On Entropy and Bit Patterns of Ring Oscillator Jitter

Markku-Juhani O. Saarinen
PQShield Ltd., Oxford, UK
mjos@pqshield.com

***Abstract*—Thermal jitter (phase noise) from a free-running ring oscillator is a common, easily implementable physical randomness source in True Random Number Generators (TRNGs). We show how to evaluate entropy, autocorrelation, and bit pattern distributions of ring oscillator noise sources, even with low jitter levels or some bias. Entropy justification is required in NIST 800-90B and AIS-31 testing and for applications such as the RISC-V entropy source extension. Our numerical evaluation algorithms outperform Monte Carlo simulations in speed and accuracy. We also propose a new lower bound estimation formula for the entropy of ring oscillator sources which applies more generally than previous ones.**

I. Introduction: Ring Oscillator Jitter

Free-running (ring) oscillators are widely used as physical noise sources in True Random Number Generators (TRNGs). In many ways, these designs are direct descendants of the oscillator-based "electronic roulette wheel" used to generate the RAND tables of random digits in the late 1940s [1].

A typical design (Fig. 1) has two oscillators; an unsynchronized ring oscillator and a reference oscillator that is used to sample bits from the free-running oscillator. Spontaneous and naturally occurring phase shifts between the oscillators will cause unpredictability of output bits. These random oscillator period variations are known as oscillator jitter [2].

A pioneering Ring Oscillator RNG chip was described and patented in 1984 by Bell Labs researchers [3], [4]. This type of noise source can be realized with "standard cells" in HDL and requires no special manufacturing processes, making it a popular choice. More modern versions are used as noise sources for cryptographic key generation in common microchips from AMD [5] and ARM [6].

Physical entropy sources are regulated in cryptographic security standards such as NIST's SP 800-90B [7] (for FIPS 140-3) and BSI's AIS 31 [8] (for Common Criteria). These mandate health monitoring (built-in statistical tests) and appropriate post-processing. Cryptographic post-processing methods such as the SHA2 hash [9] completely mask statistical defects while still allowing guessing attacks. Noise source entropy evaluation is therefore crucial for determining the sampling rate and "compression ratio" of the conditioner.

A. Physical Models and Their Limits

An important contributor to the randomness of jitter in a ring-oscillator inverter loop (Fig. 1) is Johnson-Nyquist thermal noise [10], [11], which occurs spontaneously in any conductor (regardless of quality) as a result of thermal agitation of free electrons. Jitter is a macroscopic manifestation of this quantum-level [12] Brownian effect.

Timing jitter is a relatively well-understood phenomenon for many reasons. It is an important limiting factor to the synchronous operating frequency of any digital circuit.

An example of a detailed physical model for ring oscillator phase noise and jitter is provided by Hajimiri et al. [13]–[15], which we recap here. The randomness of the timing jitter has a strongly Gaussian character. The jitter accumulates in the phase difference against the reference clock, with variance σ_t^2 growing almost linearly from one cycle to the next.

Under common conditions, the transition length standard deviation (uncertainty) σ_t after time t can be estimated for CMOS ring oscillators as (after [14, Eqns. 2.6,5.18]):

$$\sigma_t^2 = \kappa^2 t \approx \frac{8}{3\eta} \cdot \frac{kT}{P} \cdot \frac{V_{DD}}{V_{\text{char}}} \cdot t \tag{1}$$

In this derivation of physical jitter κ^2 we note especially the Boltzmann constant k and absolute temperature T; other variables include power dissipation P, supply voltage V_{DD}, device characteristic voltage V_{char}, and a proportionality constant $\eta \approx 1$. The number of stages (N) and frequency f affect power P via common dynamic (switching) power equations.

As noted in [14, Sect. 5.2.1], such derived models only express *"inevitable noise sources"* – not *"extra disturbance, such as substrate and supply noise, or noise contributed by extra circuitry or asymmetry in the waveform"* – which will increase jitter. Many of these factors are difficult to model individually or are beyond digital designers' control. In practice κ^2 is measured experimentally, and the existence of jitter (and hence, fresh thermal noise entropy) is continuously monitored by auxiliary circuits that are a part of the TRNG.

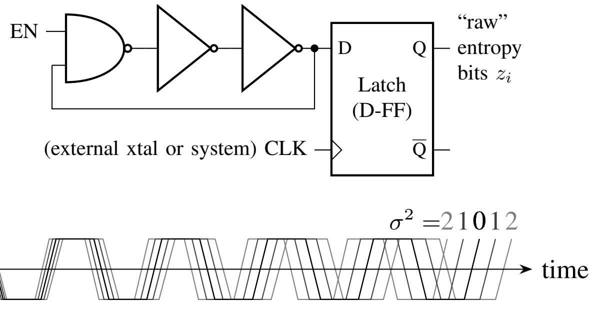

Fig. 1. A ring oscillator consists of an odd (here $N = 3$) number of inverters connected into a free-running loop. The output is sampled using an independent reference clock, such as a crystal oscillator. Transition times are affected by jitter (largely from Johnson-Nyquist thermal noise), whose accumulation causes samples to become increasingly unpredictable.

