Oded (October 13, 2023): Constructing PRG based on arbitrary OWF (following Mazor and Pass)

1 Introduction

The fact that any one-way function can be used to construct a pseudorandom generators is one of the most fundamental results of complexity theory and cryptography (see, e.g., [3, Sec. 8.2.5] and [2, Sec. 3.5], resp.). This result, proved by [10], improves on prior results of [1, 17] and [4] in which additional requirements were imposed on the one-way function. On the other hand, it has served as the starting point for a quest of simpler and more efficient constructions, where the main efficiency measures are the seed length (in terms of the length of inputs on which the one-way function is queried) and the number of such queries (see [11, 6, 7, 16, 8, 14]).

Indeed, all constructions are presented as algorithms that use the one-way function as a black box, or rather as oracle machines that make queries to (the forward direction of) the one-way function. Specifically, using an one-way function on n-bit strings, till recently, the best known complexity bounds are due to [16], which builds on [7]: It uses $\widetilde{O}(n^3)$ oracle calls and a seed of length $\widetilde{O}(n^3)$, where the $\widetilde{O}$-notation can actually be replaced by any efficiently computable and super-linear function. Expositions aimed at simpler proofs were provided in [8, 14]. Actually, the work of [14] also improves both complexity bounds to $o(n^3)$.

The current text provides an overview of the work of Mazor and Pass [14], which deviates from all prior works on the subject by using (next-bit) unpredictability rather than pseudorandomness (i.e., indistinguishability from random) as its pivot. Indeed, in their "pure" form, these notions are equivalent, but this equivalence does not seem to carry through to the relaxed forms. Specifically, *pseudoentropy* means being indistinguishable from a distribution of high entropy, whereas *quantitative unpredictablity* means that many bits (but not all) are unpredictable given the previous ones. (Indeed, both notions are quantitative and come with a suitable parameter (which quantifies how much is 'high' and how much is 'many').)[1]

Organization of the rest of this text. Our exposition proceeds as follows. First, we present the definition of quantitative unpredictablity. Next, we show that any one-way function yields a function that has more unpredictable bits than its input length. Finally, we show that a function with more unpredictable bits than its input length can be used to construct a pseudorandom generartor. We focus on providing an exposition in the model of non-uniform complexity (see Section 2). This model allows for the presentation of non-uniform security reductions, which simplifies the description of the hybrids used in the analysis. We later comment on the adaptation of these ideas to the uniform complexity setting (see Section 3).

How is this different from [7]? While the proof of [14] is analogous to that of [7], the crucial difference is that it abandons the notion of pseudorentropy and uses next-bit unpredictability instead. Indeed, the proof [7] takes a crucial step in this direction: It is pivoted on the notion of next-block pseudoentropy, where its real novelty (wrt prior work) is the next-block aspect. But, in my opinion, keeping pseudorentropy around blurs the actual issues, and leads to an argument that is somewhat more complicated both conceptually and technically.

[1] The notion of (next-bit) unpredictability may be viewed as a strengthening of the notion of next-block pseudoentropy (which was introduced and studied in [7]).

2 The non-uniform complexity model

As stated above, the key notion is that of a function having several unpredictable bits. That is, for some bit locations, but not necessarily all, predicting the next bit based on the prior ones is infeasible. These unpredictable bit locations may vary based on the input, but the definition provides a lower bound on their expected number (where the expectation is over the uniform choice of the input). Viewing n as a varying parameter, we let all other parameters depend on it.

Definition 1 (quantitative unpredictability, non-uniform version) *We say that* $g:\{0,1\}^{\ell(n)}\to\{0,1\}^{\ell'(n)}$ has $m(n)$ bits that are $(s(n),\epsilon(n))$-unpredictable *if there exists a function* $I:\{0,1\}^{\ell(n)}\to 2^{[\ell'(n)]}$ *such that* $\mathrm{E}_{x\in\{0,1\}^{\ell(n)}}[|I(x)|]\geq m(n)$ *and for every* i (in the support of some $I(x)$) *and for every* $s(n)$*-sized circuit* C_n *it holds that*

$$\mathrm{Pr}_{x\in\{0,1\}^{\ell(n)}}\left[C_n(g(x)_{[i-1]})=g(x)_i|I(x)\ni i\right]=0.5\pm\epsilon(n).$$

Although it may appear surprising (at least at first thought), it turns out that functions that have more unpredictable bits than their input length are relatively easy to construct based on any one-way function. Indeed, the analogous claim for any one-way permutation is well known (see [5]), but the following result refers to any one-way function.

