True Random Number Generators Secure in a Changing Environment: Improved Security Bounds

Maciej Skorski*
maciej.skorski@gmail.com

Cryptology and Data Security Group, University of Warsaw

Abstract. Barak, Shaltiel Tromer showed how to construct a True Random Number Generator (TRNG) which is secure against an adversary who has some limited control over the environment.
In this paper we improve the security analysis of this TRNG. Essentially, we significantly reduce the entropy loss and running time needed to obtain a required level of security and robustness.
Our approach is based on replacing the combination of union bounds and tail inequalities for ℓ-wise independent random variables in the original proof, by a more refined of the deviation of the probability that a randomly chosen item is hashed into a particular location.

1 Introduction

1.1 Random number generators

Random number generators are used for various purposes, such as simulating, modeling, cryptography and gambling. Below we briefly discuss possible approaches and issues in generating random numbers.

True Random Number Generators. The term "True Random Number Generator" (TRNG) refers to a hardware device that generates random numbers based on some *physical* phenomena (radiation, jitter, ring oscillators, thermal noise etc.). As an example of such an implementation one can mention the HotBits project [Wal11], which is based on timing the decay of Caesuim-137. Such implementations are typically very reliable and hard to tamper. Sometimes as a TRNG one considers also a software application which generates random numbers based on unpredictable human behavior, like mouse movement or typing keyboard keys. Even if they are not completely unpredictable (because knowing an operator's habits helps in predicting the output) the generated results are typically of high quality and have found real-world cryptographic applications (e.g. PGP). True Random Number Generators do not have internal states (hence are not predictable from the sampling history) and produces a high quality output. However, they are usually slow and not easy to implement (for instance,

* This work was partly supported by the WELCOME/2010-4/2 grant founded within the framework of the EU Innovative Economy Operational Programme.

because of the need of a dedicated hardware).

Pseudo-random Number Generators. On the other side we have pseudo-random number generators (PRNG's), algorithms that use mathematical formulas to produce a randomly looking output. They depend on internal states and for this reason needs to be externally "seeded". They are fast and relatively easy to implement. However, seeding them properly is of critical importance. A lot of PRGNs from standard software use predictable seeds. For example, recall the discovered vulnerability in a version of Netscape browser [GW96].

Classical TRNG Design. Typically the process of building a TRNG consists of the following stages

(a) Setting an entropy source (physical or non-physical)
(b) Post-processing part
(c) Quality evaluation

The entropy source does not necessarily provide samples of excellent quality and therefore step (b) is needed. Its purpose is to eliminate bias or dependencies. Posprocessing procedure could be very simple as the famous von Nuemann corrector or very complicated. Finally, the whole implementation should be subjected to common statistical tests, for example [Mar96,BRS+10].

1.2 TRNGs in environments under (partial) adversarial control

Imagine a setting where an attacker has some *partial* control over the environment where the sampling device operates. For instance he could influence voltage or temperature. The goal is to build a TRNG which is robust in such a setting.

Resilient Extractors. Slightly simplifying the problem, we can focus on the postprocessing algorithm, that is on how to extract random bits from randomness within a source. Suppose that we have a source that produces samples distributed according to X, where X is unpredictable in the sense that it has high entropy. This assumption is the most general way of capturing "randomness" because we cannot assume that our source, which might be a very complicated physical processes, has any specific nice properties. One can extract almost uniform bits from high entropy source by the use of so called "randomness extractors". However, no deterministic procedure can extract one bit which is close to uniform from *every* high-entropy source [SV86]. The general purpose randomness extractors, which are guaranteed to work with every source having enough entropy, require additional truly random bits (called the *seed*) as a "catalyst" [NZ96]. While this concept is generally extremely useful, the direct application of a seeded extractor to an entropy source does not provide a way to build a good TRNG:

(a) In real applications, generating even small number of *truly* random bits can be extremely hard.
(b) In some designs there might be correlations between the source and the seed. For instance, when the seed is stored in a TRNG which uses a source of randomness within the computer (like timing events).

(c) If we want a TRNG with some kind of resilience, we should ensure it also for the procedure generating seed.

One can overcome this problem by designing a deterministic extractor which works for a restricted class of sources. Some constructions for specific cases are known; see for instance [KZ03,GR05] and recall the most famous example - von Neumann sources. However they are not applicable in our case.