B. From Statistical Random Tests to Entropy Evaluation

A 1948 report by RAND [16] describes the statistical tests performed on the output of the "million digits" oscillator device [17]. The tests were based on work by Kendall and Smith [18], [19] with their late 1930s electromechanical random number device: Frequency test, Serial test, Poker test, and Gap test. It is remarkable that versions of these tests remained in use until the 2000s in the FIPS 140-2 standard [20].

While such "black box" statistical tests suites – including Marsaglia's DIEHARD and its successors [21], [22] and NIST SP 800-22 [23] — may be useful when evaluating pseudorandom generators for Monte Carlo simulations, they are poorly suited for security applications. It is illustrative that a test existed in NIST SP 800-22 even in 2010 to see if an LFSR is *"long enough"* to be *"considered random"* [23, Sect. 2.10]. Elementary cryptanalysis with finite field linear algebra shows that the internal state of an LFSR can be derived from a small amount of output, allowing both future and past outputs to be reproduced with little effort – a devastating scenario if that output is to be used for cryptographic keying.

By 2001 at least the German AIS 20/31 [8], [24] Common Criteria IT Security evaluations had diverged from the purely black-box statistical approach and instead concentrated on quantifying entropy produced by a noise source, evaluation of its post-processing methods, and also considered implementation security, cryptanalytic attacks, and vulnerabilities. Current NIST security evaluation methodology of physical noise sources [7] also acknowledges that general statistical properties of raw noise are less important than evaluation of its entropy content, but at the time of writing, do not require stochastic models or detailed analysis of physical sources.

For purposes of security engineering, pseudorandomness in the output of the physical source is an unambiguously negative feature as it makes the assessment of true entropy more difficult. On the other hand, Redundancy from a well-behaved stochastic model is easily manageable via cryptographic post-processing. Once seeded, standard (Cryptographic) Deterministic Random Bit Generators (DRBGs [25]) guarantee indistinguishability from random, in addition to providing prediction and backtracking resistance.

C. Ring Oscillators as Wiener Processes

Pioneering work on modern Physical RNG Entropy Estimation was presented by Killmann and Schindler [26], whose stochastic model uses independent and identically distributed transition times (half-periods) to model jitter. Baudet et al. [27] take a frequency domain (phase noise) approach. Our model broadly follows these and also the one by Ma et al. [28].

Baudet et al. propose a Shannon entropy lower bound [27, Eqn. 14], which has been used in engineering (e.g. [29]):

$$H_1 \geq 1 - \frac{4}{\pi^2 \ln 2} e^{-4\pi^2 Q} + O\big(e^{-6\pi^2 Q}\big). \qquad (2)$$

Here $Q = \sigma^2 \Delta t$ ("quality factor") corresponds to κ^2 in the physical model (Eqn. 1). We observe that the bound of Eqn. 2 is never lower than 0.415 even when Q approaches zero – this estimate is safe to use only under some additional assumptions.

D. Our Goals: FIPS 140-3 and More Generic ROs

Prior works generally state that the frequency of the free-running oscillator is much higher than sampling frequency and that they do not have a harmonic relationship. The source is also often taken to be unbiased and assumed to have a relatively high amount of entropy per sample. In this work, we show how to compute entropy, autocorrelation coefficients, and bit pattern probabilities also for less ideal parameters. Our goal is to have guarantees for entropy and min-entropy in TRNG designs. This is required in current cryptography standards AIS 31 [8] and FIPS 140-3 [30] / SP 800-90B [7] and for use in applications such as RISC-V Microprocessors [31], [32].

II. A Stochastic Model and its Distributions

We consider the jitter accumulation $\sigma^2 \sim Q$ at sample time rather than the variance of (half) periods [28, Sect. 2.2]. We also ease analysis by using the sampling period as a unit of time – sample z_i is at "time" i, and variance is defined accordingly. Our time-phase accumulation matches with the physical model (κ^2 of Eqn. 1) and also accounts for spontaneous, purely Brownian transitions and ripple when the relative frequency F of oscillators is very small or harmonic.

For sampled digital oscillator sources, we may ignore the signal amplitude and consider a pulse wave with period T and relative pulse width ("duty cycle") D. We assume a constant sampling rate and use the sample bits as a measure of time.

We normalize the sinusoidal phase ω as $x = \frac{\omega - \delta}{2\pi}$ to range $0 \leq x < 1$, where δ is the rising edge location. The average frequency $F \approx 1/T \bmod 1$ is a per-bit increment to the phase and σ^2 is its per-bit accumulated variance (Eqn. 1).

Definition 1 (Sampling Process): The behavior of a (F, D, σ^2) noise source and its bit sampler is modeled as:

$$x_i = \big(x_{i-1} + \mathcal{N}(F, \sigma^2)\big) \mod 1 \qquad (3)$$

$$z_i = \begin{cases} 1 & \text{if } x_i < D, \\ 0 & \text{if } x_i \geq D. \end{cases} \qquad (4)$$

Here $z_i \in \{0, 1\}$ is an output bit, and $x_i \in [0, 1)$ is the normalized phase at sampling time. F is the frequency in relation to the sampling frequency, and σ^2 represents jitter.