Theorem 2 (obtaining more unpredictable bits than the input length): *Given a* (non-uniformly strong) *one-way function* $f:\{0,1\}^n\to\{0,1\}^n$*, consider the function*

$$g(M,x)\stackrel{\rm def}{=}(M,g'(M,x))=(M,M(f(x)),M(x)) \tag{1}$$

where M *is an* n*-by-*n *matrix and* $M(z)=Mz$*. Then, for every positive polynomial* p*, the function* g' *has* $n+\log_2 p(n)$ *bits that are* $(p(n),1/p(n))$*-unpredictable. Furthermore, the function* I *indicating the unpredictable bits depends only on the second operand of* g' (i.e., x)*.*

(Actually, it holds that $\mathrm{Pr}_{x\in\{0,1\}^n}[|I(x)|<n+\log_2 p(n)]<1/p(n)^2$.)

Proof Sketch: We first prove the theorem for the special case that f is regular; that is, for some $r\in[n]$, each image of f has approximately 2^r pre-images. In that case, the first $n-r-O(\log p(n))$ bits of $g'(M,x)$ (i.e., $M(f(x))_{[n-r-O(\log p(n))]}$) are $o(1/p(n))$-close to the uniform distribution over $\{0,1\}^{n-r-O(\log p(n))}$, because they are extracted from a source of min-entropy $n-r$. Likewise, even when fixing the first n bits of $g'(M,x)$, the next $r-O(\log p(n))$ bits (i.e., $M(x)_{[r-O(\log p(n))]}$) bits are $o(1/p(n))$-close to the uniform distribution over $\{0,1\}^{r-O(\log p(n))}$ (because they are extracted from a source of min-entropy r). Needless to say, this means that these $n-O(\log p(n))$ bits are unpredictable. The point is showing that the next $O(\log p(n))$ bits (i.e., those in locations $n+r-O(\log p(n))+1,...,n+r+O(\log p(n))$) are also unpredictable.

The latter fact is proved by showing that, for every $i\in[r-O(\log p(n))+1,r+O(\log p(n))]$, the bit $B'(M,x)=M(x)_i$ is a hard-core predicate of the function $F'(M,x)=(M,f(x),M(x)_{[i-1]})$. Specifically, efficiently predicting $M(x)_i$ with non-negligible advantage ϵ (when given $(M,f(x),M(x)_{[i-1]})$) implies (via [5]) an efficient procedure R such that $\mathrm{Pr}[R(M,f(x),M(x)_{[i-1]})=x]\geq\mathrm{poly}(1/\epsilon)$.[2]

$$\mathrm{Pr}_{M\in\{0,1\}^{n^2},x\in\{0,1\}^n,u\in\{0,1\}^{i-1}}\left[R(M,f(x),u)\in f^{-1}(f(x))\right]$$

[2] Here, it is essential that the $i^{\rm th}$ column of M is statistically independent of the rest of M.

$$\begin{aligned}
&\approx\ 2^r \cdot \Pr_{M\in\{0,1\}^{n^2},x\in\{0,1\}^n,u\in\{0,1\}^{i-1}}[R(M,f(x),u)=x] \\
&\geq\ 2^r \cdot 2^{-(i-1)} \cdot \Pr_{M\in\{0,1\}^{n^2},x\in\{0,1\}^n}[R(M,f(x),M(x)_{[i-1]})=x] \\
&\geq\ 2^{r+1-i} \cdot \mathrm{poly}(\epsilon),
\end{aligned}$$

where the first (approximate) equality holds because R is oblivious of the specific x that yields $f(x)$. Lastly, letting $R'(y) \stackrel{\rm def}{=} R(M,y,u)$ such that M and u are selected uniformly at random, implies that R' inverts f with probability at least $2^{r-i} \cdot \mathrm{poly}(\epsilon)$. Using $i \leq r+O(\log p(n))$, it follows that the hypothesis that this bit can be predicted contradicts the one-wayness of f.

