbitrary* pre-processing pre?

To analyze this strategy, we study the entropy of the hard values x_i given $(1 - \frac{1}{\mathbf{p}(m)})\ell$ many easy values x_i (note that in expectation, $(1 - \frac{1}{\mathbf{p}(m)})\ell$ many values are easy) and seek to prove that their entropy is low. Towards that goal, we look at the entropy of the $\frac{1}{\mathbf{p}(m)}\ell$ many first x_i under a fixed permutation π and given the $\ell - \frac{1}{\mathbf{p}(m)}\ell$ many last x_i under that permutation. That is, we are interested in the entropy

$$h(\pi) := H(X_{\pi(1)}, \ldots, X_{\pi(\frac{\ell}{\mathbf{p}(m)})} | X_{\pi(\frac{\ell}{\mathbf{p}(m)}+1)}, \ldots, X_{\pi(\ell)}),$$

where X_i is the random variable defined as follows: sample a uniformly random s from $\{0,1\}^n$, compute $\mathsf{pre}(s)$ and take the ith output (i.e. the input to the ith F-call in g). We now prove that the expectation of entropy $h(\pi)$ is small. Lemma 19 (Small Entropy Expectation) is our main conceptual lemma.

Lemma 19 (Small Entropy Expectation). *Suppose* $\mathbf{p}(m)$ *divides* ℓ. *Then,*

$$\mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}[h(\pi)] \leq \frac{n}{\mathbf{p}(m)},$$

which is equivalent to

$$\mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[H(X_{\pi(1)}, \ldots, X_{\pi(\frac{\ell}{\mathbf{p}(m)})} | X_{\pi(\frac{\ell}{\mathbf{p}(m)}+1)}, \ldots, X_{\pi(\ell)})\right] \leq \frac{n}{\mathbf{p}(m)}. \tag{6}$$

Proof. Let's consider a permutation π of the weak OWF inputs $x_{\pi(1)}, ..., x_{\pi(\ell)}$. Let's divide the inputs x_i into $\mathbf{p}(m)$ equal-sized blocks as follows:

$$\underbrace{x_{\pi(1)}, ..., x_{\pi(\frac{\ell}{\mathbf{p}(m)})}}_{\text{one block}}, \underbrace{x_{\pi(\frac{\ell}{\mathbf{p}(m)}+1)}, ..., x_{\pi(2\frac{\ell}{\mathbf{p}(m)})}}_{\text{one block}}, x_{\pi(2\frac{\ell}{\mathbf{p}(m)}+1)}, ..., x_{\pi(\ell)}$$

Each pink index starts a new block. Let's denote the set of the pink indices by

$$J := \left\{1, \tfrac{\ell}{\mathbf{p}(m)} + 1, 2\tfrac{\ell}{\mathbf{p}(m)} + 1, ..., (\mathbf{p}(m) - 1)\tfrac{\ell}{\mathbf{p}(m)} + 1\right\}.$$

Now, let S denote the seed random variable of length n and consider the following sum

$$\sum_{j \in J} \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[H\left(\underbrace{X_{\pi(j)}, \ldots, X_{\pi(j+\frac{\ell}{\mathbf{p}(m)}-1)}}_{j\text{-th block}} | \underbrace{X_{\pi(j+\frac{\ell}{\mathbf{p}(m)})}, \ldots, X_{\pi(\ell)}}_{\text{all } X_i \text{ after the } j\text{-th block}}\right)\right] \tag{7}$$

$$= \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[\sum_{j \in J} H\left(X_{\pi(j)}, \ldots, X_{\pi(j+\frac{\ell}{\mathbf{p}(m)}-1)} | X_{\pi(j+\frac{\ell}{\mathbf{p}(m)})}, \ldots, X_{\pi(\ell)}\right)\right] \tag{8}$$

$$= \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[H\left(X_{\pi(1)}, \ldots, X_{\pi(\ell)}\right)\right] \tag{9}$$

$$\leq \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}[H(S)] \tag{10}$$

$$= n \tag{11}$$

where (8) holds by linearity of expectation, (9) holds by the chain rule for entropy (Lemma 7) and Inequality (10) holds because $(X_1, .., X_\ell)$ are computed by applying the deterministic function pre on S, and applying a deterministic function cannot increase entropy—the Inequality becomes equality if and only if pre is injective. Finally, Equality (11) follows since $H(S) = |S| = n$.

