On the Entropy of Oscillator-Based True Random Number Generators

Yuan Ma[1,2], Jingqiang Lin[1,2,3], and Jiwu Jing[1,2,3]

[1] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
{yma,linjq,jing}@is.ac.cn

Abstract. True random number generators (TRNGs) are essential for cryptographic systems, and they are usually evaluated by the concept of entropy. In general, the entropy of a TRNG is estimated from its stochastic model, and reflected in the statistical results of the generated raw bits. Oscillator-based TRNGs are widely used in practical cryptographic systems due to its elegant structure, and its stochastic model has been studied in different aspects. In this paper, we investigate the applicability of the different entropy estimation methods for oscillator-based TRNGs, including the *bit-rate entropy*, the *lower bound* and the *approximate entropy*. Particularly, we firstly analyze the two existing stochastic models (one of which is phase-based and the other is time-based), and deduce consistent bit-rate entropy results from these two models. Then, we design an approximate entropy calculation method on the output raw bits of a simulated oscillator-based TRNG, and this statistical calculation result well matches the bit-rate entropy from stochastic models. In addition, we discuss the extreme case of tiny randomness where some methods are inapplicable, and provide the recommendations for these entropy evaluation methods. Finally, we design a hardware verification method in a real oscillator-based TRNG, and validate these estimation methods in the hardware platform.

Keywords: Oscillators, true random number generators, entropy estimation, stochastic model

1 Introduction

Random number generators (RNGs) are widely used in cryptographic systems to generate sensitive parameters, such as keys, seeds of pseudo-random number generators, and initialization vectors. The security of many cryptographic schemes and protocols is built on the randomness of RNGs. The output of a RNG is expected to be a bit sequence with the properties of *unbiasedness*, *independence* and *unpredictability*. Statistical tests (such as NIST SP 800-22 [13] and

Diehard [11]) cannot evaluate the unpredictability of the sequence, as deterministic sequences with good statistical properties are able to pass the statistical tests.

The concept of *entropy*, which measures the uncertainty in bits (e.g., bit-rate entropy), is used to evaluate the unpredictability of a RNG. For a true RNG (TRNG), the predictability comes from the randomness of physical noises. The international standard ISO/IEC 18031 [7] and Germany standard AIS 31 [8] recommend to establish the entropy estimator with a stochastic model for TRNG evaluation. The stochastic model describes the extraction process from physical random noises to digitized random bits based on reasonable physical assumptions.

Oscillator-based sampling is a typical structure adopted by many TRNG designs, and the stochastic models of oscillator-based TRNGs have been well studied in recent years. To figure out the entropy of oscillator-based TRNGs, Killmann and Schindler [9] established a common stochastic model by a *time-based* approach, and gave a tight lower bound of the entropy; using the similar approach, Ma *et al.* [10] presented a calculation method to obtain the precise entropy. In addition, Amaki *et al.* [1] calculated the probabilities of certain bit patterns by using a Markov state transition matrix, but they evaluated the security using the Poker test [6] rather than entropy estimation. Baudet *et al.* [2] proposed a *phase-based* approach and provided a concise analytical formula for the entropy calculation (including the n-bit entropy and the lower bound). The entropy can be rapidly figured out by substituting the TRNG design parameters, including the jitter ratio and the frequencies of the sampling and sampled signals. This formula is then employed to estimate the entropy for a sufficient-entropy TRNG design [4].

While the entropy is estimated with these stochastic models based on the TRNG design parameters, the approximate entropy (ApEn) is obtained *statistically* based on the output bit sequence of a TRNG. ApEn is calculated by comparing the distributions of m-bit and $(m+1)$-bit blocks in the bit sequence. However, the parameter m in ApEn shall be chosen carefully to trade off between the accuracy of entropy estimation and the computation complexity.

Although various entropy estimation methods have been proposed in literature, a comprehensive and systematical study for their accuracy and applicability (e.g., the consistency of different methods, the estimation error between theory and experiment, the extreme cases of design parameters) is still lacking. In this paper, we investigate the applicability of different entropy calculation methods for oscillator-based TRNGs, including the bit-rate entropy, the lower bound and the approximate entropy. Particularly, we make the following contributions.

- We present two bit-rate entropy calculation methods based on the time-based and phase-based n-bit entropy stochastic models [2, 10], respectively. The results are analyzed, and we deduce consistent bit-rate entropy results from these two models by expanding the original analytical expression.
- We propose an approximate entropy calculation method for the output bit sequence of oscillator-based TRNGs, where the parameter m is obtained

from the autocorrelation coefficient of the bit sequence. The ApEn calculation result of a simulated oscillator-based TRNG well matches the bit-rate entropy from stochastic models, which confirms the correctness of the theoretical results.
- We investigate the applicability of these entropy estimation methods in the extreme case with tiny randomness (i.e., the accumulated jitter is very small within the sampling interval). As it is possible to make an overestimation of the entropy in such case, we provide an alternative method to acquire a conservative estimation for the entropy.
- We design a hardware verification method in a real oscillator-based TRNG. In the experiment, we calculate the randomness factor under the white noise, and validate these estimation methods in the hardware platform.