Turning to the general case, the key observation is that the same argument can be applied here too by decoupling f (or rather its domain) into n parts such that f is approximately 2^r-to-1 on the $r^{\rm th}$ part. Discarding parts that have negligible density (in $\{0,1\}^n$), we observe that each part can be analyzed separately by considering the corresponding conditional probability space and using the fact that f must be hard to invert on each of these conditional spaces. Indeed, the unpredictable bits in the $r^{\rm th}$ part are $1,...,n-r-O(\log p(n))$ and $n+r-O(\log p(n))+1,...,n+r+O(\log p(n))$, which means that different bits are unpredictable in different parts. However, this is accommodated by Definition 1, which allows these locations to be determined arbitrarily as a function of the input (to g'). Furthermore, here these locations are determined as a function of the input to f (only). ■

Motivation towards the final construction. The set of unpredictable bits of g' is a random variable, denoted $I(x)$. This random variable is a function of the uniform distribution of $x \in \{0,1\}^n$. Actually, it is a sequence of $2n$ binary random variables, corresponding to the events $I(x) \ni i$ for all $i \in [2n]$. The only information available to us about this sequence (of $2n$ random variables) is that its sum exceeds $n+\log_2 p(n)$. This claim actually holds with probability at least $1-(1/p(n))^2$; however, the original work only uses the fact that the expectation of this sum exceeds $n+\log_2 p(n)$. In any case, we have no information about the distribution of the individual binary variables, whereas unpredictability holds with respect to individual locations relative to previous ones. The solution is twofold:

1. Apply a transformation such that the random variables that represent unpredictability of different bit locations in the resulting function are identically distributed. This is done by taking t' copies of the original distribution, selecting $r \in [2n]$ uniformly, and omitting the first r bits from the first copy and the last $2n-r$ bits of the last copy.

   Hence, each location in the resulting function represent a uniformly distributed bit location of the original function. The transformation comes with a cost: We have omitted $2n$ bits, but for $t' = \Omega(n/\log n)$ this lost is compensated by the larger total gain in the number of unpredictable bits from the t' copies.

2. Using $t = \omega(n^2/\log n)$ copies of the resulting distribution, we obtain a t-by-$(t'-1)\cdot 2n$ Boolean matrix in which, with probability $1-\exp(-t\cdot((\log n)/n)^2)$, each column has at least $t\cdot(0.5+\Omega((\log n)/n))$ unpredictable bits.

   Applying a randomness extractor to each column, we can extract

   $$((t'-1)\cdot 2n)\cdot(t\cdot(0.5+\Omega((\log n)/n))) \;=\; (t'-1)\cdot t\cdot(n+\Omega(\log n))$$

   bits, which exceeds the $t'\cdot t\cdot n$ bits used to generate this matrix.

The following construction implements the foregoing suggestion.

Construction 3 (the two-step construction): *For $t' = 2n/\log_2 n$, $\overline{x} = (x^{(1)}, ..., x^{(t')}) \in (\{0,1\}^n)^{t'}$ and $r \in [2n]$, let*

$$g''(r, M, \overline{x}) = (g'(M, x^{(1)})_{[r+1,2n]}, g'(M, x^{(2)}), ..., g'(M, x^{(t'-1)}), g'(M, x^{(t')})_{[r]}). \tag{2}$$

For $\delta = (\log_2 n)/n$, let $\mathrm{Ext} : \{0,1\}^t \times \{0,1\}^d \to \{0,1\}^{(0.5+\delta)\cdot t}$ be a strong extractor for min-entropy $(0.5 + 2\delta) \cdot t$ with deviation that is negligible (in n).[3] *For $\overline{z} = (z^{(1)}, ..., z^{(t)}) \in (\{0,1\}^{t'n})^t$, $r = (r_1, ..., r_t) \in [2n]^t$ and $i \in [n']$, where $n' = (t'-1)\cdot 2n$, let $\overline{g}_i(r, M, \overline{z}) = (g''(r_1, M, z^{(1)})_i, ..., g''(r_t, M, z^{(t)})_i)$; that is, $\overline{g}_i$ takes the i^{th} bit of each of the t values $g''(r_1, M, z^{(1)}), ..., g''(r_t, M, z^{(t)})$. For $s = (s_1, ..., s_{n'}) \in (\{0,1\}^d)^{n'}$, denoting $\mathrm{Ext}_s(z) = \mathrm{Ext}(z, s)$, we construct the function*

$$G(M, r, s, \overline{z}) \stackrel{\text{def}}{=} (M, r, s, \mathrm{Ext}_{s_1}(\overline{g}_1(r, M, \overline{z})), ..., \mathrm{Ext}_{s_{n'}}(\overline{g}_{n'}(r, M, \overline{z}))). \tag{3}$$