Resilient extractors via fixing seeds. Barak, Shaltiel and Tromer [BST03] came up with the very simple but surprisingly useful idea of *fixing a seed*. Let us discuss it briefly. Suppose for a moment that we have only one source X of entropy k and an arbitrary seeded extractor , that from any X having k-bits of min-entropy extracts m close-to-uniform bits using a random seed. This means that the output is close to uniform *in average* over all possible seeds. Hence running the extractor with a *fixed* seed, for *most of the seeds*, yields an output which is still close to uniform (by the Markov Inequality). Now let us make a realistic assumption that the source X depends on some finite number of boolean[1] environmental variables (corresponding to changes in the voltage, temperature, radiation etc) and suppose that

(a) the adversary controls t of the environmental variables
(b) in every of 2^t possible configurations of the "compromised" states, entropy in the source is big enough (i.e. at least k)

Provided that t is small, by the union bound we conclude that fixing the seed, for most of the seeds, we still obtain a good extractor in every state. Below we summarize this discussion more quantitatively:

Proposition 1 (Resilient extractor from any extractor). *Let* $\{X_e\}_{e\in\{0,1\}^t}$ *be a collection of* n*-bit random variables and let* $\mathrm{Ext} : \{0,1\}^n \times \mathcal{S} \to \{0,1\}^m$ *be a function such that for every* e *the distribution of* $\mathrm{Ext}_S(X_e, S)$*, where* S *if chosen randomly from* $\mathcal{S}$*, is* ϵ*-close to uniform. Then for all but a* 2^{-u} *fraction of* $s \in \mathcal{S}$ *the distribution* $\mathrm{Ext}(X_e, s)$ *is* $2^{u+t}\epsilon$*-close to uniform for every* $e \in \{0,1\}^t$*.*

Even the best extractors need in worst case at least $k = m + 2\log(1/\epsilon) - \mathcal{O}(1)$ bits of entropy on their input in order to extract m bits which are ϵ-close to uniform [RTS00]. The optimal rate, with $k = m + 2\log(1/\epsilon) - 2$, is achieved for example by 2-universal hashing (the Leftover Hash Lemma).

Resilient TRNG from the resilient Extractor. The assumption that our extractor works only for *small* (of size 2^t) family of distributions in the context of a TRNG is not restrictive. Indeed, imagine a manufacturer who has a device producing samples of a distribution X. The seed s is chosen once and for all and every single TRNG box is built by composing a copy of the sampling device with the extractor seeded by the same s. Once s is chosen, could be even made public. The confidence level δ ensures that with high probability we can find a good s. After choosing s, the manufacturer tests the implementation against

[1] Without losing generality, since we can describe more "granulated" properties using more than one boolean variable

randomness test like `NIST` [BRS+10] and `DIEHARD` [Mar96]. For more details, we refer the reader to [BST03]. The above discussion can be summarized by the following result

Theorem 1 (Simple resilient TRNG, informal). *There exists an efficient seeded extractor* Ext *such that for every source X which in every of 2^t states of the environment has the min-entropy at least*

$$k \geqslant m + 2\log(1/\epsilon) + 2\log(1/\delta) + 2t - 2, \tag{1}$$

for all but at most a δ fraction of the seed s it holds that $\mathrm{Ext}(X, s)$ *is ϵ-close to the uniform m-bit string in every state of the environment.*

Note that the entropy loss $L = k - m$ must be substantially bigger than $2\log(1/\epsilon)$ if we want non-trivial values of δ and t. Than additional entropy loss is a price we pay for resilience of the extractor.

The resilient TRNG of Barak Shaltiel and Tromer Barak et al. showed and implemented a construction of a TRNG which is secure against any adversary who controls t environmental variables. In their proof ℓ-wise independent hash families are used as extractors. Roughly speaking, the assumption on ℓ-wise independence allows estimating higher moments of the statistical distance between output of hashing and the uniform distribution. This way we get significant improvements over the Markov Inequality used in Theorem 1.

