Cryptography and Communications
https://doi.org/10.1007/s12095-025-00778-7

RESEARCH

Observations on NIST SP 800-90B entropy estimators

Melis Aslan[1] · **Ali Doğanaksoy**[1] · **Zülfükar Saygı**[2] · **Meltem Sönmez Turan**[3] · **Fatih Sulak**[4]

Received: 29 September 2024 / Accepted: 15 January 2025

Abstract
Random numbers play a crucial role in cryptography since the security of cryptographic protocols relies on the assumption of the availability of uniformly distributed and unpredictable random numbers to generate secret keys, nonce, salt, etc. However, real-world random number generators sometimes fail and produce outputs with low entropy, leading to security vulnerabilities. The NIST Special Publication (SP) 800-90 series provides guidelines and recommendations for generating random numbers for cryptographic applications and describes 10 black-box entropy estimation methods. This paper evaluates the effectiveness and limitations of the SP 800-90 methods by exploring the accuracy of these estimators using simulated random numbers with known entropy, investigating the correlation between entropy estimates, and studying the impacts of deterministic transformations on the estimators.

Keywords Cryptography · Entropy estimation · Min-entropy · Randomness

Mathematics Subject Classification (2010) 94A60 · 94A17

All authors contributed equally to this work.

✉ Melis Aslan
melisa@metu.edu.tr

Ali Doğanaksoy
aldoks@metu.edu.tr

Zülfükar Saygı
zsaygi@etu.edu.tr

Meltem Sönmez Turan
meltem.turan@nist.gov

Fatih Sulak
fatih.sulak@atilim.edu.tr

1 Department of Mathematics, Middle East Technical University, 06800 Ankara, Türkiye

2 Department of Mathematics, TOBB ETU, 06560 Ankara, Türkiye

3 Computer Security Division, National Institute of Standards and Technology, 20899 Gaithersburg, MD, USA

4 Department of Mathematics, Atilim University, 06830 Ankara, Türkiye

1 Introduction

Random numbers are widely used in cryptographic protocols to generate secret keys, initialization vectors, nonces, salts, etc. The security of these protocols relies on the assumption that these numbers are generated uniformly at random and are unpredictable. However, real-world random number generators sometimes fail and produce outputs with low entropy, leading to security vulnerabilities [1, 2].

A variety of organizations have developed standards and guidelines on generating random numbers that are suitable for cryptographic applications, such as the National Institute of Standards of Technology (NIST) [3–6], the International Organization for Standardization (ISO) [7–10], and Bundesamt für Sicherheit in der Informationstechnik (BSI) [11–13].

Cryptographic random number generators are typically composed of multiple components, including (i) a *noise source* that extracts randomness from physical phenomena (e.g., thermal noise, mouse movements, radioactive decay, free-running oscillator) to generate a *seed* and (ii) a *pseudorandom number generator* (PRNG) (also known as a *deterministic random bit generator*) that extends the seed to generate a long random-looking sequence. Since PRNGs are deterministic, the entropy is solely provided by the noise source, and it is important to measure the unpredictability of the noise source outputs.

Designing random number generators for cryptographic use has many challenges, including finding a robust *noise source* to extract randomness and the difficulty of determining how unpredictable the outputs are (i.e., estimating its entropy).

Various statistical randomness tests can be applied to measure the quality of the random numbers. The most commonly used statistical randomness suites are TestU01 [14], DIEHARD [15], DIEHARDER [16], and NIST Special Publication (SP) 800-22 Rev.1 [17]. These tests may not be suitable for assessing noise source outputs, as they typically have strong biases and would fail these tests.

The unpredictability of noise source outputs is measured using *entropy*, and two commonly used measures of entropy are *Shannon entropy* and *min-entropy*. *Min-entropy* is a more conservative measure, which is based on the probability of guessing the most likely output of a randomness source.

Estimating the entropy of noise source outputs is challenging because the distribution of the output values is generally unknown. The BSI standards require stochastic modeling of the noise source to specify a family of probability distributions to estimate entropy. Since stochastic modeling may not be possible or practical due to the diversity and complexity of the random number generators, NIST standards allow using black-box statistical methods for entropy estimation.

SP 800-90B [4] describes ten entropy estimators: most common value, collision, Markov, compression, t-tuple, longest repeated substring (LRS), multi most common in window prediction, lag prediction, multiple Markov Model with Counting (multiMMC) prediction, and LZ78Y. The minimum of these ten estimates is used to estimate the min-entropy of the noise source outputs.