Now, from (7)-(11), we have

$$n \geq \sum_{j \in J} \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[H\left(X_{\pi(j)}, .., X_{\pi(j+\frac{\ell}{\mathbf{p}(m)}-1)} | X_{\pi(j+\frac{\ell}{\mathbf{p}(m)})}, .., X_{\pi(\ell)}\right)\right]$$

$$\geq \sum_{j \in J} \mathbb{E}_{\pi \leftarrow\$ \Pi(\ell)}\left[H\left(X_{\pi(j)}, .., X_{\pi(j+\frac{\ell}{\mathbf{p}(m)}-1)} | X_{\pi(i)}, i = 1, .., j-1, j+\tfrac{\ell}{\mathbf{p}(m)}, .., \ell\right)\right] \tag{12}$$

$$= \sum_{j \in J} \mathbb{E}_{\pi' \leftarrow\$ \Pi(\ell)}\left[H\left(X_{\pi'(1)}, .., X_{\pi'(\frac{\ell}{\mathbf{p}(m)})} | X_{\pi'(\frac{\ell}{\mathbf{p}(m)}+1)}, .., X_{\pi'(\ell)}\right)\right] \tag{13}$$

$$= \mathbf{p}(m)\, \mathbb{E}_{\pi' \leftarrow\$ \Pi(\ell)}\left[H\left(X_{\pi'(1)}, .., X_{\pi'(\frac{\ell}{\mathbf{p}(m)})} | X_{\pi'(\frac{\ell}{\mathbf{p}(m)}+1)}, .., X_{\pi'(\ell)}\right)\right] \tag{14}$$

where (12) follows since conditioning on more random variables can only decrease entropy. Here, we additionally condition on all $X_{\pi(i)}$ for $i < j$ and not only on those for $i \geq j + \frac{l}{\mathbf{p}(m)}$. In Equation (13) we change to a more convenient indexing where we perform a bijective mapping on all permutations $\pi'(1) = \pi(j)$,...,$\pi'(\frac{l}{\mathbf{p}(m)}) = \pi(j + \frac{l}{\mathbf{p}(m)} - 1)$ thus maintaining the same distribution over all permutations. Since the summands do not depend on j anymore and $|J| = \mathbf{p}(m)$, Equation (14) follows which concludes the proof of Lemma 19.

□

With Lemma 19 at hand, we now turn to the proof of Theorem 17.

Proof of Theorem 17. Let g be a (n,m)-NANPP g with input length $n \leq \mathbf{dp}(m)$ and let ℓ be the number of queries to F which g makes. The adversary $\mathcal{A}$ (described on the right) now tries to invert all y_1,..,y_ℓ using INV and puts a placeholder $\perp$ as x_i when inversion fails. $\mathcal{A}$ then computes a random pre-image of the pre-processing that matches the known x_is and d, which is possible in polynomial-time using the PSPACE oracle. We now argue that a random pre-image of the pre-processing, that matches the known x_is and d, is an actual preimage of $y_1||...||y_\ell||d$ under g with constant probability.

$\mathcal{A}(y_1||...||y_\ell||d)$
for $i \in 1,...,\ell$
  $x_i \leftarrow \mathsf{INV}(y_i)$
$s \leftarrow\!\!\$\ \mathsf{pre}^{-1}(x_1,...,x_\ell,d)$
return s

From now on, we assume that $|d| = 0$. This is w.l.o.g. because the data d is known to the adversary, so it cannot add entropy. Further and also w.l.o.g., we assume that $\mathbf{p}(m)$ divides ℓ for all m,n (if there was a remainder, we could add constant dummy F-calls until there is no remainder. Such F-calls would not make g weaker nor stronger, so our result would still hold.) Note that if $\ell \leq \mathbf{p}(m)$, then with constant probability all x_i are easy and INV inverts all of them (cf. Theorem 18). In that case $\mathcal{A}$ can use the PSPACE oracle to find a correct preimage s with probability 1. Hence, we can assume that $\ell > \mathbf{p}(m)$.

First, recall that $h(\pi)$ denotes the entropy of the hard values conditioned on knowing the easy values, i.e., $h(\pi)$ is equal to

$$H(X_{\pi(1)},\ldots,X_{\pi(\frac{\ell}{\mathbf{p}(m)})}|X_{\pi(\frac{\ell}{\mathbf{p}(m)}+1)},\ldots,X_{\pi(\ell)}).$$

Now, Lemma 19 (Small Entropy Expectation) established that the expectation of entropy $h(\pi)$ is small:

$$\mathbb{E}_{\pi\leftarrow\!\!\$\Pi(\ell)}[h(\pi)] \leq \frac{n}{\mathbf{p}(m)} < \mathbf{d}, \tag{15}$$

where we use that Theorem 17 assumes that $\mathbf{dp}(m) \geq n$. We now want to lower bound the probability that the remaining entropy $h(\pi)$ is less than $\frac{2n}{\mathbf{p}(m)}$. We call such a permutation π *good* and next we lower bound the probability of permutation π being good. By Markov's inequality

$$\mathbb{E}_{\pi\leftarrow\!\!\$\Pi(\ell)}[h(\pi)] \geq \frac{2n}{\mathbf{p}(m)} \cdot \Pr_{\pi\leftarrow\!\!\$\Pi(\ell)}\left[h(\pi) \geq \frac{2n}{\mathbf{p}(m)}\right].$$