The rest of the paper is organized as follows. In Section 2, we introduce the preliminary about the principle and existing entropy estimation methods for oscillator-based TRNGs. We propose our evaluation method on the different types of entropy in Section 3. In Section 4, we present the evaluation results and investigate the case of tiny randomness. In Section 5, we investigate the effectiveness of the estimation methods in the hardware platform. In Section 6, we conclude the paper.

2 Preliminary

In this section, we first introduce the principle of oscillator-based TRNGs. Then we summarize the methods of entropy estimation. The types of entropy include n-bit entropy, lower bound of entropy and approximate entropy.

2.1 Oscillator-based TRNGs

The basic structure of oscillator-based TRNGs contains an unstable oscillator generating a fast oscillating signal with jitter, and a sampling reference clock that is assumed without jitter, as shown in Figure 1. The randomness comes from jitter in the fast signal periods that is caused by noises. In general, the noises that affect jitter are assumed to be independent and identically distributed (i.i.d.) for the simplicity of the model. As an exception, the model of [9] partially allows short-term dependency in the half-periods of the fast oscillating signal.

We firstly present some definitions of the parameters in the aspect of time evolution. The half-periods of fast oscillating signal are assumed to be i.i.d. with mean $m_X = E(X_k)$ and variance $s_X^2 = V(X_k)$, where X_k is the k-th half-period. The fixed sampling interval is denoted as Δt.

As the tiny jitter ($s_X/m_X \ll 1$) accumulates within the sampling interval, the probability of guessing the sampling point lying in the high or low voltage is decreasing. Hence, the jitter ratio and the frequency ratio jointly determine the quality of this type of TRNG, and the integrated factor is often called as *quality factor* [2, 10]. Another considerable factor is the divisibility of the half-period

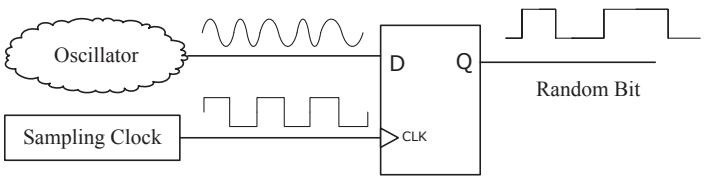

Fig. 1. The basic structure of oscillator-based TRNGs

m_X to the sampling interval Δt, which is measured using variable $r = \Delta t/m_X \mod 1$. The divisibility increases when r approaches to 0.5 from either 0 or 1 and reaches its maximum at $r = 0.5$. The cases of $r = 0$ and $r = 0.5$ represent the worst and the best case of the TRNG output quality, respectively. This property has been discovered in [2, 10, 1].

2.2 *n*-bit entropy

The n-bit entropy represents the amount of entropy for n-bit output random sequences. In general, there are two methods to get the n-bit entropy, time-based and phase-based. The basic idea is to calculate the probability of n-bit pattern, which is denoted as $p(\boldsymbol{b})$, from the stochastic model, and then iterate all the patterns to get n-bit entropy via Equation (1).

$$H_n = \sum_{\boldsymbol{b}\in\{0,1\}^n} -p(\boldsymbol{b}) \log p(\boldsymbol{b}). \tag{1}$$

Time-based method. Ma *et al.* [10] use the classic model of [9] in the aspect of time evolution. They utilize the waiting time W_i to represent the relationship between the adjacent sampling bits, where W_i is the distance of the i-th sampling position to the closest following edge of fast oscillating signal. They use a set of conditional probability functions to calculate the n-bit pattern probability by iterating, and eliminate W_i from the final expression by probability integration for the uniform distribution of W_i. Here we do not list the detailed computing process. Furthermore, they gave several curves from the worst to the best case to demonstrate the entropy variation using numerical computation, but an analytical probability or entropy expression was not given in their work.

Phase-based method. Baudet *et al.* [2] use the phase-oriented approach to model the stochastic behavior of the oscillating signal. The phase evolution of an oscillation is modeled by a Wiener stochastic process $\varphi(t)$ with drift $\mu > 0$ and volatility $\sigma^2 > 0$. The parameters are equivalent to the time-based definitions following the equations: $\mu = \frac{1}{2m_X}$ and $\sigma^2 = \frac{s_X^2}{4m_X^3}$.

Another quality factor is denoted as $Q = \sigma^2 \Delta t = \frac{s_X^2 \Delta t}{4m_X^3}$. The frequency ratio of the fast signal to the slow one is denoted as $\nu = \mu \Delta t = \frac{\Delta t}{2m_X}$, so $r = 2\nu \mod 1$. Note that, as the investigated target is the same as the time-based method, two

sets of parameters can be converted to each other. The quality fact Q in the phase-based method equals $4q^2$, where $q = \sqrt{\frac{\Delta t}{m_X}} \cdot \frac{s_X}{m_X}$ is the parameter defined in the time-based model [10]. For convenience, we use Q and r to compute the entropy for either time-based or phase-based method in the subsequent.