Theorem 2 ([BST03] Resilient TRNG from any ℓ-universal hash family, informal). *Let $\mathcal{H}$ be an ℓ-wise independent family of hash functions from n to m bits. Suppose that an n-bit source X in every of 2^t states of the environment has the min-entropy at least*

$$k \geqslant \frac{\ell + 2}{\ell} \cdot m + 2\log(1/\epsilon) + \frac{2\log(1/\delta)}{\ell} + \frac{2t}{\ell} + \log \ell - 2 + \frac{4}{\ell}. \tag{2}$$

Then for all but δ fraction of $h \in \mathcal{H}$ it holds that $h(X)$ is ϵ-close to the uniform m-bit string in every state of the environment. For $\ell = 2$ we have the better bound $k \geqslant m + 2\log(1/\epsilon) + 2\log(1/\delta) + 2t - 2$.

We remark that the constant -2 in Theorem 2 is slightly better than what is stated in [BST03]. This is because the authors used a slightly weaker statement of the Leftover Hash Lemma.

Optimizing the parameters. The construction of Barak et al. depends on several parameters and gives a lot of freedom to optimize the design for a particular real-world application. For instance, minimizing the entropy loss (i.e. minimizing k) is of the crucial importance when the source produces entropy slowly or expensively (for instance when one uses patterns typing by a user, like mouse clicking, as the source). In such a case, one may prefer the (slightly more complicated) implementation with universal families of a higher degree. In the other hand, when the sampling procedure is more efficient (like thermal noise) one can afford entropy losses and prefer faster running time of the extractor, a

higher confidence level for the choice of the seed or to increase the number of the environmental variables under adversarial control.

Advantages and disadvantages. The big advantage of Theorem 2 over trivial Theorem 1 is that one can increase t proportionally to the degree ℓ of hashing family, which is actually a bit surprising. The main disadvantage is the entropy loss $L = k - m$ needs to be bigger than $\frac{2m}{\ell}$ which is $\Omega(m)$ for small ℓ. Theoretically, one can reduce this with ℓ big enough, however this could be inconvenient because of the following two reasons: (a) the running time increases by a factor $\mathrm{poly}(\ell)$ and (b) the description of an ℓ-wise independent hashing family on $\{0,1\}^n$ takes ℓn bits hence there could be a problem with sampling a good function h (note that n could be even much bigger than k, which is the case of low entropy rate).

1.3 Our results and techniques

Summary of our contribution. We reduce the entropy loss in Theorem 2 by $2m/\ell$ for any ℓ, saving linear amount of entropy. This matches the RT-bound and hence is tight. Our approach is based on the more refined analysis of the concentration properties of universal hashing.

Hashing into a given slot - bounds on the deviation. Applying estimates for ℓ-wise independent random variables [BR94] we prove the following formula

Lemma 1. *Let $\ell \geqslant$ be an even integer, $\mathcal{H}$ be an ℓ-universal family of hash functions from n to m bits and $k - m \gg 2\log \ell$. Then for any X of min-entropy at least k we have*

$$\mathop{\mathbf{E}}_{h \leftarrow \mathcal{H}} \left|\Pr\left(h(X) = y\right) - \Pr(U_m = y)\right|^{\ell} \lesssim C_{\ell} \cdot \left(2^{-k-m}\ell\right)^{\ell/2} \tag{3}$$

where $C_{\ell} = 2\sqrt{\pi\ell} \cdot \mathrm{e}^{1/6\ell} \cdot \left(\frac{5}{2\mathrm{e}}\right)^{\ell/2}$.

The left-hand side of Equation (3) gives the deviation of the probability (over the choice of the hash functions) of hashing a random variable X into a particular slot from its expectation (equal to 2^{-m} by the universal property). Studying such deviations is a natural idea, used essentially in [BSW03].

Remark 1 (Sharp bounds on the deviation). One can get *symptomatically* sharp bounds in Equation (3) with more effort, by expanding the bracket and compute the ℓ-th moment precisely. The improvement is by a factor of c^{ℓ} and leads to further non-trivial improvements of the results of Barak et al. We find this gain too small and do not optimize the bounds in Equation (3).