Due to normalization ($x \bmod 1 \equiv x - \lfloor x \rfloor$), and negative $-F$ symmetry, F can be reduced to range $[0, 1/2]$. One may view this as a "harmonic" reduction but there is no restriction for the sampler to run faster than the source oscillator.

The sampling process can be easily implemented to generate simulated bits $z_1, z_2, z_3, ..$ for given parameters (F, D, σ^2). This Wiener process is clearly only an idealized stochastic model, and its applicability for modeling specific physical random number generators must be individually evaluated.

A. Distance to Uniform

The Gaussian probability density function in Eqn. 3 becomes modularly wrapped (Fig. 2.) The classical assumption of ring oscillators is that if the accumulated variance σ^2 is large enough in relation to sampling rate, the modular step density function will become essentially "flat" in $[0, 1)$;

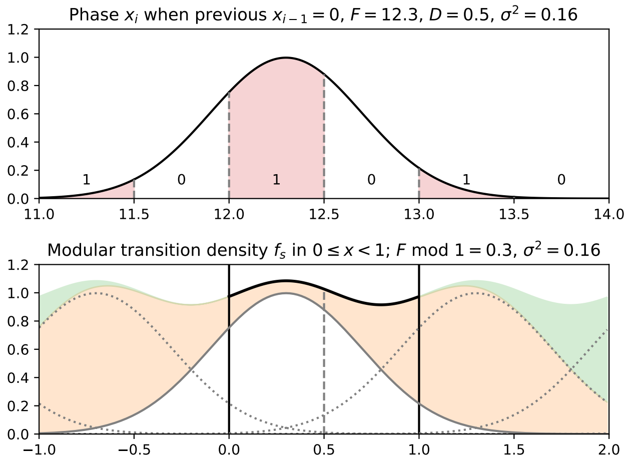

Fig. 2. Gaussian phase transition and equivalent modular density.

furthermore, if $(x_i - x_{i-1}) \bmod 1$ is uniformly random, then the bit sequence z_i is correlation-free. Some sources simply state ad hoc criteria for decorrelation (e.g. that $\sigma^2 > 1$).

We will calculate the step function's statistical distance to the uniform distribution. The density of the unbounded step function (Eqn. 3) can be equivalentl y defined over domain $0 \le x < 1$ or as a 1-periodic function in $\mathbb{R}/\mathbb{Z}$ (See Fig. 2):

$$f_s(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{i \in \mathbb{Z}} e^{-\frac{(x-F+i)^2}{2\sigma^2}}. \tag{5}$$

We have $f_s(a) = f_s(a+1)$ and $\int_a^{a+1} f_s(x)\,\mathrm{d}x = 1$ for all $a \in \mathbb{R}$. By choosing a tailcut value τ one can limit the sum to $\lfloor -\tau\sigma \rfloor \le i \le \lceil \tau\sigma \rceil$. This allows us to determine max at $f_s(F)$ and min at $f_s(F + 1/2)$ for given σ. These are bounds for its statistical (total variation) distance to the uniform distribution (See Table I.) We see that this idealized "1-dimensional lattice Gaussian" would be cryptographically uniform at $\sigma^2 > 9$.

TABLE I
EXTREMA OF THE PROBABILITY DENSITY FUNCTION $f_s(x)$ FOR SOME σ.

σ	f_s min	f_s max
0.10	0.000030	3.989423
0.20	0.175283	1.994726
0.30	0.663191	1.340089
0.50	0.985616	1.014384
0.75	0.999970	1.000030

σ	f_s range
1.00	$1 \pm 2^{-27.47766}$
1.50	$1 \pm 2^{-63.07473}$
2.00	$1 \pm 2^{-112.9106}$
2.50	$1 \pm 2^{-176.9854}$
3.00	$1 \pm 2^{-255.2989}$

B. Autocorrelation and Sampling Intervals

We define a scaled, binary delay-k autocorrelation measure $-1 \le C_k \le +1$:

$$C_k = 2\Pr(z_i = z_{i+k}) - 1. \tag{6}$$

We may estimate C_k, $k \ge 1$ for a finite m-bit sequence as

$$C'_k = \frac{1}{m-k} \sum_{i=1}^{m-k} (2z_i - 1)(2z_{i+k} - 1). \tag{7}$$

For convenience, we set $C'_0 = \frac{1}{m}\sum_{i=1}^{m}(2z_i - 1)$ to represent simple bias in the same vector; C'_0 approximates $2D - 1$.

Theorem 1: With fixed D and $\sigma^2 > 0$ or $F \notin \mathbb{Q}$ we have

$$C_k(F, D, \sigma^2) = C_1(kF \bmod 1, D, k\sigma^2) \text{ for } k \ge 1. \tag{8}$$

Proof 1: The variance of independent random variables is additive by induction in k, as is the mean. The difference $x_k - x_0$ will then have the distribution $\mathcal{N}(kF, k\sigma^2) \bmod 1$. Only with either noisy or non-rational (non-harmonic) F we may take x_0 in Equation 3 to be uniformly distributed in $[0, 1)$.