Equivalently,

$$\Pr_{\pi\leftarrow\!\!\$\Pi(\ell)}\left[h(\pi) \geq \frac{2n}{\mathbf{p}(m)}\right] \leq \frac{\mathbf{p}(m)}{2n} \cdot \mathbb{E}_{\pi\leftarrow\!\!\$\Pi(\ell)}[h(\pi)] \overset{(15)}{\leq} \frac{\mathbf{p}(m)}{2n} \cdot \frac{n}{\mathbf{p}(m)} = \frac{1}{2}.$$

Hence,

$$\Pr_{\pi\in\Pi(\ell)}\left[h(\pi) < \frac{2n}{\mathbf{p}(m)}\right] \geq \frac{1}{2}. \tag{16}$$

Now we know that a permutation π is good with probability at least $\frac{1}{2}$. If the permutation π is good, then the remaining entropy of the input is small and thus, some inputs are very likely (cf. Lemma 21 (Predictable Inputs) in Appendix A) and thus likely chosen by adversary $\mathcal{A}$ which chooses a random pre-image amongst the possible candidates. With this intuition of the proof in mind, we can now lower-bound the probability of $\mathcal{A}$'s success formally.

$$\begin{aligned}
&\Pr_{\mathsf{F},s}[\mathcal{A} \text{ inverts } g(s)] \\
&\geq \Pr_{\mathsf{F}}\left[\exists\pi : x_{\pi(1)},...,x_{\pi((1-\frac{1}{\mathbf{p}(m)})\ell)} \in \mathsf{EASY}^m_{\mathsf{in}}\right] \\
&\quad \cdot \Pr_s\left[\mathcal{A} \text{ inverts } g(s) \;\middle|\; \exists\pi : x_{\pi(1)},...,x_{\pi((1-\frac{1}{\mathbf{p}(m)})\ell)} \in \mathsf{EASY}^m_{\mathsf{in}}\right] \\
&\geq \frac{1}{2}\Pr_s\left[\mathcal{A} \text{ inverts } g(s) \;\middle|\; \underbrace{\exists\pi : x_{\pi(1)},...,x_{\pi((1-\frac{1}{\mathbf{p}(m)})\ell)} \in \mathsf{EASY}^m_{\mathsf{in}}}_{:=B}\right]
\end{aligned} \tag{17}$$

$$\geq \frac{1}{2}\Pr_s\Bigg[\underbrace{\mathrm{H}\Big(X_{\pi(1)},\ldots,X_{\pi(\frac{\ell}{\mathbf{p}(m)})}\;\Big|\;X_{\pi(\frac{\ell}{\mathbf{p}(m)}+1)},\ldots,X_{\pi(\ell)}\Big) < \frac{2n}{\mathbf{p}(m)}}_{=:C}\,|B\Bigg] \tag{18}$$

$$\cdot\Pr_s\left[\Pr_{s'}\left[\begin{array}{l|l}\forall k\in{}_{\pi(1),\ldots,\pi(\ell/\mathbf{p}(m))}: & \forall j \in \\ \mathsf{pre}(s')_k=\mathsf{pre}(s)_k & {}_{\pi(\ell/\mathbf{p}(m)+1),\ldots,\pi(\ell)}: \\ & \mathsf{pre}(s')_j=\mathsf{pre}(s)_j\end{array}\right] > \frac{1}{2^{2\mathbf{d}+1}}\;|\,C,B\right] \tag{19}$$

$$\cdot\Pr_s\left[\mathcal{A}\text{ inverts }g(s)\;\middle|\;\Pr_{s'}\left[\begin{array}{l|l}\forall k\in{}_{\pi(1),\ldots,\pi(\ell/\mathbf{p}(m))}: & \forall j \in \\ \mathsf{pre}(s')_k=\mathsf{pre}(s)_k & {}_{\pi(\ell/\mathbf{p}(m)+1),\ldots,\pi(\ell)}: \\ & \mathsf{pre}(s')_j=\mathsf{pre}(s)_j\end{array}\right] > \frac{1}{2^{2\mathbf{d}+1}}\;\wedge\;C,B\right] \tag{20}$$

$$\geq \frac{1}{2}\cdot\frac{1}{2}\cdot\left(1-\frac{2\cdot\mathbf{d}}{2\mathbf{d}+1}\right)\cdot\frac{1}{2^{2\mathbf{d}+1}} = \text{constant} \tag{21}$$

where (17) follows from the fact that whether x_i is easy or not follows binomial distribution with $(1-\frac{1}{\mathbf{p}(m)})\ell$ many easy values in expectation. Inequality (18) uses chain rule of probability. The fractions at (21) follow from the lemmas, namely, the probability on line (18) is less than $1/2$ by Lemma 19 (Small Entropy Expectation) and probability on line (19) is less than $1-\frac{2\cdot\mathbf{d}}{2\mathbf{d}+1}$ by Lemma 21 (Predictable Inputs). The last fraction follows from the definition of adversary $\mathcal{A}$ and the probability statement at (20). Namely, if adversary guesses a random s' which is consistent with the known x_i, and we condition the probability on such s' being correct $\frac{1}{2^{2\mathbf{d}+1}}$ of the time, adversary must be right $\frac{1}{2^{2\mathbf{d}+1}}$ of the time.