The following two formulas computing the probability and n-bit entropy are provided in their work, where $B = e^{-2\pi^2 Q}$.

1. The probability to output a vector $\mathbf{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$ satisfies

$$p(\mathbf{b}) = \frac{1}{2^n} + \frac{8}{2^n \pi^2}\left(\sum_{j=1}^{n-1}(-1)^{b_j + b_{j+1}}\right)\cos(2\pi\nu)B + O(B^2). \tag{2}$$

2. The entropy of such an output is

$$H_n = \sum_{\mathbf{b}\in\{0,1\}^n} -p(\mathbf{b})\log p(\mathbf{b}) = n - \frac{32(n-1)}{\pi^4 \ln(2)}\cos^2(2\pi\nu)B^2 + O(B^3). \tag{3}$$

2.3 Lower Bound of Entropy

Min-entropy or lower bound of entropy is the most conservative measurement of entropy, and is useful in determining the worst-case entropy of a TRNG. In the aspect of entropy calculating complexity, min-entropy or a lower bound has considerable advantages for dependent stochastic process, as only the probability in the worst case is involved. The methods for calculating a lower bound of entropy for oscillator-based TRNG are presented in [9, 2], and the worst case is also investigated in [1].

The calculation expression of the lower bound [9], which is denoted as H_{lo}, was presented in Equation (4):

$$H(B_i|B_{i-1}, \ldots, B_1) \geq H_{lo} = H(B_i|W_{i-1}) \approx \int_0^s H(R^{(s-u)} \bmod 2)P_W(\mathrm{d}u), \tag{4}$$

where B_i is the ith sampling bit and $R^{(s-u)}$ represents the number of crossing edges in the duration of $(s-u)$. The idea is inspired by the fact that W_i tells more information about B_{i+1} than all the previous bits. Following the similar idea, [2] also provides an analytical expression for H_{lo}, as shown in Equation (5).

$$H_{lo} = 1 - \frac{4}{\pi^2 \ln(2)}e^{-4\pi^2 Q} + O(e^{-6\pi^2 Q}) \tag{5}$$

2.4 Approximate Entropy

Approximate entropy (ApEn) is originally proposed to quantify the unpredictability of fluctuations in a time series. ApEn is a statistical value derived from the tested sequences. Note that, although the entropy of an TRNG shall be estimated from the stochastic model of a TRNG, but ApEn of the raw bits of a TRNG can also reflect the contained randomness. ApEn randomness test is also

adopted in the NIST statistical test suite [13], which compares the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a random sequence. The calculation process of ApEn for $\mathbf{b} = (b_1, \ldots, b_n)$ is presented in Algorithm 1. The block length m in Algorithm 1 has an important impact on ApEn calculation, which is treated as a trade-off. The larger value of m improves the accuracy of entropy estimation, but meanwhile significantly increase the computation complexity and the required length of the tested bit sequence.

Algorithm 1 Approximate entropy calculation [13]

Input: block length m, bit sequence $\mathbf{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$
Output: $ApEn$

1: Augment the n-bit sequence to create n overlapping m-bit sequences by appending $m-1$ bits from the beginning of the sequence to the end of the sequence.
2: Make a frequency count of the n overlapping blocks. The count is represented as $\#i$, where i is the m-bit value.
3: Compute $C_i^m = \#i/n$ for each value of i.
4: Compute $\delta_m = \Sigma_{i=0}^{2^m-1} C_i^m \log_2 C_i^m$.
5: Replace m by $m+1$ and repeat Steps 1-4.
6: Compute $ApEn = \delta_{m+1} - \delta_m$.
7: **return** $ApEn$

3 Our Evaluation Method

In this section, we provide three estimation methods for the entropy: phase-based, time-based and ApEn. The former two utilize the jitter parameters to perform the estimation in theory, while the latter analyzes the output sequences.

3.1 Bit-rate Entropy Calculation

In practice, the concept of entropy per bit is preferred for entropy evaluation, rather than the n-bit entropy. As the unit of the lower bound and ApEn is one bit, it is necessary to transfer n-bit entropy to entropy per bit, which is called bit-rate entropy. The bit-rate entropy is closely related to the expected workload that is necessary to guess (sufficiently long) sequences of random bits [7]. In addition, a precise Shannon entropy expression, which contains more parameters, allows the TRNG designers to optimize their structures and specifically adjust the parameters to get more entropy.

The bit-rate entropy H should be calculated from infinitely long random sequences, as Equation (6) shows. As n shall be infinity, the calculation of H is nearly infeasible in either statistical or iterative computation. One way is to figure out reliable H is to deduce the precise expression of H_n in terms of n.

Another possible case is that n actually can be a finite value, rather than being asymptotically infinite.

$$H = \lim_{n\to\infty} \frac{H_n}{n} = \lim_{n\to\infty} H(B_n|B_{n-1}, ..., B_1) \tag{6}$$

Time-based method. In the aspect of time evolution, it is observed that the correlation between two adjacent sampling bits is decreasing with the increase of the sampling interval. When the sampling interval is sufficient long, the sampling bits can be treated as independent. Here we provide a method to determine the required sampling interval for independent sampling bits.