C. Computing C_k to High Precision Without Simulation

Let p_{00}, p_{01}, p_{10}, p_{11} be frequencies of adjacent bit pairs $p_{(z_i, z_{i+1})}$ present in bit sequence z_i (Eqn. 4) in the model.

We'll pick one, $p_{11} = \Pr(z_i = 1 \text{ and } z_{i+1} = 1)$. The condition $z_i = 1$ limits the density of x_i to "boxcar" g_1:

$$g_1(x) = \begin{cases} 1 & \text{if } x \in [0, D) \\ 0 & \text{if } x \notin [0, D). \end{cases} \tag{9}$$

We also define $g_0(x) = 1$ if $x \in [D, 1)$ and zero elsewhere.

The addition of random variables corresponds to convolution of their density functions; convolution $f_1 = g_1 * f_s$ with the step function f_s (Eqn. 5) yields the probability density of x_{i+1} conditioned on $x_i = 1$. The probability mass of the second bit $z_{i+1} = 1$ is in range $x_{i+1} \in [0, D)$ and we have

$$p_{11} = \int_0^D f_1(x)\,\mathrm{d}x. \tag{10}$$

Convolution $f_1 = g_1 * f_s$ density can be expressed as

$$f_1(x) = \frac{1}{2} \sum_{i \in \mathbb{Z}} \left[\operatorname{erf}(a_i) - \operatorname{erf}(b_i) \right] \tag{11}$$

Where $a_i = (x+i-F)/\sqrt{2\sigma^2}$ and $b_i = (x+i-F-D)/\sqrt{2\sigma^2}$. An indefinite integral $S_1 = \int f_1(x)\,\mathrm{d}x$ with the same a_i,b_i is

$$S_1(x) = \frac{\sqrt{2\sigma^2}}{2} \sum_{i \in \mathbb{Z}} \left[a_i \operatorname{erf}(a_i) - b_i \operatorname{erf}(b_i) + \frac{e^{-a_i^2} - e^{-b_i^2}}{\sqrt{\pi}} \right]. \tag{12}$$

Again, one can choose a tailcut bound τ for desired precision $\epsilon \approx \operatorname{erfc}(\tau/\sqrt{2})$ (via Gaussian CDF) and compute the sums just over the integer range $\lfloor -\tau\sigma \rfloor \le i \le \lceil \tau\sigma \rceil$. A typical choice for IEEE floating point is $\tau = 10$ ("ten sigma").

Choosing $p_{11} = \int_0^D f_1(x)\,\mathrm{d}x = S_1(D) - S_1(0)$ has some computational advantages. From p_{11} we can derive other parameters $p_{01} = p_{10} = D - p_{11}$, $p_{00} = 1 - 2D + p_{11}$, and $C_1 = 4(p_{11} - D) + 1$. To compute arbitrary C_k, substitute parameters $F' = kF \bmod 1$ and $\sigma'^2 = k\sigma^2$ (Thm. 1). We then have C_k as $C'_1 = 4[S_1(D) - S_1(0) - D] + 1$.

Figure 3 shows the density functions g for the four bit pair frequencies when $F = 0.1$, $D = 0.625$, $\sigma^2 = 0.04$ ($\sigma = 0.2$). The dotted line on upper boxes corresponds to shape of f_0 and lower row to f_1; these have been chopped (shaded area) to $g_{00}, g_{01}, g_{10}, g_{11}$. Note that even though g_{10} has a different shape from g_{01}, they have equivalent area and hence frequency $p_{01} = p_{10} = \frac{1-C_1}{4}$. This is natural since the frequency of rising edges must match the frequency of falling edges.

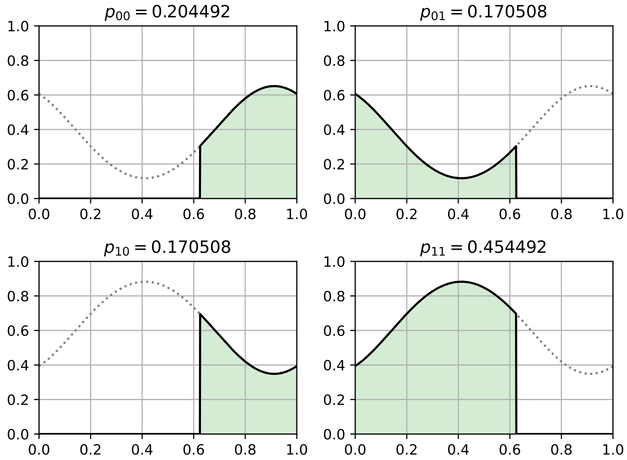

Fig. 3. Bit pair probabilities for $F = 0.1$, $D = 0.625$, $\sigma^2 = 0.04$.

D. *Use of C_k in Modeling of Physical Sources*

The output from bit generation simulations agrees with these explicit autocorrelation values as expected. Analytic C_k is of course much faster to compute.