Now that we know that

$$\Pr_{\mathsf{F},s}[\mathcal{A}\text{ inverts }g(s)] \geq \text{const} > 0,$$

we can use an averaging argument to show that $\Pr_{\mathsf{F}}[\Pr_s[\mathcal{A}\text{ inverts }g(s)] > \text{const} > 0] \geq \text{const} > 0$: Suppose $\Pr_{\mathsf{F},s}[\mathcal{A}\text{ inverts }g(s)] \geq c_1 > 0$. Now

$$\begin{aligned} c_1 \leq \Pr_{\mathsf{F},s}[\mathcal{A}\text{ inverts }g(s)] &= \sum_{\mathsf{F}}\Pr_{\mathsf{F}}[\mathsf{F}]\underbrace{\Pr_s[\mathcal{A}\text{ inverts }g(s)\;|\,\mathsf{F}]}_{=:a_{\mathsf{F}}} \\ &= \sum_{\mathsf{F}\text{ s.t. }a_{\mathsf{F}}>c_1/2}\Pr_{\mathsf{F}}[\mathsf{F}]\underbrace{\Pr_s[\mathcal{A}\text{ inverts }g(s)\;|\,\mathsf{F}]}_{\leq 1} + \underbrace{\sum_{\mathsf{F}\text{ s.t. }a_{\mathsf{F}}\leq c_1/2}\Pr_{\mathsf{F}}[\mathsf{F}]\Pr_s[\mathcal{A}\text{ inverts }g(s)\;|\,\mathsf{F}]}_{\leq 1} \\ &\leq \Pr_{\mathsf{F}}[a_{\mathsf{F}} > c_1/2] + c_1/2 \end{aligned}$$

Since we established that indeed, $\Pr_{\mathsf{F}}[\Pr_s[\mathcal{A}\text{ inverts }g(s)] > c_1/2] \geq c_1/2$, where c_1 is a constant >0, this conclude the proof of Theorem 17. □

Theorem 12 follows from the Theorems 17 and 16 by union bound, namely

$$\begin{aligned}\Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\left[\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] &= \Pr\left[\text{Case 1 of }\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\text{ or Case 2 of }\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\right] \\ &\leq 1/\mathbf{c} + \underbrace{\text{constant from Theorem 17}}_{\leq 1-c_1/2\text{ i.e. prob. that }g\text{ is secure-ish}} < 1\end{aligned}$$

Note that since the constant $\mathbf{c}$ in Theorem 16 can be made arbitrarily large, in particular, it can be chosen s.t. $1/\mathbf{c}+\text{constant from Theorem 17}$ is < 1.

6 Constructions with post-processing

In this section, we prove Theorem 13. Towards this goal, we use the oracles F,INV and PSPACE (cf. Section 4), and show that there are no short input NAIPP constructions under the oracles.

Theorem 20 (No Strong OWFs with Injectiveish Post-Processing). *$\forall$ poly $\mathbf{p}$, $\forall(n,m)$-NAIPP g with input length $n \leq \frac{1}{4}\mathbf{p}(m)$, $\exists$poly $q(n)=n^c, c\in\mathbb{N}_+$, $\exists$ poly-query $\mathcal{A}^{\mathsf{F},\mathsf{INV},\mathsf{PSPACE}}$ such that*

$$\Pr_{(\mathsf{F},\mathsf{INV})\leftarrow\$D_{\mathbf{p}}}\left[\Pr_{s,\ coins\ of\ \mathcal{A}}\left[\mathcal{A}^{\mathsf{F},\mathsf{INV},\mathsf{PSPACE}}\ inverts\ g(s)\right] \leq q(n)\right] = \mathsf{constant} < 1$$

$$\textit{and thus } \Pr_{(\mathcal{O}_1,\mathcal{O}_2)\leftarrow\$\mathcal{D}}\Big[\textit{Case 2 of } \mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}\Big] = \mathsf{constant} < 1$$

where $\mathsf{Bad}_m^{\mathcal{R},\mathcal{A},g}$ *is defined as in Theorem 13.*

Theorems 16 and 20 together imply Theorem 13 by union bound analogously to the NANPP case. It thus remains to prove Theorem 20.

Proof. Let g be (n,m)-NAIPP which makes ℓ queries to F and let $\mathcal{A}$ be the adversary on the right which samples a uniformly random pre-image of z under post, then inverts the easy queries and returns a seed s which is consistent with the pre-image of the easy values. Firstly observe that $\mathcal{A}$ runs in polynomial-time since it can use the PSPACE oracle for inverting post. Moreover, it makes only a polynomial number of queries since ℓ is a polynomial.