The correlation coefficient of adjacent bits B_i and B_{i+1} is represented as:

$$\mathrm{cor}(B_i, B_{i+1}) = \frac{\mathrm{COV}(B_i, B_{i+1})}{\sqrt{\mathrm{Var}(B_i)\mathrm{Var}(B_{i+1})}},$$

where $\mathrm{COV}(\cdot)$ is the covariance function, and $\mathrm{Var}(\cdot)$ represents the variance. Then, using the stationary property [9] that $\mathrm{Prob}(B_i = 1) = \mathrm{Prob}(B_{i+1} = 1)$, the correlation coefficient is deduced as:

$$\mathrm{cor}(B_i, B_{i+1}) = \frac{\mathrm{Prob}(B_i = 1, B_{i+1} = 1) - \mathrm{Prob}(B_i = 1)^2}{\mathrm{Prob}(B_i = 1) \cdot \mathrm{Prob}(B_i = 0)}.$$

For different values of Q, we compute the correlation coefficients, as shown in Figure 2. We observe that the dependence oscillatingly decreases with the increasing of Q. The absolute value of the coefficient drops below 10^{-3} when Q is larger than 0.16, where we consider that the correlation can be ignored and the adjacent sampling bits are treated as independent.

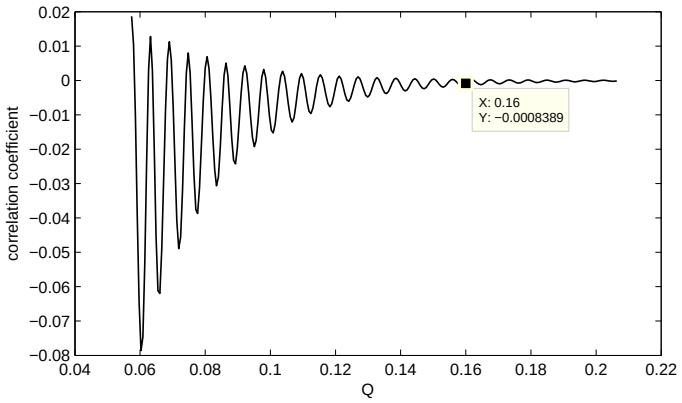

Fig. 2. The correlation coefficient in terms of Q

Using this conclusion, we further determine the longest timing distance, within which all the sampling behaviors are dependent. That is to say, when the distance between two sampling bits is longer than that distance, the two bits are

treated to be independent. We denote l as the correlation length, which means the $(i+l)$th sampling bit B_{i+l} is only dependent on the previous l bits. Given a value of Q for a oscillator-based TRNG, the correlation length is deduced as $l = \lfloor (\frac{Q_{ind}}{Q}) \rfloor$, where Q_{ind} is the required Q value for the independence, and Q_{ind} is set to 0.16 in this paper. Then, combining with the additional conclusion that the sampling process is stationary [9], H can be derived as

$$H = \lim_{n \to \infty} H(B_n | B_{n-1}, ..., B_1) = H(B_{l+1} | B_l, ..., B_1). \tag{7}$$

A lower threshold of the coefficient certainly is helpful for getting a more reliable result, but the derived correlation length may be too large to complete the iterating computation of entropy within an acceptable time. The maximum l in our computation is limited below 15. For $Q_{ind} = 0.16$, setting $l = 15$ means reliable values can be acquired for $Q > 0.0107$.

Phase-based method. In [2], since an analytical expression of n-bit entropy exists, for n is approaching infinity, the approximated bit-rate entropy is expressed as Equation (8).

$$H \approx 1 - \frac{32}{\pi^4 \ln(2)} \cos^2(2\pi\nu) B^2 \tag{8}$$

Note that using this equation to calculate bit-rate entropy is tentative, since H_n in Equation (3) is not provably uniform in n [2]. The problem of non-uniformness in n dose not exist in the time-based method, because the parameter l has been chosen before calculating Equation (7). In the following sections, we will learn that Equation (8) is applicable under some parameters, but has non-ignorable errors under other parameters. Hence, in the next section, we improve the equation by performing further expansion of original expression, and validate the effectiveness of the improvement by comparing with the time-based method and the ApEn of simulated sequences.

3.2 Approximate Entropy for Short-Term Dependent Bits

ApEn is a statistical result to estimate the entropy of the tested sequence. An important parameter in the algorithm is the block length m, which partially determine the estimation accuracy of the algorithm. The ideal case is that the tested bits are independent beyond the bit interval of m, which means the estimation algorithm can have a comprehensive overlook on the tested sequence. Fortunately, for the output of oscillator-based TRNGs, the correlation lags are limited due to the independence condition, hence the sampling bits only have short-term dependence.