Since autocorrelation estimates C'_k (Eqn. 7) may also be easily derived from the output of physical ring oscillators, we can find a good approximate (F, D, σ^2) model for a physical source by matching their autocorrelation properties. We use least-squares minimization of few initial entries of autocorrelation vectors for this type of modeling.

We can also experimentally derive parametrized models where frequency F and jitter σ^2 are functions of environmental aspects such as temperature or aging of circuitry; this, in turn, allows us to extrapolate and assign safe bounds for statistical health tests parameters and entropy output (yield of conditioner) over the lifetime of the device.

E. *Bit Pattern Probabilities via FFT Convolutions*

To compute probabilities of bit triplets and beyond, we may "chop" a density function to zero the part which we know to be conditioned out; $g_{10}(x) = (f_1 \cdot g_0)(x)$, is f_1 chopped to zero outside $[D, 1)$ range so we have $\int_{-\infty}^{\infty} g_{10}(x)\,\mathrm{d}x = p_{10}$.

Let z be a sequence of bits for which conditional distribution f_z is known; "chopping" with g_0 or g_1 and convoluting with step function f_s we obtain distributions of one additional bit: $f_{z,0} = (f_z \cdot g_0) * f_s$ and $f_{z,1} = (f_z \cdot g_1) * f_s$. This chop-and-convolute process can be continued to determine the probability and phase distribution of an arbitrary bit pattern.

Direct probability distribution integration formulas such as Eqn. 12 become cumbersome for more generic bit patterns. We instead perform numeric computations on probability density functions f_z represented as real-valued polynomial coefficients. This approach is attractive since the Fast Fourier Transform offers an especially efficient way to compute *cyclic* convolutions of polynomials, as is required by our 1-periodic step function f_s (Eqn. 5). Unlike unbounded Gaussians our random variables $x_i \in [0, 1)$ have a strictly limited range.

Algorithm 1 Evaluate bit pattern probability p_z

```
function PZFFT($F$, $D$, $\sigma^2$, $z_1 z_2 \cdots z_n$)
    for $j \leftarrow 0, 1, \cdots, m-1$ do                ▷ Init: Approximation.
        $s_j \leftarrow \frac{1}{m} f_s(\frac{j}{m})$            ▷ Eqn. 5 for $F$ and $\sigma^2$.
        $g_{1,j} \leftarrow \max(\min(mD - j, 1), 0)$   ▷ Eqn. 9 for $D$.
        $g_{0,j} \leftarrow 1 - g_{1,j}$                   ▷ Select zero – inverse.
        $v_j \leftarrow \frac{1}{m}$              ▷ Start with uniform distribution.
    end for
    for $i \leftarrow 1, 2, \cdots, n$ do                        ▷ For each bit.
        for $j \leftarrow 0, 1, \cdots, m-1$ do                  ▷ Chop half.
            $t_j \leftarrow v_j g_{z_i,j}$              ▷ Note bit select index $z_i$.
        end for
        $v \leftarrow t * s \bmod (x^m - 1)$          ▷ Convolution (FFT).
    end for
    return $p_z = \sum_{i=0}^{m-1} v_i$                   ▷ Probability mass.
end function
```

These probability density functions $f(x)$ correspond to real-valued m-degree polynomials $v = \sum_{i=0}^{m} x^i v_i$ in Algorithm 1. The unit interval domain $x \in [0, 1)$ is mapped to coefficients via $v_i \approx \int_{i/m}^{(i+1)/m} f(x)\,\mathrm{d}x$. For the step function of we approximate this as $s_i = \frac{1}{m} f_s(i/m)$ and for chop functions so that $\sum_i g_{1,i} = D$ and $\sum_i g_{0,i} = 1 - D$. We write the circular convolution using polynomial product and reduction modulo $x^m - 1$, which can be very efficiently computed with FFT.

The chopping operation is a point-by-point multiplication with g_0 or g_1 in the normal (time) domain, while step convolution is a point-by-point multiplication with $\hat{f}_s$ in the transformed (complex, frequency) domain, and hence each additional bit z_i requires one forward and one inverse transform as $\hat{f}_s$ remains the same. Our open-source, FFTW3-based [33] portable C implementation allows accurate computation of probabilities of almost arbitrarily long patterns [1].

III. Entropy Evaluation

Let Z_n be a random variable representing n-bit sequences $z = (z_1, z_2, ..z_n)$ which are sequentially generated by the stochastic process of Sect. II characterized by stationary parameters (F, D, σ^2). Each of 2^n possible outcomes z can be assigned a probability $p_z = \Pr(Z_n = z)$.

The NIST SP 800-90B [7] entropy source standard focuses on min-entropy H_∞, a member of the Rényi family of entropies [34]. Min-entropy (or "worst-case entropy") has a simple definition in case of a discrete variable, based on the likelihood of the most likely outcome of Z_n:

$$H_\infty(Z_n) = \min_z(-\log_2 p_z) = -\log_2(\max_z p_z) \qquad (13)$$

The AIS 31 [8] Common Criteria evaluation method additionally uses the traditional Shannon entropy metric

$$H_1(Z_n) = -\sum_z p_z \log_2 p_z. \qquad (14)$$

For Shannon entropy we consider its *entropy rate* $H(Z)$. This is a $[0, 1]$-valued limit $H(Z) = \lim_{n\to\infty} \frac{1}{n} H_1(Z_n)$.