$\mathcal{A}(z)$

$y_1, ..., y_\ell, d \leftarrow \mathsf{post}^{-1}(z)$
for $i \in 1, ..., \ell$
  $x_i \leftarrow \mathsf{INV}(y_i)$
$s \leftarrow\$ \mathsf{pre}^{-1}(x_1, ..., x_\ell, d)$
return s

As the post-processing of g is almost injective, $y_1, ..., y_\ell, d \leftarrow\$ \mathsf{post}^{-1}(z)$ returns the values $y_1, ..., y_\ell, d$ which the one-wayness experiment used to compute z with probability $\frac{1}{\mathsf{poly}(n)}$. This probability is independent of F. If $y_1, ..., y_\ell, d$ are indeed the correct values, then adversary $\mathcal{A}$ also finds a pre-image s with constant probability by the same arguments as in Theorem 17. Thus, the overall success of $\mathcal{A}$ is $\frac{1}{\mathsf{poly}(n)} \cdot$constant which is inverse polynomial as required by Theorem 20. □

Acknowledgments

We thank the anonymous reviewers for valuable comments. Parts of this work have been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1119 – 236615297 and by the Academy of Finland.

References

AMN+18. N. Attrapadung, T. Matsuda, R. Nishimaki, S. Yamada, and T. Yamakawa. Constrained PRFs for NC^1 in traditional groups. pages 543–574, 2018.
BBF13. P. Baecher, C. Brzuska, and M. Fischlin. Notions of black-box reductions, revisited. pages 296–315, 2013.
BFL91. L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational complexity*, 1(1):3–40, 1991.
BFNW93. L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. Bpp has subexponential time simulations unlessexptime has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.
BSV96. C. Blundo, A. D. Santis, and U. Vaccaro. Randomness in distribution protocols. *Inf. Comput.*, 131(2):111–139, 1996.
CRS+07. R. Canetti, R. L. Rivest, M. Sudan, L. Trevisan, S. P. Vadhan, and H. Wee. Amplifying collision resistance: A complexity-theoretic treatment. pages 264–283, 2007.
DGI+19. N. Döttling, S. Garg, Y. Ishai, G. Malavolta, T. Mour, and R. Ostrovsky. Trapdoor hash functions and their applications. pages 3–32, 2019.
GGK03. R. Gennaro, Y. Gertner, and J. Katz. Lower bounds on the efficiency of encryption and digital signature schemes. pages 417–425, 2003.
GIL+90. O. Goldreich, R. Impagliazzo, L. A. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. pages 318–326, 1990.
GNW95. O. Goldreich, N. Nisan, and A. Wigderson. On yao's xor lemma. Technical Report TR95–050, Electronic Colloquium on Computational Complexity, 1995.
GOR11. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. pages 182–200, 2011.
GT00. R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. pages 305–313, 2000.
HHR06. I. Haitner, D. Harnik, and O. Reingold. On the power of the randomized iterate. In *Advances in Cryptology - CRYPTO 2006*, pages 22–40, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
HILL99. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. 28(4):1364–1396, 1999.
HLO12. B. Hemenway, S. Lu, and R. Ostrovsky. Correlated product security from any one-way function. pages 558–575, 2012.
HR04. C.-Y. Hsiao and L. Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? pages 92–105, 2004.

HRV10. I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. pages 437–446, 2010.
HVV04. A. Healy, S. P. Vadhan, and E. Viola. Using nondeterminism to amplify hardness. pages 192–201, 2004.
IKNP03. Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. pages 145–161, 2003.
Imp95. R. Impagliazzo. Hard-core distributions for somewhat hard problems. pages 538–545, 1995.
IR89. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. pages 44–61, 1989.
IR90. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. pages 8–26, 1990.
IW97. R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. pages 220–229, 1997.
KST99. J. H. Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. pages 535–542, 1999.
Lip91. R. Lipton. New directions in testing. *Distributed computing and cryptography*, 2:191–202, 1991.
LTW05. H. Lin, L. Trevisan, and H. Wee. On hardness amplification of one-way functions. pages 34–49, 2005.
Lu06. C.-J. Lu. On the complexity of parallel hardness amplification for one-way functions. pages 462–481, 2006.
Lu09. C.-J. Lu. On the security loss in cryptographic reductions. pages 72–87, 2009.
MMN+16. M. Mahmoody, A. Mohammed, S. Nematihaji, R. Pass, and a. shelat. A note on black-box separations for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2016/316, 2016. `https://eprint.iacr.org/2016/316`.
RS09. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. pages 419–436, 2009.
RTV04. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. pages 1–20, 2004.
Sim98. D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? pages 334–345, 1998.
STV01. M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the xor lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.
SV08. R. Shaltiel and E. Viola. Hardness amplification proofs require majority. pages 589–598, 2008.
Tre03. L. Trevisan. List-decoding using the XOR lemma. pages 126–135, 2003.
Tre05. L. Trevisan. On uniform amplification of hardness in NP. pages 31–38, 2005.
Vio05. E. Viola. The complexity of constructing pseudorandom generators from hard functions. *computational complexity*, 13(3-4):147–188, 2005.
VZ12. S. P. Vadhan and C. J. Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. pages 817–836, 2012.
Wee07. H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. pages 419–433, 2007.
Wic13. D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. pages 111–126, 2013.
Yao82. A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). pages 80–91, 1982.