Related work Zhu et al. [18] showed that the collision and compression estimates provide significant underestimates and proposed a new estimator that achieves better accuracy for min-entropy. Kim et al. [19] also showed that the compression estimate underestimates min-entropy and proposed two kinds of min-entropy estimators to improve computational complexity and estimation accuracy by leveraging two variations of Maurer's test. Hill [20] demonstrated that the collision and compression estimators incorrectly use the central limit theorem. Hill [20] also claimed that the Markov estimator should not be directly compared to other estimators since it does not use confidence intervals during estimation. Additionally, Turan et al. [21] provided a correlation and sensitivity analysis of statistical randomness tests.

Contributions This paper evaluates the accuracy, effectiveness, and limitations of the SP 800-90B estimators using simulated random numbers with known entropy, investigates the correlation between entropy estimates, and studies the impacts of deterministic transformations on the estimators.

Our study indicates that both compression and collision estimates tend to underestimate entropy for both uniform and biased distributions, aligning with earlier results. On the other hand, LRS and lag prediction overestimate entropy for biased distributions.

Our experiments reveal a strong correlation between the Markov and MCV tests for uniform distributions. For biased datasets that meet the IID assumption, we observe increased correlations among several estimators, particularly MultiMCW, MultiMMC, and LZ78Y. MCV also shows high correlation with multiple estimators, including Markov, Compression, MultiMCW, and MultiMMC. Conversely, for biased datasets that do not meet the IID assumption, only moderate correlations are noted between pairs such as (Markov, MCV) and (LZ78Y, Markov).

Lastly, studies on the impacts of deterministic transformations show that binary derivation significantly affects entropy estimates, particularly for prediction-based estimators.

Organization Section 2 provides preliminaries on SP 800-90B entropy estimation and overviews of two correlation metrics. Section 3 describes the paper's methodology. Section 4 presents experimental results and Section 5 provides discussion. The appendix 5 contains various statistical data and graphs related to the experimental results.

2 Preliminaries

2.1 Min-Entropy

In information theory, entropy is a measure of uncertainty associated with the outcomes of a random variable. There are different measures of entropy, and NIST SP 800-90B [4] uses *min-entropy*, which is a conservative entropy measurement based on the probability of guessing the most likely output of a randomness source.

Definition 1 *Let $\mathcal{X}$ be a random variable that takes values from the set $A = \{x_1, x_2, \ldots, x_n\}$ with probabilities $Pr(\mathcal{X} = x_i) = p_i$ for $i = 1, 2, \ldots, n$. The min-entropy of the random variable $\mathcal{X}$ is defined as*

$$
\begin{aligned}
H_\infty &= \min_{1 \le i \le n} (-\log_2 p_i) \\
&= -\log_2 (\max_{1 \le i \le n} p_i).
\end{aligned}
$$

The random variable $\mathcal{X}$ is said to have min-entropy h if the probability of observing any particular value for $\mathcal{X}$ is at most 2^{-h}. When the random variable has a uniform probability distribution (i.e., $p_1 = p_2 = \cdots = p_n = 1/n$), the variable has the maximum possible value for the min-entropy, which is $\log_2 n$.

In this paper, the term *entropy* specifically refers to *min-entropy*.

2.2 Entropy estimation based on SP 800-90B

SP 800-90B [4] describes an *entropy source* model, composed of a noise source, health tests, and an optional conditioning function. The standard also provides guidelines for generating random numbers using entropy sources and specifies entropy estimation techniques to ensure the randomness and unpredictability of the outputs. These black-box techniques are applied to noise source outputs and are independent of the internals of the noise source.

SP 800-90B [4] defines two tracks to estimate the min-entropy of an entropy source: independent and identically distributed (IID) and non-IID. To determine which track to use, several statistical tests are applied to an output sequence generated by the entropy source to check the IID assumption. If the output sequence passes these tests, the source is assumed to generate IID outputs, and only the most common value method is used to estimate the entropy. Otherwise, the source is assumed to generate non-IID outputs, and the minimum of the 10 SP 800-90B estimators is used to estimate the entropy of the source. Table 1 lists the estimators and corresponding metrics provided in the standard. Except for collision, Markov, and compression, the estimators provide support for non-binary noise source outputs.