[1] Reference source code: https://github.com/mjosaarinen/bitpat

A. Entropy Upper Bounds

Probabilities p_z obtained via Algorithm 1 and related techniques in Sect. II-E can be substituted to Eqns. 13 and 14 to evaluate $H_\infty(Z_n)$ and $H_1(Z_n)$, respectively.

Shannon entropy $H_1(Z_n)$ provides increasingly accurate upper bounds since we have $H_\infty(Z_n) \leq H_1(Z_n)$ and

$$H(Z) \leq \ldots \leq \frac{1}{3}H_1(Z_3) \leq \frac{1}{2}H_1(Z_2) \leq H_1(Z_1). \tag{15}$$

This relationship follows from subadditivity of joint entropy in case of Shannon entropy H_1; the monotonic relationship of Eqn. 15 does not hold for min-entropy H_∞.

All Rényi entropies are upper bounded by max-entropy (Hartley entropy) H_0, i.e. the number (cardinality) of n-bit z with nonzero probability; $H_0(Z_n) = \log_2 |p_z > 0|$. If an m-bit encoding exists for all elements with $p_z > 0$ of Z_n, then its cardinality is at most 2^m and $H(Z) \leq H_0(Z_n) \leq m$.

This leads to limit $H(Z) \to 0$ for a noiseless ($\sigma^2 = 0$) source, regardless of F oscillation. A simple ϵ, δ argument shows that each n-bit sequence z_i with $\sigma^2 = 0$ can be encoded by expressing F, D, and x_0 with asymptotic $O(\log n)$ bits of precision. Claim follows from $\lim_{n\to\infty} \frac{1}{n}\log n \to 0$.

B. Entropy Lower Bounds as a function of σ^2

For an entropy lower bound we consider the entropy contribution of jitter to an individual bit z_i when all of the parameters (F, D, σ^2) and the previous phase x_{i-1} are known (in addition to previous bit z_{i-1}). Let $p_e = \Pr(z_i = z'_i)$ where the expected bit value z'_i is deterministic (from $x_{i-1} + F$).

We observe that F cancels out in this case and we have $p_e = p_{00} + p_{11}$ with $F = 0$ for equations of Section II-C. In case of an unbiased source $D = \frac{1}{2}$, a further simplification yields frequency-independent bounds as a function of σ^2:

$$p_e = 2 \cdot [S_1(1/2) - S_1(0)] = 4 \cdot S_1(1/2) \tag{16}$$

$$H_1(Z) \geq -p_e \log_2 p_e - (1 - p_e) \log_2(1 - p_e) \tag{17}$$

$$H_\infty(Z) \sim -\log_2 p_e. \tag{18}$$

where S_1 is Eqn. 12 with $F = 0$, $D = \frac{1}{2}$. From Eqn. 16 we can show a looser approximate bound $p_e \leq 1 - \frac{1}{2}\tanh(\pi\sigma)$.

These estimates are lower than some previously proposed lower bounds (See Eqn. 2) as they are based on fewer assumptions. Crucially they cover the entire range of σ^2 – and are therefore safer to use in cryptographic engineering.

C. Min-Entropy Estimates

One part of min-entropy estimation of $H_\infty(Z_n)$ is to find a maximum-likelihood bit sequence z, and the second is to determine its probability p_z. The second part can be accomplished with Algorithm 1 – we have $H_\infty(Z_n) = -\frac{1}{n}\log_2 p_z$.

A reasonable z string "guess" is to select x_0 at random and use the peak probability path $x_i = x_{i-1} + F \pmod 1$ to determine $z_1, z_2, \cdots, z_n$. This approach is asymptotically sound, but overestimates entropy for small n.

A practical depth-first approach is to proceed as in Alg. 1 but evaluate weights $q_0 = \sum_{j=0}^{m-1} v_j g_{0,j}$ and $q_1 = \sum_{j=0}^{m-1} v_j g_{1,j}$ at each step i, and select z_i with the higher probability mass.

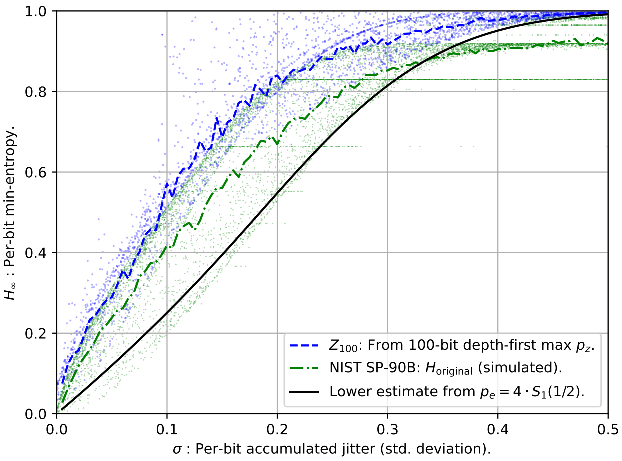

Fig. 4. Min-entropy from distribution of Z_{100} with depth-first z selection, NIST SP 800-90B estimates, and our p_e bit-prediction lower bound. Experimental data is represented as a scatter plot, with a line at the average.