Improved bounds on the fraction of good seeds in hashing. We prove the following inequality

Proposition 2. *Let X be an n-bit random variable and $\mathcal{H}$ be an arbitrary family of functions from n bits to m bits. Let $\ell \geqslant 2$ be an even integer and $\epsilon > 0$. Then*

$$\Pr_{h \leftarrow \mathcal{H}}\left[\mathrm{SD}(h(X); U_m) > \epsilon\right] \leqslant \frac{\mathbf{E}_{y \leftarrow U_m} \mathbf{E}_{h \leftarrow \mathcal{H}} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^{\ell}}{2^{-m\ell}(2\epsilon)^{\ell}}. \tag{4}$$

This estimate allows us to bound the fraction of the seeds (i.e. hash functions) for which the statistical distance is small, in terms of the deviation of the hashing probability. This bound offers a significant improvement over an alternative approach which bounds the deviation $|\Pr[h(X) = y] - \Pr[U_m = y]|$ for every y *separately* and after that uses the union bound to upper-bound the sum (this is essentially the strategy of Barak et al.). Intuitively, the gain could be even of a factor 2^m which should save a linear (in m) amount of entropy. Indeed, unlike Theorem 2 we are able to get meaningful security even for $k < m(1+2/\ell)$ (assuming small t and ℓ).

Improved efficiency and security of the construction of Barak et al. Using the tools discussed above, we prove the following result

Theorem 3 (A resilient TRNG from any ℓ-universal hash family, informal). *Let $\mathcal{H}$ be an ℓ-universal family of hash functions from n to m bits, where ℓ is an even integer. Suppose that a source X which in every of 2^t states of the environment has the min-entropy at least*

$$k \geqslant m + 2\log(1/\epsilon) + \frac{2\log(1/\delta)}{\ell} + \frac{2t}{\ell} + \log \ell - 2. \tag{5}$$

Then for all but at most a δ fraction of $h \in \mathcal{H}$ it holds that $h(X)$ is ϵ-close to the uniform m-bit string in every state of the environment.

The theorem is valid under the assumption $k - m \gg \log \ell$ which we omitted as it is satisfied for interesting values of parameters. Our improvements over [BSW03] can be summarized as follows:

(i) For $\ell = 2$ (the simplest extractor) we save $\log(1/\delta) + t$ bits of entropy. Alternatively, the probability of choosing a bad hash functions in the preprocessing phase gets squared and the number of the states under adversarial control can be doubled. Entropy savings is important for expensive or slow sources. Higher confidence level for the choice of the seed is important if we want to subject the implementation to the statistical tests, like `DIEHARD` [Mar96]. Finally, the more variables under adversarial control the more robust the PRNG is.

(ii) For $\ell > 2$ (but not too big), in comparison to Theorem 2, we save the *linear amount* of entropy, precisely $\frac{2m}{\ell}$. The case $\ell > 2$ is preferable for slow or expensive entropy sources or when the priority is the high robustness (i.e. big number of states).

(iii) Even for $\ell \gg 2$ our result is still much better in some settings. For example, for $\epsilon = 2^{-\sqrt[10]{m}}$ (reasonable subexponential security) and $\ell \approx \log(1/\epsilon)$ the entropy loss $L = k - m$ becomes close to $L \approx 2\log(\epsilon)$ whereas Theorem 2 gives $L \approx 2\log(\epsilon) + 2m^{0.9}$. In general, the entropy amount $2m/\ell$ we save can be used to increase the number of the adversary's degrees of freedom by m, which is quite a lot.

Remark 2. Even reducing the entropy loss by only *constant* number of bits gives *non-trivial* results! This is because decreasing the minimal k by d is equivalent to increasing t by $d\ell/2$ (keeping ϵ, δ unchanged). In particular, optimizing the bound in Theorem 3 would slightly improve our results (see Remark 1).

2 Preliminaries

STATISTICAL DISTANCE. For any two random variables X, Y taking values in the same space we define the statistical distance of X and Y to be $\Delta(X;Y) = \sum_x |\Pr[X = x] - \Pr[Y = x]|$. When $\Delta(X;Y) \leqslant \epsilon$ we say that X and Y are ϵ-close.

ENTROPY NOTIONS. The min-entropy of a random variable X is defined to be $\mathbf{H}(X) = \log(1/\max_x \Pr[X = x])$.