A Additional Lemmas and Proofs

Lemma 21 (Predictable Inputs). *Consider* $(X_1,..,X_\ell) = \mathsf{pre}(s)$ *for uniformly random* s*. If (a fixed) permutation* π *is good, i.e. such that the entropy of the* $X_{\pi(i)}$ *satisfies*

$$\mathrm{H}\Big(X_{\pi(1)},\ldots,X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}\right)} \;\Big|\; X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}+1\right)},\ldots,X_{\pi(\ell)}\Big) < \frac{2n}{\mathbf{p}(m)}$$

then

$$\Pr_s\left[\Pr_{s'}\left[\begin{array}{c}\forall k=\pi(i), i=1,..,\frac{\ell}{\mathbf{p}(m)}: \\ \mathsf{pre}(s')_k=\mathsf{pre}(s)_k\end{array}\;\middle|\;\begin{array}{c}\forall j=\pi(i), i=\frac{\ell}{\mathbf{p}(m)}+1,..,\ell: \\ \mathsf{pre}(s')_j=\mathsf{pre}(s)_j\end{array}\right] > \frac{1}{2^{2\mathbf{d}+1}}\right] \geq 1-\frac{2\cdot\mathbf{d}}{2\mathbf{d}+1}$$

Proof. Since $n \leq \mathbf{dp}(m)$, we get that

$$\mathrm{H}\Big(X_{\pi(1)},\ldots,X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}\right)} \;\Big|\; X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}+1\right)},\ldots,X_{\pi(\ell)}\Big) < \frac{2n}{\mathbf{p}(m)} \leq 2\cdot\mathbf{d} \tag{22}$$

Let $S_{h,e} \subseteq \{0,1\}^m$ be defined as

$$S_{h,e} = \{s : \Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] < \frac{1}{2^{2\mathbf{d}+1}}\},$$

where

$$p_e(s) := \mathsf{pre}(s)_{\pi\left(\frac{\ell}{\mathbf{p}(m)}+1\right)}, ..., \mathsf{pre}(s)_{\pi(\ell)}$$

and

$$p_h(s) := \mathsf{pre}(s)_{\pi(1)}, \ldots, \mathsf{pre}(s)_{\pi\left(\frac{\ell}{\mathbf{p}(m)}\right)}.$$

Using (22) and the definition of conditional Shannon entropy, we get that

$$\begin{aligned}
2 \cdot \mathbf{d} &> \mathrm{H}\left(\underbrace{X_{\pi(1)}, \ldots, X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}\right)}}_{=:p_h(s)} \,\middle|\, \underbrace{X_{\pi\left(\frac{\ell}{\mathbf{p}(m)}+1\right)}, \ldots, X_{\pi(\ell)}}_{=:p_e(s)}\right) \\
&= \sum_{s \in \{0,1\}^m} \Pr_{s'}[p_h(s') = p_h(s) \textbf{ and } p_e(s') = p_e(s)] \cdot |\log \Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)]| \\
&= \sum_{s \in S_{h,e}} \Pr_{s'}[p_h(s') = p_h(s) \textbf{ and } p_e(s') = p_e(s)] \cdot |\log \Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)]| \\
&\quad + \sum_{s \notin S_{h,e}} \Pr_{s'}[p_h(s') = p_h(s) \textbf{ and } p_e(s') = p_e(s)] \cdot |\log \Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)]| \\
&\geq \left(\sum_{s \in S_{h,e}} \Pr_{s'}[p_h(s') = p_h(s) \textbf{ and } p_e(s') = p_e(s)]\right) \cdot \left|\log \frac{1}{2^{2\mathbf{d}+1}}\right| \\
&\quad + \left(\sum_{s \notin S_{h,e}} \Pr_{s'}[p_h(s') = p_h(s) \textbf{ and } p_e(s') = p_e(s)]\right) \cdot |\log 1| \\
&\geq \Pr_s\left[\Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] < \frac{1}{2^{2\mathbf{d}+1}}\right] \cdot (2\mathbf{d}+1) + 0
\end{aligned}$$

where log is the base-2 logarithm and the last inequality is equality exactly when pre-processing is injective. Now