In the statistical method, we first use the autocorrelation test to find out the correlation length in the sequence, and set m as the length to calculate ApEn. The autocorrelation test is based on the autocorrelation plot [3], which is a commonly-used tool for checking randomness in a data set. Here, we do not adopt the autocorrelation test in [12] for random bits, because the basis of that test is the uniformity of the tested sequence. Otherwise (the uniformity is not

satisfied), a higher correlation value will be acquired and autocorrelation test is failed. Hence, we return to the original test approach that only focuses on the correlation. The autocorrelation coefficient is represented as $R_h = C_h/C_0$, and C_h is the autocovariance function: $C_h = \frac{1}{n}\sum_{t=1}^{n-h}(b_t - \bar{b})(b_{t+h} - \bar{b})$, where $\bar{b}$ is the mean of $b_1, ..., b_n$, and C_0 is the variance function: $C_0 = \frac{1}{n}\sum_{t=1}^{n}(b_t - \bar{b})^2$.

For randomness tests, it is recommended to use 99% confidence band to justify whether the test is passed or not. In this case, the test is passed when C_h lies in the interval $[-z_{1-\alpha/2}/\sqrt{n}, z_{1-\alpha/2}/\sqrt{n}]$, where the significance level $\alpha = 0.01$ and z is the cumulative distribution function of the standard normal distribution. Therefore, for the calculation of approximate entropy for short-term dependent bits, we provide the following statistical method on the oscillator-based TRNG output, as shown in Algorithm 2. Note that, due to the Type-I error in the hypothesis test, the intrinsic independent sequences still has the probability of α to fail the test. However, this fact, which increases the correlation length m, would not lead to estimation error of the entropy as long as the sequence length is satisfied, since larger m is preferred for estimation.

Algorithm 2 Approximate entropy calculation for short-term dependent bits

Input: $h = 1$, bit sequence $\mathbf{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$
Output: *ApEn*
1: **while** $|C_h| > z_{0.995}/\sqrt{n}$ **do**
2: $h = h + 1$
3: **end while**
4: Compute *ApEn* using Algorithm 1 with the parameter $m = h$
5: **return** *ApEn*

4 Entropy Evaluation

In this section, by comparing the results of different entropy calculation methods, we evaluate the applicability and accuracy of these methods for oscillator-based TRNGs. Particularly, as the original analytical formula has biases on the bit-rata entropy estimation for some TRNG parameters, we present a more accurate formula by performing further deducing, and the correctness is verified with other entropy results. Finally, we investigate the limitations of these methods in the case of tiny randomness, i.e., very small Q.

4.1 Bit-rate Entropy Calculation Results

We use the proposed time-based and phase-based methods to calculate bit-rate entropy, and the results in terms of Q and r are shown in Figure 3. However, we find that the approximated bit-rate entropy derived from Equation (8) is not consistent with that calculated by Equation (7). The inconsistency has been preliminarily pointed out in [10]. Note that the entropy at $r = 1 - x$ is identical to that at $r = x$, where $x \in (0, 0.5]$, thus we only present the cases for r ranging

from 0 to 0.5. More precisely, the difference between the two results is maximized with the parameter r approaching 0.5, as shown in Figure 3. Their results are almost identical in the worse cases ($r \in [0, 0.2]$), but in the other cases of r with a modest Q value the deviation occurs, especially at $r = 0.5$.

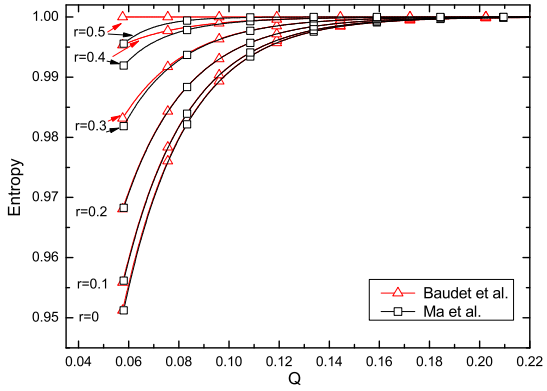

Fig. 3. The bit-rate entropy calculated from Ma *et al.*'s (time-based) and Baudet *et al.*'s (phase-based) methods

From the physical perspective, the r value is related to the fractional part of the ratio of sampling interval to the mean of half-periods ($\Delta t/m_X$). From the theoretical result of [10], to achieve a sufficient bit-rate entropy (such as 0.9999), the required sampling frequency in the best case is about two times faster than that in the worst case under the same quality factor. Hence, in the condition of fixed Q, the value of r has a non-negligible impact on the entropy, as shown in Figure 3. Also, from the perspective of the designer, adjusting r can significantly improve the entropy without the degradation of the sampling frequency.

4.2 Improved Bit-Rate Entropy Expression Formula

We expand the original approximated expression formula of n-bit entropy by performing further deducing. The improved results are presented in Theorem 1.