INDEPENDENT HASH FUNCTIONS. A family $\mathcal{H}$ from n to m bits is called ℓ-wise independent if and only if for every different n-bit strings $x_1, x_2, \ldots, x_\ell$ and h chosen at random from $\mathcal{H}$ the random variables $h(x_1), h(x_2), \ldots, h(x_\ell)$ are independent and uniform.

2.1 Security definitions

CHANGING ENVIRONMENT - SECURITY GAME. We consider the following ideal setting[BST03]

(i) An adversary chooses 2^t distributions $X_1, \ldots, X_{2^t}$ over $\{0,1\}^n$, such that $\mathbf{H}_\infty(X) \geqslant k$ for all $i = 1, \ldots, 2^t$.
(ii) A public parameter h is chosen at random and independently of the choices of X_i
(iii) The adversary receives h, and selects $i \in \{1, \ldots, 2^t\}$
(iv) The user computes $\mathrm{Ext}(X)$, where X is sampled from X_i.

Note that in the game defining the security of an extractor, the adversary chooses the distribution and the user chooses (independently) a seed. Here the adversary is in some sense "semi-adaptive", because he can choose an arbitrary distribution but from the class of distributions he had *committed* to before he saw a seed. Of course, the adversary cannot be made fully-adaptive in the sense that he chooses a distribution without any restriction after seeing the seed. Thus, this definition seems to be a reasonable compromise.

RESILIENT EXTRACTOR. We define resilient extractor exactly as in [BST03] except that we state the confidence level δ explicitly.

Definition 1 (Resilient extractor [BST03]). *Given $n, k, m, \epsilon, \delta$ and t an extractor* E *is t-resilient with the confidence level δ if, in the above setting, with probability $1 - \delta$ over the choice of the public parameter s the statistical distance between $\mathrm{Ext}(X, s)$ and U_m is at most ϵ. For shortness, we also call it (k, ϵ, t, δ)-resilient extractor.*

This together with the entropy source yields a construction of a TRNG which is robust against some adversarial influences. One possible concern here is how to ensure that the entropy amount, under possible influences, is still sufficient? This is a serious problem but must be solved *independently* on the designing an extractor, because if the adversary had a way to significantly decrease entropy

amount then no scheme would be secure anymore, regardless of what an extraction function is applied. We note that, as mentioned in [BST03], the security definition might be even too strong for real world applications. For example, the assumption that the adversary is computationally unlimited and that all distributions X_i could be completely independent[2]. For long data streams, the extractor can be applied sequentially to consecutive blocks, provided that each block has enough entropy conditioned on all previous blocks.

3 Improved analysis for pairwise independent hashing

Motivating discussion. Let $\mathcal{H}$ be a family of 2-universal hash functions from n to m bits and let X be a distribution over $\{0,1\}^n$ of min-entropy at least k. We will show that if $L = k - m$ is big enough, then the distribution $H(X), H$, where H is a random member of $\mathcal{H}$, is ϵ-close to U_m, H. This result is known as the Leftover Hash Lemma:

Theorem 4. *For $\mathcal{H}$, H and X as above we have* $\mathrm{SD}((H(X), H); (U_m, H)) \leqslant \sqrt{2^{m-k}}$.

Note that $L = k - m$ needs to be roughly $2\log(1/\epsilon)$ if we want to guarantee that the statistical distance at most ϵ. We will refer to L as the entropy loss, because it equals the difference between the amount of entropy we invest and the length of the extracted output. By the Markov Inequality we trivially obtain the following corollary (see also [Gol08], the remark after Theorem D.5)

Corollary 1. *For all but most a δ fraction of the functions $h \in \mathcal{H}$ it holds that* $\mathrm{SD}(h(X); U_m) \leqslant \sqrt{2^{m-k}}/\delta$.

This corollary states that for a *fixed* source X, a *fixed* hash function yields a good extractor for all but a small fraction of hash functions. In particular we obtain the existence of an resilient extractor with parameters as in Theorem 1.

Improved analysis by the second moment technique. In Lemma 2 below we will prove a much better result than Corollary 1. We will apply the Markov Inequality for the second moment. Essentially, we bound the deviation of the probability of hashing X into particular value from its expectation which is 2^{-m} (from the universal property).