Note that the length of the input to G is $k(n) \stackrel{\text{def}}{=} n^2 + t \cdot \log_2(2n) + n' \cdot d + t \cdot t'n$, whereas the length of its output is $n^2 + t \log_2(2n) + n'd + n' \cdot (0.5 + \delta) \cdot t = k(n) + t \cdot ((1 + 2\delta) \cdot (t' - 1) - t') \cdot n$, which equals $k(n) + (2\delta \cdot t' - (1 + 2\delta)) \cdot tn > k(n) + \delta \cdot t'tn$. Assuming that $d = O(t)$, we get seed-length $O(t'tn)$ and a stretch of $\omega(t't)$ bits. Given these mild requirements of a randomness extractor, we can just use linear functions over a huge field or affine transformations effected by Toeplitz matrices.

Theorem 4 (the pseudorandomness of Construction 3): *Suppose that, for every positive polynomial p, the function $g' : \{0,1\}^{n^2+n} \to \{0,1\}^{2n}$ has $n + \log_2 p(n)$ bits that are $(p(n), 1/p(n))$-unpredictable. Furthermore, suppose that the locations of unpredictability are determined by the second operand of g; that is, $I(M, x)$ is independent of M, allowing us to use $I(x) \stackrel{\text{def}}{=} I(M, x)$. Then, G as defined in Construction 3 is a pseudorandom generator.*

Proof Sketch: We first note that, for every polynomial p, the function $g'' : [2r]\{0,1\}^{n^2+t'n} \to \{0,1\}^{(t'-1)\cdot 2n}$ has $(t' - 1) \cdot (n + \log_2 p(n))$ bits that are $(p(n), 1/p(n))$-unpredictable. Recall that $n' = (t - 1) \cdot 2n$ and that G has seed-length $O(t'tn)$ and a stretch of $\Omega(t't \cdot \log n)$ bits.

To analyze G, we use a hybrid argument, where (for $i \in \{0, 1, ..., n'\}$) the i^{th} hybrid distribution is

$$H^{(i)} \stackrel{\text{def}}{=} (M, r, s, \mathrm{Ext}_{s_1}(\overline{g}_1(r, M, \overline{z})), ..., \mathrm{Ext}_{s_i}(\overline{g}_i(r, M, \overline{z})), U_{(n'-i)\cdot t}). \tag{4}$$

where M, s, r and $\overline{z}$ are distributed uniformly (as in Eq. (3)), and U_m denotes the uniform distribution over m-bit strings.

Note that $H^{(n')}$ coincides with the output distribution of G whereas $H^{(0)}$ is uniformly distributed over the set of all strings of corresponding length. The indistinguishability of $H^{(i)}$ and $H^{(i-1)}$ follows by observing that $\overline{g}_i(r, M, \overline{z})$ has at least $(0.5 + 2\delta) \cdot t$ pseudorandom bits with respect to (equiv., when given) $(M, r, s, \overline{g}_1(r, M, \overline{z}), ..., \overline{g}_{i-1}(r, M, \overline{z})$. Specifically, we use the following facts.

1. The number of bits in the i^{th} column that are unpredictable with respect to their own row is the sum of t independent (and identically) distributed binary variables that have each expectation at least $0.5 + ((\log_2 p(n))/2n) > 0.5 + 3\delta$. Hence, with probability at least $1 - \exp(-\Omega(t \cdot \delta^2)) = 1 - \exp(-\omega(\log n))$, this number is at least $(0.5 + 2\delta) \cdot t$.

[3] Recall that a function $F : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a strong (k, ϵ)-extractor if, for every random variables of min-entropy at least k (i.e., $\max_x\{\Pr[X = x]\} \leq 2^{-k}$), it holds that the total variation distance between $(U_d, F(X, U_d))$ and U_{d+m} is upper-bounded by ϵ.

2. This sequence of individually unpredictable bits (wrt their own rows) is pseudorandom with respect to all prior columns. Here we use another hybrid argument (i.e., on the t rows of the t-by-n' matrix).