Theorem 1. *For* $r = \frac{\Delta t}{m_X} \mod 1$ *and* $Q = \frac{s_X^2 \Delta t}{4m_X^3}$, *the n-bit entropy is:*

$$\begin{aligned} H_n \doteq\ & n - \frac{32}{\ln(2)\pi^4}\cos^2(\pi r)(n-1)e^{-4\pi^2 Q} \\ & - \frac{32}{\ln(2)\pi^4}\Big[\cos^4(\pi r)\big(1.524n - 0.092\big) - 2.379\cos^2(\pi r)(n-2) + (n-2)\Big]e^{-8\pi^2 Q} \\ & + O(e^{-10\pi^2 Q}). \end{aligned} \tag{9}$$

The (trial) approximated bit-rate entropy is expressed as:

$$H \approx 1 - \frac{32B^2}{\ln(2)\pi^4}\cos^2(\pi r) - \frac{32B^4}{\ln(2)\pi^4}\Big[1.524\cos^4(\pi r) - 2.379\cos^2(\pi r) + 1\Big]. \tag{10}$$

In the improved expression Equation (9), the first two terms are derived from the original one. Our work focuses on the deduction of the third term, the higher-order term. We strictly follow the same assumptions used in [2], but perform the further deduction on the entropy calculation process based on series expansion. The proof details are presented in the full version of this paper.

4.3 Bit-Rate Entropy Comparison: Time-based vs. Phase-based

In order to validate the reliability of the improved result, we first compare it with the bit-rate entropy derived from the time-based method. The comparison result is shown in Figure 4. We can see that after our improvement the two results become very close in all six cases from $r = 0$ to $r = 0.5$. Note that as the expression is analytical, the derived entropy is not only the data lying in the six curves, but the values for all the possible cases of Q and r. We remark that it is no surprise that the two entropy results are identical, because the focusing target and the physical assumption of small jitter are both the same. The equivalence between the two models has been presented in [2].

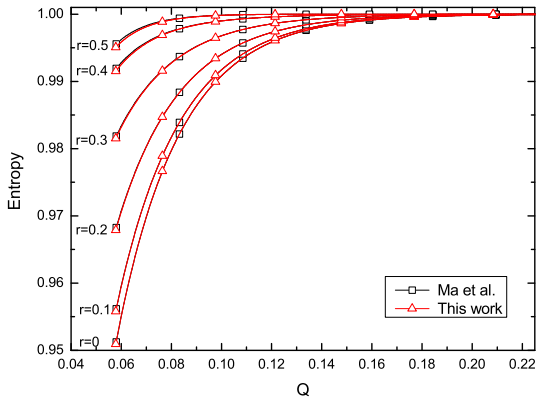

Fig. 4. The comparison of bit-rate entropy with the improved formula

Furthermore, from Equation (10) we also explain why the original expression has a significant estimation error when r is large. When the coefficients of B^2, B^4,... ($0 < B = e^{-2\pi^2 Q} < 1$) are comparable, the subsequent terms after B^2 can be ignored for large Q and the estimation error is acceptable. However, with r increasing from 0 to 0.5, the coefficient of B^2 decreases from maximum to 0, while the impact of B^4 increases. Especially, when r approaches 0.5, the coefficient of B^2 approaches 0, while the B^4 term does not become zero due to

the existence of the constant 1 in the coefficient of B^4. Therefore, in this case the impact of B^4 term cannot be ignored and only using B^2 term to estimate the entropy is not enough.

Another important observation is that expanding the Taylor series to B^4 is enough to reliably estimate the entropy for such $Q \in (0.06, +\infty)$, where Q is not a very small value. The improved expression might have a bias when Q becomes a much smaller value, as the impact of the higher-order term of B (such as B^6) exists. But we must admit that getting the higher-order term of B seems infeasible, as the series in Equation (3) after further expanding are too complex.

4.4 Bit-Rate Entropy vs. Approximate Entropy

After the improvement, the bit-rate entropy values derived from the two methods are consistent, but it is necessary to confirm that the theoretical results is consistent with the experimental. For this purpose, we use the approximate entropy, which is a statistical measurement from the output bit sequence, to verify the applicability of the entropy evaluation method. Note that, the statistical entropy values are also random for random sequences, so directly using ApEn to do entropy estimation would lead to measurement errors. However, it is valuable to compare the trends of ApEn and bit rate entropy, which can be treated as experimental and theoretical results, respectively.

Following the assumptions in the aspect of time evolution, we perform a simulation experiment to calculate ApEn. In the experiment, the fast signal periods are independent and identically distributed, and the distribution is set as the normal distribution $N(1, 0.01^2)$. Each ApEn is computed from 10^5 sampling bits for each sampling interval which corresponds the values of Q and r. As the two bit-rate entropy results are almost the same, we use the improved phase-based result as the reference to compare with ApEn. The comparison results from $r = 0$ to $r = 0.5$ are shown in Figure 5. We find that the two sets of results are well-matched for all r values. Therefore, Algorithm 2 is suitable to estimate the bit-rate entropy for this type of short-term dependent sequences. A more precise results can be acquired by averaging the estimated values of many statistical experiments.

4.5 Entropy Estimation for Smaller Quality Factor

In the previous entropy estimation results, the investigated values of Q are not very small, which are available for the entropy evaluation of most practical TRNGs. However, for very samll Q values, the presented entropy calculation methods are not applicable. The reasons are explained as follows.