The estimators take noise source outputs $S = (s_1, s_2, \ldots, s_L)$, where $s_i \in A = \{x_1, x_2, \ldots, x_n\}$ and return an min-entropy estimate between 0 and $\log_2 n$. Some of the estimators, namely collision, Markov and compression, are only defined for binary inputs (i.e., $n = 2$). Note that to establish the final entropy estimate, the standard additionally considers the entropy estimate from the designers and the impact of the conditioning components, etc. This study focuses on the black-box estimators, and the additional considerations, including IID testing, are outside the scope of this study.

Table 1 Entropy estimators of NIST SP 800-90B

Estimator	***Metric***	Support for $n > 2$?
E1: Most Common Value	Proportion of the most common value in the input data set	✓
E2: Collision	Probability of the most-likely output, depending on the number of collisions	×
E3: Markov	Dependencies between consecutive values	×
E4: Compression	Compression amount of the input dataset	×
E5: t-Tuple	Frequency of t-tuples	✓
E6: Longest Repeated Substring (LRS)	Number of repeated substrings	✓
E7: Multi Most Common in Window Prediction	Number of correct predictions based on the most common value	✓
E8: Lag Prediction	Number of correct predictions based on periodicity	✓
E9: MultiMMC Prediction	Number of correct predictions based on multiple Markov models	✓
E10: LZ78Y Prediction	Number of correct predictions based on a dictionary constructed using observed tuples	✓

The estimators take noise source outputs $S = (s_1, s_2, \ldots, s_L)$, where $s_i \in A = \{x_1, x_2, \ldots, x_n\}$, and return a min-entropy estimate between 0 and $\log_2 n$. The collision, Markov, and compression estimators are only defined for binary inputs (i.e., $n = 2$). To establish the final entropy estimate, the standard considers the entropy estimate from the designers and the impact of the conditioning components. This study focuses on the black-box estimators, and the additional considerations — including IID testing — are outside of the scope of this study.

2.3 Correlation analysis

The Pearson [22] and Spearman [23] correlation coefficients are commonly used metrics to measure the correlation between two random variables. The correlation coefficients take values between −1 and 1. A value close to 1 or −1 shows a strong positive or negative association between variables, whereas a value close to 0 shows a weak association. The Pearson correlation [22] measures the strength of a linear relationship between two random variables, assuming that the variables are distributed normally, whereas the Spearman correlation [23] describes the monotonic relationship between variables without the assumption that the variables have a normal distribution. See Table 2 for the interpretation of the Pearson r and Spearman correlation coefficients ρ.

Definition 2 *Let $\mathcal{X}$ and $\mathcal{Y}$ be random variables. The Pearson correlation coefficient r between a given paired dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is defined as*

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

where n is the sample size, x_i and y_i are sample points, $\bar{x}$ is the sample mean of $\mathcal{X}$, and $\bar{y}$ is the sample mean of $\mathcal{Y}$.

Definition 3 *Let $\mathcal{X}$ and $\mathcal{Y}$ be random variables. The Spearman correlation coefficient ρ between a given paired dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is defined as*

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)},$$

where n is the sample size, and d_i is the difference between the rank of the paired samples.

A positive correlation in either method indicates that as one variable increases, the other also increases, with Pearson requiring proportionality (linear growth) and Spearman only requiring consistent growth (rank-based). Conversely, a negative correlation means that as one variable increases, the other decreases, with Pearson emphasizing linearity and Spearman

Table 2 Interpretation of Pearson r and Spearman ρ correlation coefficients

Interval	**Interpretation**
$0 < \lvert r\rvert, \lvert\rho\rvert \leq 0.20$	Negligible correlation
$0.2 < \lvert r\rvert, \lvert\rho\rvert \leq 0.40$	Weak correlation
$0.4 < \lvert r\rvert, \lvert\rho\rvert \leq 0.60$	Moderate correlation
$0.6 < \lvert r\rvert, \lvert\rho\rvert \leq 0.80$	High correlation
$0.8 < \lvert r\rvert, \lvert\rho\rvert \leq 1$	Strong correlation

focusing on consistent decline. Pearson is sensitive to outliers, while Spearman is more robust and suitable for non-linear but monotonic trends. Considering the absolute value of correlation coefficients is meaningful when the focus is on the strength of the relationship, regardless of direction.

For necessary cases, to control the false discovery rate, the Benjamini-Hochberg procedure [24] was applied to interpret the results. We had multiple hypotheses regarding the correlations between the tests. Therefore, we adjusted the P-values using Benjamini-Hochberg procedure in order to reduce the false positive outcomes.