$$\begin{aligned}
& 2 \cdot \mathbf{d} \geq \Pr_s\left[\Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] < \frac{1}{2^{2\mathbf{d}+1}}\right] \cdot (2\mathbf{d}+1) \\
\Leftrightarrow \quad & \frac{2 \cdot \mathbf{d}}{2\mathbf{d}+1} \geq \Pr_s\left[\Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] < \frac{1}{2^{2\mathbf{d}+1}}\right] \\
\Rightarrow & \Pr_s\left[\Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] \geq \frac{1}{2^{2\mathbf{d}+1}}\right] \\
& \quad = 1 - \Pr_s\left[\Pr_{s'}[p_h(s') = p_h(s) \mid p_e(s') = p_e(s)] < \frac{1}{2^{2\mathbf{d}+1}}\right] \\
& \quad > 1 - \frac{2 \cdot \mathbf{d}}{2\mathbf{d}+1}
\end{aligned}$$

which proves the statement.

□

B Proof of Theorem 16 (F is a weak OWF)

In order to prove Theorem 16, we show that F is a weak OWF with inversion probability $1 - 1/2\mathbf{p}(m)$ with all but small constant probability over $D_{\mathbf{p}}$. Namely, we show that for all polynomials $\mathbf{p}$, for

all poly-query $\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}$, for all adversaries $\mathcal{R}$ making polynomially many (in m) queries to the oracles $\mathsf{F}, \mathsf{PSPACE}, \mathsf{INV}, \mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}$, the probability

$$\Pr_{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}\leftarrow\$D_{\mathbf{p}}}\left[\mathsf{SuccInv}^{\mathsf{F},\mathsf{INV}}_{\mathcal{A},\mathcal{R}} \geq 1 - \frac{1}{2\mathbf{p}(m)}\right] \leq 1/\mathbf{c}, \tag{23}$$

where $\mathsf{SuccInv}^{\mathsf{F},\mathsf{INV}}_{\mathcal{A},\mathcal{R}}$ is defined as

$$\Pr_{x\leftarrow\$\{0,1\}^m,\mathcal{R},\mathcal{A}}\left[\mathcal{R}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV},\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}}(1^m,\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))\right].$$

Proof of Theorem 16. Fix $\mathbf{p}$, $\mathcal{R}$ and $\mathcal{A}$. Since $\mathcal{A}$ and $\mathcal{R}$ both make polynomially many queries to the same oracles, $\mathcal{R}$ can simply simulate $\mathcal{A}$. Thus, w.l.o.g., we can assume that $\mathcal{R}$ only makes queries to F, PSPACE and INV. Additionally, we consider $\mathcal{R}$ to be a computationally unbounded algorithm so that w.l.o.g., we can assume that it does not make queries to the PSPACE oracle.

Let q be a polynomial such that adversary $\mathcal{R}$ makes exactly $q(m)$ queries to the oracle F and an arbitrary number of queries to INV. Since we let the adversary $\mathcal{R}$ make an arbitrary number of queries to INV, that is, the adversary can be assumed to know the sets $\mathsf{EASY}^m_{\mathsf{in}}$ and $\mathsf{EASY}^m_{\mathsf{out}}$ and how F maps $\mathsf{EASY}^m_{\mathsf{in}}$ to $\mathsf{EASY}^m_{\mathsf{out}}$ completely. This only makes the adversary stronger. Importantly, using INV does not give the adversary any information on F on the *hard* values (only the fact that the values are hard, but no information on how $\mathsf{HARD}^m_{\mathsf{in}}$ maps to $\mathsf{HARD}^m_{\mathsf{out}}$).

For conciseness, we now denote $\mathcal{R} := \mathcal{R}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV},\mathcal{A}^{\mathsf{F},\mathsf{PSPACE},\mathsf{INV}}}$ and we omit the PSPACE oracle everywhere—the PSPACE oracle is deterministic anyway. Moreover, we denote $\mathcal{R}$'s queries to F by $x_1, ..., x_{q(m)}$ and the $\mathcal{R}$'s guess for the pre-image of its input y by $x_{q(m)+1}$.

Eventually we want to bound the following:

$$\begin{aligned}
&\Pr_{\mathsf{F},\mathsf{INV}}\left[\mathsf{SuccInv}^{\mathsf{F},\mathsf{INV}}_{\mathcal{A},\mathcal{R}} \geq 1 - \tfrac{1}{2\mathbf{p}(m)}\right] \\
=&\Pr_{\mathsf{F},\mathsf{INV}}\left[\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))\right] \geq 1 - \tfrac{1}{2\mathbf{p}(m)}\right] \\
\leq&\Pr_{\mathsf{F},\mathsf{INV}}\left[\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \geq \tfrac{1}{2\mathbf{p}(m)}\right]
\end{aligned} \tag{24}$$

where the last inequality follows from

$$\begin{aligned}
&\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))\right] \\
&= \Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \underbrace{\Pr_{x,\mathcal{R}}[x \notin \mathsf{EASY}^m_{\mathsf{in}}]}_{<1} \\
&\quad + \underbrace{\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \in \mathsf{EASY}^m_{\mathsf{in}}\right]}_{\leq 1} \underbrace{\Pr_{x,\mathcal{R}}[x \in \mathsf{EASY}^m_{\mathsf{in}}]}_{=1-1/\mathbf{p}(m)}.
\end{aligned}$$