- For the time-based bit-rate entropy calculation method, a very small Q means that the dependent length l is very large. For example, when $Q = 0.005$, l equals to 32, meaning that the traversal space should be 2^{32}, which is infeasible for computation. In this case, the estimation would be larger than the real entropy value, i.e., the overestimation occurs.

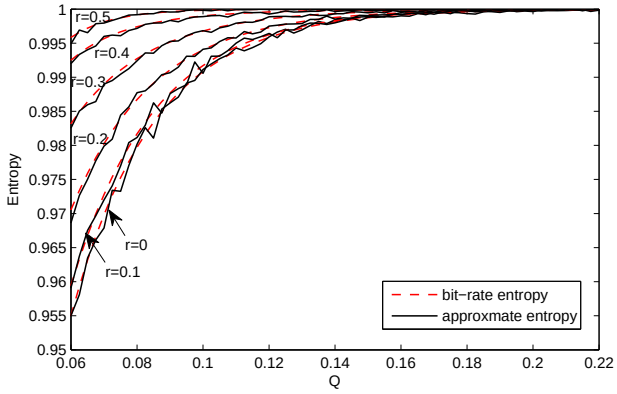

Fig. 5. The comparison between the bit-rate entropy and the approximate entropy

– For the phase-based bit-rate entropy calculation method, when Q decreases, the estimation error increases with no limitation, as the H_n expression is not uniform in n. Therefore, the approximated formula is not applicable to estimate the bit-rate entropy in this case, though our improvement has extended the applicable range of the formula.

– For the presented approximate entropy estimation method, a very small Q makes the statistical correlation lasts very long lags, which causes that the parameter m in Algorithm 2 is too large to complete the computation. For example, in our experiment, when $Q = 0.01$ the statistical m of Algorithm 2 is about 30, thus the workload for the traversal loops and the requirement for the sequence length are unacceptable in this case. The problem also leads to an overestimation for the entropy of the tested sequence.

Actually, as we mentioned, the lower bound expression formula has be presented in [2]. As Equation (5) shows, the expression formula of the lower bound also contains a term of O. Using this approximated expression also causes overestimation of the entropy when Q is smaller than 0.01. As shown in Figure 6, the approximated lower bound becomes larger than the bit-rate entropy derived from time-based methods with the worst case of $r = 0$, though the bit-rate entropy might have been overestimated. However, we emphasize that the computation of this O term in Equation (5) is feasible since the traverse of 2^n states is nonexistent. Therefore, we present the calculation results for the precise lower bound of entropy for smaller Q values, as labeled in Figure 6. The comparison result indicates that the precise expression of the lower bound eliminates the overestimation of entropy for very small Q values.

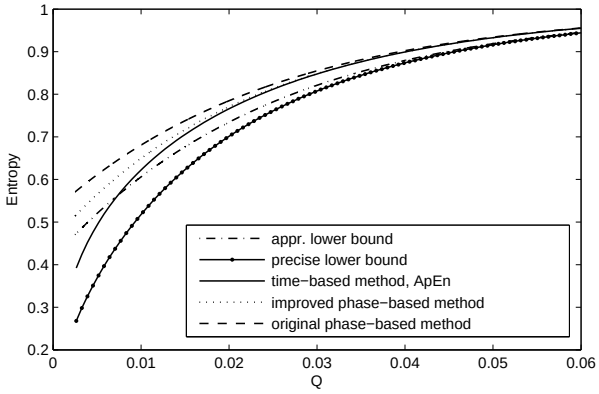

Fig. 6. The comparison of entropy values with small Q at $r = 0$

5 On the Relationship with Physical RNGs

The existing models [2, 10] assume that the oscillating period or phase increment is independently distributed due to the influence of white noise. This is a common assumption in literature, which allows to guarantee the simplicity of the model. However, in real TRNG circuits, the jitter or phase is also influenced by colored noises (such as $1/f$ noise) more than white noise, and the phenomenon has been demonstrated in recent works [5, 10, 14]. Under these colored noises, the period jitter has long-term dependence, and the dependence is also inherited by the sampling bit sequence [10]. In practical TRNGs, it is infeasible to perform similar confirmatory experiment as our simulation where the entropy is calculated via the output sequence, as the randomness amount is inevitably increased by colored noises and the offset r is hard to be precisely measured.

Fortunately, the white noise is independent from colored noises in principle, so the existing model and corresponding entropy estimation methods can still work for estimating the contribution of the white noise. When the estimated contribution (i.e., entropy) derived from *independent jitter* is sufficient, we can also claim that the entropy of the TRNG is satisfied. In practical entropy evaluation, the independent jitter can be acquired by employing an inner measurement method that excludes the dependent component of the jitter in the measurement (such as [5]). This evaluation approach neatly sidesteps the impact of colored noises.