Lemma 2. *Let $\mathcal{H}$ be a 2-universal family of hahs functions from n to m bits and let X be a distribution of min-entropy at least k. Then for all but an δ fraction of $h \in \mathcal{H}$ we have* $\mathrm{SD}(h(X); U_m) < \sqrt{2^{m-k}/\delta}$.

As an easy corollary we obtain the following theorem, which is much better than Theorem 2

Theorem 5 (A resilient TRNG from 2-universal family). *Let $\mathcal{H}$ be a 2-universal family of hahs functions from n to m bits and let δ, ϵ be parameters.*

[2] They should be related being a perturbed version of the same distribution.

Then for all but a δ fraction of $h \in \mathcal{H}$, the function h is a (k, ϵ, t, δ)-resilient extractor where

$$k \geqslant m + 2\log(1/\epsilon) + \log(1/\delta) + t \tag{6}$$

Proof. The proof will follow from the following claims:

Claim 1. For every X we have

$$\Pr_{h \leftarrow \mathcal{H}}[\mathrm{SD}(h(X); U_m) > \epsilon] \leqslant \frac{\mathbf{E}_{y \leftarrow U_m} \mathbf{E}_{h \leftarrow \mathcal{H}} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2}{2^{-2m}\epsilon^2} \tag{7}$$

Claim 2. The expression

$$\mathop{\mathbf{E}}_{h \leftarrow \mathcal{H}} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2$$

over the distributions X of min-entropy at least k is maximized for a flat X, i.e. X uniform over a set of size 2^k.

Claim 3. For every X uniform over a set of size 2^k we have

$$\mathop{\mathbf{E}}_{h \leftarrow \mathcal{H}} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2 \approx 2^{-m-k}. \tag{8}$$

Now we give the proofs.

Proof (Proof of Claim 1). By the definition of the statistical distance and the Markov Inequality we obtain

$$\begin{aligned}\Pr_{h \leftarrow \mathcal{H}}[\mathrm{SD}(h(X); U_m) > \epsilon] &= \Pr_{h \leftarrow \mathcal{H}}\left[\mathop{\mathbf{E}}_{y \leftarrow U_m} |\Pr[h(X) = y] - \Pr[U_m = y]| > 2^{-m}\epsilon\right] \\ &\leqslant \frac{\mathbf{E}_{h \leftarrow \mathcal{H}} \left(\mathbf{E}_{y \leftarrow U_m} |\Pr[h(X) = y] - \Pr[U_m = y]|\right)^2}{2^{-2m}\epsilon^2}\end{aligned} \tag{9}$$

The inequality between the first and the second moment (which follows immediately from the Jensen Inequality) yields

$$\left(\mathop{\mathbf{E}}_{y \leftarrow U_m} |\Pr[h(X) = y] - \Pr[U_m = y]|\right)^2 \leqslant \mathop{\mathbf{E}}_{y \leftarrow U_m} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2. \tag{10}$$

Combining Equation (11) and Equation (10) and changing the order of the expectations we obtain

$$\begin{aligned}\Pr_{h \leftarrow \mathcal{H}}[\mathrm{SD}(h(X); U_m) > \epsilon] &\leqslant \frac{\mathbf{E}_{h \leftarrow \mathcal{H}} \mathbf{E}_{y \leftarrow U_m} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2}{2^{-2m}\epsilon^2} \\ &\leqslant \frac{\mathbf{E}_{y \leftarrow U_m} \mathbf{E}_{h \leftarrow \mathcal{H}} \left(\Pr[h(X) = y] - \Pr[U_m = y]\right)^2}{2^{-2m}\epsilon^2}\end{aligned} \tag{11}$$

which finishes the proof. □

Proof (Proof of Claim 2). This fact easily follows from the extreme point technique. It is known that every distribution of min-entropy k is a convex combination of flat distributions of min-entropy k. Our expression is a *convex* function of the distribution X. Hence, the maximum is on a flat distribution. □

Proof (Proof of Claim 3). By expanding the square we get

$$\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\left(\Pr[h(X)=y]-\Pr[U_m=y]\right)^2=\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=y]^2 - 2\cdot 2^{-m}\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=y]+2^{-2m} \tag{12}$$