While $\max p_z$ can usually be found with subexponential z guesses, worst-case complexity of this problem remains open. Certainly, a simple depth-first search will not always work. Consider $\max p_z$ for source $(D = \frac{1}{2}, F = 0.15, \sigma^2 = 0.04)$:

$$\begin{aligned}
\tfrac{1}{3}H_\infty(Z_3) &= 0.844807, & p_{000} &= p_{111} = 0.172609. \\
\tfrac{1}{4}H_\infty(Z_4) &= 0.849297, & p_{0000} &= p_{1111} = 0.0949171. \\
\tfrac{1}{5}H_\infty(Z_5) &= 0.846341, & p_{00011} &= p_{00111} = \\
& & p_{11000} &= p_{11100} = 0.0532267.
\end{aligned}$$

We first note that the entropy increase from Z_3 violates subadditivity (and would not be possible for H_1; Eqn. 15). Furthermore, the maximum-probability bit strings of Z_4 are not substrings of those for Z_5; not reachable via iteration.

D. Comparison to SP 800-90B Estimation

Current SP 800-90B min-entropy estimation methodology [7, Section 3.1.3] used by FIPS 140-3 [30] proceeds by taking the minimum of ten conservative, standardized entropy tests. The results of this methodology plateau below $H_\infty \approx 0.9$ even for completely random (cryptographically generated) test data. Use of stochastic models is also suggested (for $H_{\text{submitter}}$).

We generated 16,000 simulated sequences of $8 * 10^6$ bits with random σ and $F \in [0, 1/2]$, and subjected them to the official NIST SP 800-90 Entropy Assessment[2]. Fig. 4 contrasts these results with min-entropy estimate for Z_{100} where z is chosen to follow maximum probability mass (Section III-C).

As expected, the black-box heuristic which has been designed to *"lean toward a conservative underestimate of min-entropy"* [7, Sect G.2] reports less entropy than our estimates.

Fig. 4 also shows $H_\infty(Z) \sim -\log_2 p_e$ min-entropy derived from the bit-prediction bound of Eqns. 16 and 18. This curve mostly traces the lower reaches of the stochastic model estimates (which are scattered here due to randomness of F). We suggest that this is a safe min-entropy engineering estimate from variance σ^2, assuming an unbiased source ($D = 1/2$).

[2] NIST: https://github.com/usnistgov/SP800-90B_EntropyAssessment

Acknowledgments

Author thanks Joshua E. Hill for helpful comments. This work has been supported by Innovate UK Research Grant 105747.

References

[1] George W. Brown. History of RAND's random digits – summary. Research Paper P-113, RAND Corporation, June 1949. Also appeared in: Monte Carlo Method, Nat. Bur. Stand., Appl. Math. Series 12 (1951), pp. 31-32. URL: https://www.rand.org/pubs/papers/P113.html.

[2] John A. McNeill and David S. Ricketts. *The Designer's Guide to Jitter in Ring Oscillators*. Springer, 2009. doi:10.1007/978-0-387-76528-0.

[3] Robert C. Fairfield, Robert L. Mortenson, and Kenneth B. Coulthart. An LSI random number generator (RNG). In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 203–230. Springer, 1984. doi:10.1007/3-540-39568-7_18.

[4] Kenneth B. Coulthart, Robert C. Fairfield, and Robert L. Mortenson. Random number generator. U.S. Patent 4,641,102 A, August 1984. URL: https://patents.google.com/patent/US4641102A.

[5] AMD. AMD random number generator. AMD TechDocs, June 2017. URL: https://www.amd.com/system/files/TechDocs/amd-random-number-generator.pdf.

[6] ARM. ARM TrustZone true random number generator (TRNG): Technical reference manual. ARM 100976_0000_00_en (Second release r0p0 0000-01), May 2020. URL: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.100976_0000_00_en.

[7] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation. NIST Special Publication SP 800-90B, January 2018. doi:10.6028/NIST.SP.800-90B.

[8] Wolfgang Killmann and Werner Schindler. A proposal for: Functionality classes for random number generators. AIS 20 / AIS 31, Version 2.0, English Translation, BSI, September 2011. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.html.

[9] NIST. Secure hash standard (SHS). Federal Information Processing Standards Publication FIPS 180-4, August 2015. doi:10.6028/NIST.FIPS.180-4.

[10] John Bertrand Johnson. Thermal agitation of electricity in conductors. *Phys. Rev.*, 32(1):97–109, July 1928. doi:10.1103/PhysRev.32.97.

[11] Harry Nyquist. Thermal agitation of electric charge in conductors. *Phys. Rev.*, 32(1):110–113, July 1928. doi:10.1103/PhysRev.32.110.