We perform the hardware experiment on an FPGA (Field Programmable Gate Array) platform (Xilinx XC5LX110T), where two ring oscillators are implemented using Look-Up Tables (LUTs) with the close frequency of 280.5 MHz. The sampling interval is set as the period number of one oscillating signal, and the counting period number of the other signal is treated as the random variable, thus the random bit is the LSB of the counting number. Here, we do not use the number of half-periods to eliminate the impact of the imbalance of the

duty cycle, and the change is compatible with the above models. The period number of the sampling signal is set to $256 \times i$, where $i \in \{20, 21, ..., 40\}$. For each sampling interval, we collect the random number sequence with length 2^{20}, and calculate the ApEn of the bit sequence. Particularly, the quality factor that is influenced by white noises Q_W is computed by employing the method of [5]. The comparison between the ApEn and the theoretical entropy (worst case and best case) is depicted in Figure 7. It is observed that ApEn increases between the worst and the best case of theoretical entropy as expected. As the bit sequence has been affected by colored noises, its statistical randomness is much better than the worst case of the theoretical entropy. From Figure 7, we can conclude that our improved theoretical entropy is suitable to estimate the lower and upper bounds of the output bit sequence.

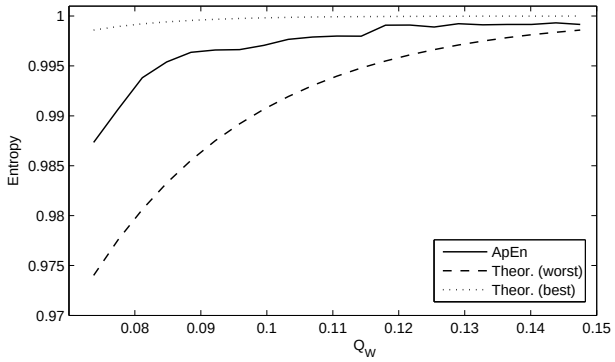

Fig. 7. The comparison between ApEn and theoretical entropy in the physical RNG

6 Conclusion

Entropy estimation is essential for TRNG security testing, and a reliable result of entropy estimation is preferred for both designers and verifiers. In this paper, we investigate the applicability the different entropy calculation methods for oscillator-based TRNGs, including bit-rate entropy, the lower bound and approximate entropy. In the evaluation, we present two effective methods for bit-rate entropy calculation in theory, and design a specific method for the approximate entropy. The evaluation results indicate that the theoretical estimation results are consistent with the experimental measurements, thus the presented methods are reliable for not small Q values. The mutual verifications among these estimation methods make us believe that the calculated results are reliable. Furthermore, for the case with very small quality factor, the existing entropy estimation methods are inapplicable, thus we recommend to use the precise lower bound as a conservative estimation. In the hardware experiment, we validate that the ApEn still lies in the interval between the worst and best case of the theoretical entropy, though the bit sequence is effected by colored noises.

Acknowledgments.

This work was partially supported by National Basic Research Program of China (973 Program No. 2013CB338001), National Natural Science Foundation of China (No. 61602476, No. 61402470) and Strategy Pilot Project of Chinese Academy of Sciences (No. XDA06010702).

References

1. Amaki, T., Hashimoto, M., Mitsuyama, Y., Onoye, T.: A worst-case-aware design methodology for noise-tolerant oscillator-based true random number generator with stochastic behavior modeling. IEEE Transactions on Information Forensics and Security 8(8), 1331–1342 (2013)
2. Baudet, M., Lubicz, D., Micolod, J., Tassiaux, A.: On the security of oscillator-based random number generators. Journal of Cryptology 24(2), 398–425 (2011)
3. Box, G.E.P., Jenkins, G.: Time Series Analysis: Forecasting and Control, pp. 28–32. Holden-Day (1976)
4. Fischer, V., Lubicz, D.: Embedded evaluation of randomness in oscillator based elementary TRNG. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems (CHES). pp. 527–543 (2014)
5. Haddad, P., Teglia, Y., Bernard, F., Fischer, V.: On the assumption of mutual independence of jitter realizations in P-TRNG stochastic models. In: IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE). pp. 1–6 (2014)
6. Information Technology Laboratory: FIPS 140-2: Security Requirement For Cryptographic Modules (2011)
7. ISO/IEC 18031: Information Technology - Security Techniques - Random bit generation (2011)
8. Killmann, W., Schindler, W.: A proposal for functionality classes for random number generators. `http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile` (2011)
9. Killmann, W., Schindler, W.: A design for a physical RNG with robust entropy estimators. In: Oswald, E., Rohatgi, P. (eds.) Cryptographic Hardware and Embedded Systems (CHES). pp. 146–163 (2008)
10. Ma, Y., Lin, J., Chen, T., Xu, C., Liu, Z., Jing, J.: Entropy evaluation for oscillator-based true random number generators. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems (CHES). pp. 544–561 (2014)
11. Marsaglia, G.: Diehard Battery of Tests of Randomness. `http://www.stat.fsu.edu/pub/diehard/`
12. Menezes, A., Oorschot, P.v., Vanstone, S.: Handbook of Applied Cryptography. CRC Press (1997)
13. Rukhin, A., et al.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800–22. `http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf`
14. Valtchanov, B., Fischer, V., Aubert, A., Bernard, F.: Characterization of randomness sources in ring oscillator-based true random number generators in FPGAs. In: DDECS. pp. 48–53 (2010)