Let X' be an independent copy of X. By the universality of $\mathcal{H}$, we can compute the first term as follows

$$\begin{aligned}\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=y]^2 &= \mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=h(X')=y]\\ &= \mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=h(X')=y|X\neq X']\Pr[X\neq X']\\ &\quad+\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=h(X')=y|X=X']\Pr[X=X']\\ &= 2^{-2m}\Pr[X\neq X']+2^{-m}\Pr[X=X']\\ &\approx 2^{-2m}+2^{-m-k}\end{aligned} \tag{13}$$

where the last approximation follows from $\Pr[X=X']=2^{-k}\ll 1$. By the universality we also have

$$\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}\Pr[h(X)=y]=2^{-m}. \tag{14}$$

The claim follows by plugging Equation (13) and Equation (14) into Equation (12). □

The proof is finished. □

4 Improved analysis for ℓ-wise independent hashing

It is easy to see that Proposition 2 and Lemma 1, together with the observation that the right hand side Proposition 2 among all X of min-entropy is maximized for flat X (which follows by convexity, see Claim 2), imply the following

Theorem 6 (An resilient from ℓ-universal hashing). *Let $\mathcal{H}$ be an ℓ-universal family of hash functions from n to m bits and let ϵ,δ be parameters. Then for all but a δ fraction of $h\in\mathcal{H}$ the function h is a (k,ϵ,t,δ)-resilient extractor where*

$$k\geqslant m+2\log(1/\epsilon)+\frac{2\log(1/\delta)}{\ell}+\frac{2t}{\ell}+\log\ell-2 \tag{15}$$

The proofs of Proposition 2 and Lemma 1 are discussed in the next two subsections. For consistency with some standard notations we denote $\ell=p$.

4.1 Bounds on the fraction of good seeds.

We give the proof of Proposition 2

Proof (Proof of Proposition 2). Let $\delta(y,h) = \Pr[h(X)=y] - \Pr[U_m = y]$. Note that we have $\mathrm{SD}(h(X);U_m) = \frac{1}{2}\cdot 2^m \mathbf{E}_{y\leftarrow U_m} |\delta(y,h)|$. By the Markov Inequality we obtain

$$\begin{aligned}\Pr_{h\leftarrow\mathcal{H}}[\mathrm{SD}(h(X);U_m) > \epsilon] &= \Pr_{h\leftarrow\mathcal{H}}\left[\mathop{\mathbf{E}}_{y\leftarrow U_m}|\delta(y,h)| > 2\cdot 2^{-m}\epsilon\right] \\ &\leqslant \frac{\mathbf{E}_{h\leftarrow\mathcal{H}}\left(\mathbf{E}_{y\leftarrow U_m}|\delta(y,h)|\right)^\ell}{2^{-mp}(2\epsilon)^\ell}. \end{aligned} \tag{16}$$

Since for every h we have $\left(\mathbf{E}_{y\leftarrow U_m}|\delta(y,h)|\right)^\ell \leqslant \mathbf{E}_{y\leftarrow U_m}|\delta(y,h)|^\ell$ by the Jensen Inequality, the last equation implies

$$\Pr_{h\leftarrow\mathcal{H}}[\mathrm{SD}(h(X);U_m) > \epsilon] \leqslant \frac{\mathbf{E}_{h\leftarrow\mathcal{H}}\left(\mathbf{E}_{y\leftarrow U_m}|\delta(y,h)|^p\right)}{2^{-m\ell}(2\epsilon)^\ell}. \tag{17}$$

The result follows by exchanging the order of the expectations. □

4.2 L_p-distance between the output of hashing and the uniform distribution

Proof (Proof of Lemma 1). We can assume that X is flat. We will use the well-known estimate on ℓ-wise independent random variables.

Lemma 3 (ℓ-wise independence moment estimate [BR94]). *Let $\ell \geqslant 4$ be an even integer. Let $Z_1,\ldots,Z_t$ be ℓ-wise independent random variables taking values in $[0,1]$. Let $Z = Z_1+\ldots+Z_n$, $\mu = \mathbf{E}\,Z$. Then we have*

$$\mathbf{E}\,|Z-\mu|^\ell \leqslant C_\ell\cdot\left(\ell\mu+\ell^2\right)^{\ell/2} \tag{18}$$

where $C_\ell = 2\sqrt{\pi\ell}\cdot \mathrm{e}^{1/6\ell}\cdot(5/2\mathrm{e})^{\ell/2} \leqslant 8$.