[12] Herbert B. Callen and Theodore A. Welton. Irreversibility and generalized noise. *Phys. Rev.*, 83(1):34–40, July 1951. doi:10.1103/PhysRev.83.34.

[13] Ali Hajimiri and Thomas H. Lee. A general theory of phase noise in electrical oscillators. *IEEE Journal of Solid-State Circuits*, 33(2):179–194, 1998. doi:10.1109/4.658619.

[14] Ali Hajimiri and Thomas H. Lee. *The Design of Low Noise Oscillators*. Kluwer, 1999. doi:10.1007/b101822.

[15] Ali Hajimiri, Sotirios Limotyrakis, and Thomas H. Lee. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits*, 34(6):790–804, June 1999. URL: https://authors.library.caltech.edu/4916/1/HAJieeejssc99a.pdf, doi:10.1109/4.766813.

[16] Bernice B. Brown. Some tests on the randomness of a million digits. Research Paper P-44, RAND Corporation, October 1948. URL: https://www.rand.org/pubs/papers/P44.html.

[17] RAND Corporation. *A Million Random Digits with 100,000 Normal Deviates*. Free Press, 1955. URL: https://www.rand.org/pubs/monograph_reports/MR1418.html.

[18] Maurice G. Kendall and Bernard Babington-Smith. Randomness and random sampling numbers. *Journal of the Royal Statistical Society*, 101(1):147–166, 1938. doi:10.2307/2980655.

[19] Maurice G. Kendall and Bernard Babington-Smith. Second paper on random sampling numbers. *Supplement to the Journal of the Royal Statistical Society*, 6(1):51–61, 1939. doi:10.2307/2983623.

[20] NIST. Security requirements for cryptographic modules. Federal Information Processing Standards Publication FIPS 140-2 (With change notices dated October 10, 2001 and December 3, 2002), May 2001. doi:10.6028/NIST.FIPS.140-2.

[21] George Marsaglia. The Marsaglia random number CDROM including the diehard battery of tests of randomness. CDROM and Online Publication, 1995.

[22] Robert G. Brown, Dirk Eddelbuettel, and David Bauer. Dieharder: A random number test suite. Software distribution, accessed January 2021, 2003. URL: https://webhome.phy.duke.edu/~rgb/General/dieharder.php.

[23] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, JamesDray, and San Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications, April 2010. doi:10.6028/NIST.SP.800-22r1a.

[24] Werner Schindler and Wolfgang Killmann. Evaluation criteria for true (physical) random number generators used in cryptographic applications. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 431–449. Springer, 2002. doi:10.1007/3-540-36400-5_31.

[25] Elaine Barker and John Kelsey. Recommendation for random number generation using deterministic random bit generators. NIST Special Publication SP 800-90A Revision 1, June 2015. doi:10.6028/NIST.SP.800-90Ar1.

[26] Wolfgang Killmann and Werner Schindler. A design for a physical RNG with robust entropy estimators. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 146–163. Springer, 2008. doi:10.1007/978-3-540-85053-3_10.

[27] Mathieu Baudet, David Lubicz, Julien Micolod, and André Tassiaux. On the security of oscillator-based random number generators. *J. Cryptology*, 24(2):398–425, 2011. doi:10.1007/s00145-010-9089-3.

[28] Yuan Ma, Jingqiang Lin, Tianyu Chen, Changwei Xu, Zongbin Liu, and Jiwu Jing. Entropy evaluation for oscillator-based true random number generators. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 544–561. Springer, 2014. doi:10.1007/978-3-662-44709-3_30.

[29] Oto Petura, Ugo Mureddu, Nathalie Bochard, Viktor Fischer, and Lilian Bossuet. A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices. In Paolo Ienne, Walid A. Najjar, Jason Helge Anderson, Philip Brisk, and Walter Stechele, editors, *26th International Conference on Field Programmable Logic and Applications, FPL 2016, Lausanne, Switzerland, August 29 - September 2, 2016*, pages 1–10. IEEE, 2016. URL: https://ieeexplore.ieee.org/xpl/conhome/7573873/proceeding, doi:10.1109/FPL.2016.7577379.

[30] NIST and CCCS. Implementation guidance for FIPS 140-3 and the cryptographic module validation program. CMVP, August 2021. URL: https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf.

[31] Markku-Juhani O. Saarinen, G. Richard Newell, and Ben Marshall. Development of the RISC-V entropy source interface. *Journal of Cryptographic Engineering*, page to appear, 2021. URL: https://eprint.iacr.org/2029/866.

[32] Ben Marshall, editor. *RISC-V Cryptographic Extension Proposals. Volume I: Scalar & Entropy Source Instructions*. RISC-V International, 2021. URL: https://github.com/riscv/riscv-crypto.

[33] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proc. IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation". doi:10.1109/JPROC.2004.840301.

[34] Alfréd Rényi. On measures of entropy and information. In Jerzy Neyman, editor, *Proc. Fourth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*, pages 547–561. University of California Press, 1961. URL: https://projecteuclid.org/euclid.bsmsp/1200512181.