We first compute a bound when sampling oracles F, INV, input x and randomness for $\mathcal{R}$ *together* and then deduce (24) via averaging. In order to bound the probability of $\mathcal{R}$ successfully inverting, given that x is not in the easy set, we first bound the successful inversion event by the event that for any $i \leq q+1$, $\mathsf{F}(x_i) = \mathsf{F}(x)$ (and not just for x_{q+1}). Equation $(+)$ then splits the big OR into $q+1$ disjoint events.

$$\begin{aligned}
&\Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \\
\leq&\Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\bigvee_{1\leq i\leq q+1} \mathsf{F}(x_i) = \mathsf{F}(x) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \\
\overset{(+)}{=}&\sum_{i=1}^{q(m)+1} \Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\mathsf{F}(x_i) = \mathsf{F}(x) \;\middle|\; \begin{array}{l}\mathsf{F}(x_1), ..., \mathsf{F}(x_{i-1}) \neq \mathsf{F}(x) \wedge \\ x \notin \mathsf{EASY}^m_{\mathsf{in}}\end{array}\right] \\
&\qquad \cdot \Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}[\mathsf{F}(x_1), ..., \mathsf{F}(x_{i-1}) \neq \mathsf{F}(x) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}] \\
\leq&\sum_{i=1}^{q(m)+1} \Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\mathsf{F}(x_i) = \mathsf{F}(x) \;\middle|\; \begin{array}{l}\mathsf{F}(x_1), ..., \mathsf{F}(x_{i-1}) \neq \mathsf{F}(x) \wedge \\ x \notin \mathsf{EASY}^m_{\mathsf{in}}\end{array}\right]
\end{aligned} \tag{25}$$

Finally, the probabilities are maximized when all the x_i are distinct and happen to lie in $\{0,1\}^m \setminus \mathsf{EASY}^m_{\mathsf{in}}$ which is a set of size $\frac{1}{\mathbf{p}(m)}2^m$. The function F induces a bijection from $\{0,1\}^m \setminus \mathsf{EASY}^m_{\mathsf{in}}$ to $\{0,1\}^m \setminus \mathsf{EASY}^m_{\mathsf{out}}$ and thus, for the $i+1$-th query, we need to reduce the set size of $\{0,1\}^m \setminus \mathsf{EASY}^m_{\mathsf{in}}$ by i, since—in the worst case—we already know i values—note that the sum now goes from 0 to q instead of 1 to $q+1$. Hence, the probability that $\mathsf{F}(x_{i+1}) = \mathsf{F}(x)$ is upper bounded by 1 divided by $\frac{1}{\mathbf{p}(m)}2^m - i$. Thus, we can upper bound (25) by the sum

$$\Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \leq \sum_{i=0}^{q(m)} \frac{1}{\frac{1}{\mathbf{p}(m)}2^m - i}. \tag{26}$$

In order to derive an upper bound for (24), we need to split the probability into sampling F, INV and sampling $x, \mathcal{R}$.

By Markov's inequality, for any real number r,

$$\begin{aligned}
&\Pr_{\mathsf{F},\mathsf{INV}}\left[\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right] \geq r\right] \\
\leq &\frac{\mathbb{E}_{\mathsf{F},\mathsf{INV}}\left[\Pr_{x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right]\right]}{r} \\
= &\frac{\Pr_{\mathsf{F},\mathsf{INV},x,\mathcal{R}}\left[\mathcal{R}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x)) \mid x \notin \mathsf{EASY}^m_{\mathsf{in}}\right]}{r} && \text{by def of expectation} \\
\leq &\frac{\sum_{i=0}^{q(m)} \frac{1}{\frac{1}{\mathbf{p}(m)}2^m - i}}{r} && \text{by (26)}
\end{aligned}$$

Now we plug in the correct value $r = \frac{1}{2\mathbf{p}(m)}$ from (24) and get

$$\frac{\sum_{i=0}^{q(m)} \frac{1}{\frac{1}{\mathbf{p}(m)}2^m - i}}{\frac{1}{2\mathbf{p}(m)}} = \sum_{i=0}^{q(m)} \frac{2\mathbf{p}(m)}{\frac{1}{\mathbf{p}(m)}2^m - i} \leq \sum_{i=0}^{q(m)} \frac{2^{m/4}}{2^{m/2}} = q(m)2^{m/4} = \mathrm{negl}(m) < \frac{1}{\mathbf{c}}$$

where the first inequality holds for any big enough m and the last inequality holds for any big enough m since $\mathbf{c}$ is a constant.

□