3 Methodology

The goal of this study is to answer the following questions regarding the entropy estimators introduced in SP 800-90B [4]:

1. *How closely do the entropy estimators match the true entropy of the source?*
2. *How correlated are the entropy estimators?*
3. *How do different deterministic transformations impact the entropy estimate?*

3.1 Entropy estimation using known distributions

One approach to understanding the accuracy of the entropy estimators is to simulate various sequences with known probability distributions (hence, known entropy) and check the difference between the estimated entropy and the true entropy. In cases where certain entropy estimators consistently yield outlier results compared to others, it is essential to investigate the underlying reasons for such discrepancies. This could involve examining the specific characteristics of the input data, inherent biases in the estimation techniques, or the impacts of using different input lengths and sample sizes.

3.2 Correlation of the entropy estimators

Understanding the correlation between different entropy estimators can provide insights into the reliability, robustness, and limitations of the estimators for cryptographic applications. One aspect to consider is the agreement between different entropy estimation methods by assessing whether they tend to produce similar entropy estimates for the same set of input sequences. This study employed correlation analysis to quantify the relationship between pairs of entropy estimates, using the Pearson and Spearman correlation coefficients.

3.3 Impact of deterministic transformations

The noise source outputs are typically processed using deterministic conditioning functions to reduce their statistical bias and improve their entropy rate (i.e., entropy per bit). The impacts of several deterministic transformations applied to the output sequence are of interest here.

Let $S = (s_1, s_2, \ldots, s_L)$ be a noise source output with length L, and let $S' = (s'_1, s'_2, \ldots, s'_L)$ be generated from S via a deterministic transformation. This study uses the following transformations:

- **Reverse:** This transformation generates a new sequence by changing the order of the sequence. The generated sequence $S' = (s_L, s_{L-1}, \ldots, s_2, s_1)$ is constructed with

$s'_i = s_{L-i+1}$ for each $i = 1, 2, \ldots, L$. For example, the reversed sequence of $S = (10110001110010)$ is $S' = (01001110001101)$.

- **Binary Derivative:** This transformation generates a new sequence by XORing (i.e., modulo 2 addition) the consecutive bits of the sequence. The generated sequence $S' = (s'_1, s'_2, \ldots, s'_L)$ is constructed with

$$s'_i = \begin{cases} s_i \oplus s_{i+1}, & i = 1, 2, \ldots, L-1, \\ s_1, & i = L. \end{cases}$$

  For example, the binary derivative of $S = (10110001110010)$ is $S' = (11010010010111)$.

- **t-Rotation:** This transformation applies a t-bit rotation to the input sequence, i.e., t-bit rotation of $S = (s_1, s_2, \ldots, s_L)$ is $S' = (s_{t+1}, s_{t+2}, \ldots, s_L, s_1, s_2, \ldots, s_t)$, where $t = 16, 64, 128$, or 1024. For example, 2-bit rotation of $S = (101100011100\ 10)$ is $S' = (11000111001010)$.

4 Experimental results

4.1 Accuracy of entropy estimators

The following datasets with known entropy were simulated for the experiments:

1. **Uniform distribution with full entropy.** The datasets are generated using the Cipher Block Chaining (CBC) mode of the block cipher Advanced Encryption Standard (AES) [25]. Sequences are generated for three different sample sizes (i.e., the size of the noise source output): binary, 4-bit, and 8-bit. For each sample size, 1000 sequences of length 1 000 000 were generated. In these sequences, all outputs are assumed to have an equal probability of occurring and are independent. Hence, the outputs have full entropy.
2. **Biased binary distribution with entropy=0.5.** The dataset follows a biased binary distribution, where the probability of observing a 0 is 0.7, and the probability of observing a 1 is 0.3. For each sample size, 1000 sequences of length 1 000 000 were generated. In these sequences, the expected entropy of a sequence is 0.5 per bit. This data is generated using the random number generator Mersenne Twister (MT19937) in C++.
3. **4-bit near-uniform with entropy=0.5.** This dataset follows a 4-bit near-uniform distribution, where the probability of observing the template 0000 is 0.25, and the probability of observing other 4-bit templates is 0.05. For each sample size, 1000 sequences of length 1 000 000 (bit) were generated. In these sequences, the expected entropy of a sequence is 0.5 per bit. This data is generated using the random number generator in C++.
4. **8-bit near-uniform with entropy=0.5.** This dataset follows an 8-bit near-uniform distribution, where the probability of observing the template 00000000 is 0.06, and the probability of observing other 8-bit templates is 0.003686. For each sample size, 1000 sequences of length 1 000 000 (bit) were generated. In these sequences, the expected entropy of a sequence is 0.5 per bit. This data is generated using the random number generator in C++.
5. **First-order Markov sequences with transition matrix** $P = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$. This dataset consists of binary sequences generated using a Markov process with the given transition matrix P. The sequences are constructed such that the transition probabilities between states are governed by $P(0 \to 0) = 0.7$, $P(0 \to 1) = 0.3$, $P(1 \to 0) = 0.3$, and $P(1 \to$