We will apply Lemma 3 to the random variables $Z_x = \mathbf{1}_{\{h(x)=y\}}$ where $x\in\mathrm{supp}(X)$ and y is fixed. Let $\delta(x,y) = \Pr_X[h(X)=y] - \Pr[U_m=y]$. We obtain

$$\begin{aligned}\mathop{\mathbf{E}}_{h\leftarrow\mathcal{H}}|\delta(x,y)|^\ell &= 2^{-k\ell}\cdot\mathbf{E}\left|\sum_{x\in\mathrm{supp}(X)} Z_x - \mathbf{E}\,Z\right|^\ell \\ &\leqslant 2^{-k\ell}\cdot C_\ell\cdot\left(\ell\cdot 2^{k-m}+\ell^2\right)^{\ell/2} \\ &= 2^{-k\ell}\cdot C_\ell\cdot\left(2^{k-m}\ell\right)^{\ell/2}\cdot\left(1+2^{m-k}\ell\right)^{\ell/2} \\ &\leqslant C_\ell\cdot \mathrm{e}^{2^{m-k}\ell^2/2}\cdot\left(2^{-k-m}\ell\right)^{\ell/2}\end{aligned}$$

and the result follows. □

5 Conclusion

We improved the security analysis of the TRNG of Barak et al. by carefully studying the deviation of the probability of hashing into a given location. The loss in the entropy amount seems to be optimal. An interesting problem for the future work is to propose different models for controlling the environment.

References

BR94. M. Bellare and J. Rompel, *Randomness-efficient oblivious sampling*, Proceedings of the 35th Annual Symposium on Foundations of Computer Science (Washington, DC, USA), SFCS '94, IEEE Computer Society, 1994, pp. 276–287.

BRS+10. Lawrence E. Bassham, III, Andrew L. Rukhin, Juan Soto, James R. Nechvatal, Miles E. Smid, Elaine B. Barker, Stefan D. Leigh, Mark Levenson, Mark Vangel, David L. Banks, Nathanael Alan Heckert, James F. Dray, and San Vo, *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*, Tech. report, Gaithersburg, MD, United States, 2010.

BST03. Boaz Barak, Ronen Shaltiel, and Eran Tromer, *True random number generators secure in a changing environment*, In Workshop on Cryptographic Hardware and Embedded Systems (CHES, Springer-Verlag, 2003, pp. 166–180.

BSW03. Boaz Barak, Ronen Shaltiel, and Avi Wigderson, *Computational analogues of entropy.*, RANDOM-APPROX, Lecture Notes in Computer Science, vol. 2764, Springer, 2003, pp. 200–215.

Gol08. Oded Goldreich, *Computational complexity: A conceptual perspective*, 1 ed., Cambridge University Press, New York, NY, USA, 2008.

GR05. Ariel Gabizon and Ran Raz, *Deterministic extractors for affine sources over large fields*, Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS '05, IEEE Computer Society, 2005, pp. 407–418.

GW96. Ian Goldberg and David Wagner, *Randomness and the netscape browser*, 1996.

KZ03. Jesse Kamp and David Zuckerman, *Deterministic extractors for bit-fixing sources and exposure-resilient cryptography*, Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS '03, IEEE Computer Society, 2003, pp. 92–.

Mar96. George Marsaglia, *DIEHARD: A battery of tests of randomness*, Technical report ??, Florida State University, Tallahassee, FL, USA, 1996.

NZ96. Noam Nisan and David Zuckerman, *Randomness is linear in space*, J. Comput. Syst. Sci. **52** (1996), no. 1, 43–52.

RTS00. Jaikumar Radhakrishnan and Amnon Ta-Shma, *Bounds for dispersers, extractors, and depth-two superconcentrators*, SIAM JOURNAL ON DISCRETE MATHEMATICS **13** (2000), 2000.

SV86. Miklos Santha and Umesh V. Vazirani, *Generating quasi-random sequences from semi-random sources*, Journal of Computer and System Sciences **33** (1986), no. 1, 75 – 87.

Wal11. John Walker, *Hotbits: Genuine random numbers, generated by radioactive decay*, January 2011.