Hence, we can replace these unpredictable bits by random one, and obtain a column that has min-entropy at least $(0.5 + 2\delta) \cdot t$. Applying randomness extraction, the indistinguishability claim follows. ■

The seed length. Construction 3 does improve (slightly) the complexity bounds on the number of invocations of the one-way function (from $\omega(n^3)$ to $\omega(n^3/\log^2 n)$), but the seed-length it uses is $\omega(n^4/\log^2 n)$. To improve the seed-length, one may apply the "re-cycling" technique of [16], and this is indeed what is done in [14, Sec. 8]. This allows to cut the seed-length by a factor of $\Omega(t')$, yielding seed-length $\omega(n^4/\log n)$.

3 Adaptation to the uniform complexity model

Working in the model of non-uniform complexity allows for hiding the non-uniformity of the security reductions (i.e., reducing the inversion of f to the violation of the pseudorandomness of G). In the current context, following [13], non-uniform steps are typically done in the definition of hybrid distributions. The current source of the problem is the fact that the function $I : \{0,1\}^n \to 2^{[2n]}$ that indicates the unpredictable bits is unlikely to be efficiently computable. Fortunately, as noted in [14, Apdx A], this problem can be addressed by using suitable results regarding "hard-core regions" (see [14, Lem A.3], which follows [7], which in turn follows [12]). Details follow.

The first step is to prove a strengthening of Theorem 2. In this stronger version, the unpredictability holds also with respect to algorithms that make input-oblivious queries regarding whether a specific bit is unpredictable with respect to a specific (other) input. This stronger version will later allow us to conduct the hybrid argument that takes place in the proof of Theorem 4. Specifically, we will provide the (single-instance) distinguisher that is used in the hybrid argument with the foregoing oracle, which will enable this distinguisher to emulate the rest of the hybrid (by sampling instances with corresponding unpredictable/predictable bits).

Actually, we outline two different (strong) versions of Theorem 2. The first version refers to the function g' as defined in Eq. (1), and is implicit in the first part of the proof of [14, Lem. A.5].[4] Specifically, this version establishes the average unpredictability of bits in $g'(M, x) = (M(f(x)), M(x))$ (also when given input-oblivious access to an oracle that on input (M', x', i') indicates whether or not $(M'(f(x')), M'(x'))_{i'}$ is unpredictable). The problem with this version is that it leads to a seed length of $\Omega(t \cdot n^2)$, because (in the uniform setting) t different n-by-n matrices are used for the t copies of g''. A second version proves an analogous result when referring to a related function in which the (random) n-by-n matrix M is replaced by a hash function of description length $k' = \omega(n \log n)$, which will yield seed length $O(t \cdot k')$. Before detailing the construction of this collection of hash functions, we outline the rest of the argument, which proceeds very much as in [14, Apdx A.2].

The issue is that, here (unlike in the non-uniform version), the predicate that indicates whether a bit is unpredictable depends not only on x but also on the matrix M (or the hash function that replaces it). Thus, we cannot uses the same M in all t copies of g', but rather have to use t

[4] The second part of the proof of [14, Lem. A.5] is analogous to [14, Sec. 6], which refers to g''.

independently distributed M's (resp., hashing functions). Once we do so, the argument proceeds analogously to the non-uniform case, when having gained the ability to construct the various hybrids.

The alternative hash functions. In order to establish a (strong) version of Theorem 2, we need the collection $\mathcal{H}$ of hash function that replaces all n-by-n matrices to satisfy two properties:

1. The collection should yield a strong randomness extractor; in particular, any collection in which a uniformly selected function maps points in a pairwise independent manner will do. That is, for every $x \neq x' \in \{0,1\}^n$ and $u, v \in \{0,1\}^n$, it should hold that

$$\mathrm{Pr}_{h\in\mathcal{H}}[h(x)=u\,\&\,h(x')=v] = 2^{-2n}.$$

   Actually, a deviation by a factor of $1 \pm (1/\mathrm{poly}(n))$ can be tolerated.

2. For every $i \in [n]$ and every $x \in \{0,1\}^n$, the sub-collection of $h \in \mathcal{H}$ obtained by conditioning on the value of $h(x)_{[i-1]}$ yields a hardcore predicate (akin to [5]). That is, there exists a probabilistic polynomial-time oracle machine R such that, for every $i \in [n]$, $(x, w) \in \{0,1\}^n \times \{0,1\}^{i-1}$, and $P : \mathcal{H} \times \{0,1\}^{n+i-1} \to \{0,1\}$ such that

$$\mathrm{Pr}_{h\in\mathcal{H}}[P(h, f(x), w)=h(x)_i \mid h(x)_{[i-1]}=w] \geq \frac{1}{2} + \frac{1}{\mathrm{poly}(n))}$$

   it holds that

$$\mathrm{Pr}_{h\in\mathcal{H}}[R^P(h, f(x), w)=x \mid h(x)_{[i-1]}=w] \geq 1/\mathrm{poly}(n).$$

(We comment that a related construction appears in [7, Sec. 4.1], but it requires some adaptation, and it seems easier and nicer to provide a (partially) self-contained construction here.)

Focusing on the second property, we take the Cartesian product of n copies of a collection that corresponds to a locally list-decodable code. Specifically, for $\epsilon = \exp(-\omega(\log n))$ and $m = \mathrm{poly}(n/\epsilon)$, we consider the code $C : \{0,1\}^n \to \{0,1\}^m$, presented in [15, Sec. 4.2], and let $\mathcal{H} \stackrel{\rm def}{=} \{h_{i_1,...,i_n} : i_1, .., i_n \in [m]\}$ such that $h_{i_1,...,i_n}(x) = (C(x)_{i_1}, .., C(x)_{i_n})$. We stress that, as shown [15, Sec. 4.3–4.4], for every $\delta \geq \epsilon$, list-decoding all C-codewords that are $(0.5-\delta)$-close to a given n-bit string is performed in time $\mathrm{poly}(n/\delta)$. Hence, $\mathcal{H}$ satisfies the second property. We also note that the code C is linear and that all (nonzero) codewords are of Hamming weight $(0.5 \pm \epsilon) \cdot m$, which implies that the first property is also satisfied. Lastly, note that the length of the description of functions in $\mathcal{H}$ is $n \cdot \log_2 m = O(n \log(n/\epsilon))$.

Acknowledgments

I am grateful to Madhu Sudan for a helpful discussion.

References

[1] Manuel Blum and Silvio Micali. How to Generate Cryptographically Strong Sequences of Pseudo Random Bits. *SIAm J. on Comput.*, Vol. 13, pages 850–864, 1984. Preliminary version in *23rd FOCS*, 1982.

[2] Oded Goldreich. *Foundations of Cryptography – Volume 1: Basic Tools*. Cambridge University Press, 2001.

[3] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[4] Oded Goldreich, Hugo Krawczyk, and Michael Luby, On the Existence of Pseudorandom Generators. *SIAM J. on Computing*, Vol. 22-6, pages 1163–1175, 1993.

[5] Oded Goldreich and Leonid A. Levin. A Hard-Core Predicate for all One-Way Functions. In *21st STOC*, 1989.

[6] Iftach Haitner, Danny Harnik, and Omer Reingold. On the Power of the Randomized Iterate. In *CRYPTO06*, Springer LNCS (4117), pages 22–40, 2006.

[7] Iftach Haitner, Omer Reingold, and Salil Vadhan. Efficiency Improvements in Constructing Pseudorandom Generators from One-Way Functions. *SIAM Journal on Computing*, Vol. 42 (3), pages 1405–1430, 2013.

[8] Iftach Haitner and Salil Vadhan. The many entropies in one-way functions. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 159–217, Springer, 2017.

[9] Johan Hastad. Pseudo-Random Generators under Uniform Assumptions. In *22nd STOC*, 1990.

[10] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, Vol. (28) 4, pages 1363-1396, 1999. Combines the results of [13] and [9].

[11] Thomas Holenstein. Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In *3rd TCC*, 2006.

[12] Thomas Holenstein. Strengthening key agreement using hard-core sets. PhD thesis, ETH Zurich, 2006

[13] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random Generation from one-way functions. In *21st STOC*, 1989.

[14] Noam Mazor and Rafael Pass. Counting Unpredictable Bits: A Simple PRG from One-way Functions. *ECCC*, TR23-143, 2023.

[15] Madhu Sudan, Luca Trevisan, Salil P. Vadhan. Pseudorandom Generators without the XOR Lemma. *JCSS*, Vol. 62 (2), pages 236–266, 2001.

[16] Salil Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *44th STOC*, 2012.

[17] Andrew C. Yao. Theory and Applications of Trapdoor Functions. In *23rd FOCS*, 1982.