1) = 0.7. 1000 sequences of length 1 000 000 bits were generated. These sequences exhibit dependencies dictated by the transition matrix. The expected minimum entropy of a sequence is determined by the stationary distribution and transition probabilities, and it is computed as approximately 0.5155. The data is generated using a Markov process implemented in `C++`.

Table 3 compares the actual and estimated entropy values for binary, 4-bit, and 8-bit uniformly distributed data with full entropy. It shows that compression and collision estimates produce the smallest estimates for binary data, which is consistent with the findings of Zhu et al. [18] and Kim et al. [19]. Figure 1 in Appendix shows the distribution of the entropy estimation, and compression, and LRS estimators seem to show high variation compared to other estimators.

The same experiments were repeated for biased binary distribution, 4-bit near-uniform distribution, and 8-bit near-uniform distribution, and the results are summarized in Table 4. Similar to a uniform distribution, the compression estimate underestimates entropy for biased

Table 3 Mean and standard deviation of entropy estimators for binary, 4-bit, and 8-bit sources with full entropy

	1-bit Mean	**Std. D.**	**4-bit Mean**	**Mean/bit**	**Std. D.**	**8-bit Mean**	**Mean/bit**	**Std. D.**
E1	0.9951	0.0009	3.9514	0.9879	0.0056	7.6736	0.9592	0.0222
E2	0.9141	0.0194	*	*	*	*	*	*
E3	0.9982	0.0011	*	*	*	*	*	*
E4	0.8535	0.0287	*	*	*	*	*	*
E5	0.9294	0.0104	3.7799	0.9450	0.0149	7.6736	0.9592	0.0222
E6	0.9785	0.0262	3.8928	0.9732	0.1131	7.7468	0.9683	0.1878
E7	0.9954	0.0114	3.9635	0.9909	0.0662	7.8169	0.9771	0.1315
E8	0.9957	0.0072	3.9677	0.9919	0.0416	7.8116	0.9764	0.1679
E9	0.9951	0.0129	3.9616	0.9904	0.0778	7.8197	0.9775	0.1302
E10	0.9956	0.0096	3.9616	0.9904	0.0778	7.8198	0.9775	0.1302

Table 4 Mean and standard deviation of entropy estimators of datasets for biased binary, 4-bit near-uniform, and 8-bit near-uniform distributions

	Biased Binary		**4-bit Near-uniform**			**8-bit Near-uniform**		
	Mean	**Std. D.**	**Mean**	**Mean/bit**	**Std. D.**	**Mean**	**Mean/bit**	**Std. D.**
E1	0.5122	0.0009	1.9872	0.4968	0.0050	4.0169	0.5021	0.0160
E2	0.5095	0.0020	*	*	*	*	*	*
E3	0.5146	0.0011	*	*	*	*	*	*
E4	0.3224	0.0009	*	*	*	*	*	*
E5	0.5031	0.0116	1.9710	0.4928	0.0197	3.9993	0.4999	0.0380
E6	0.7692	0.0205	3.2364	0.8091	0.0954	6.9466	0.8683	0.1884
E7	0.5121	0.0055	1.9860	0.4965	0.0200	4.0063	0.5008	0.0738
E8	0.7756	0.0263	3.2812	0.8203	0.0923	6.9558	0.8695	0.2984
E9	0.5118	0.0055	1.9861	0.4965	0.0200	4.1557	0.5195	0.1028
E10	0.5118	0.0055	1.9860	0.4965	0.0200	4.1556	0.5194	0.1027

Table 5 Mean and standard deviation of entropy estimators of datasets for first-order Markov sequences

Estimator	Mean	Std. Dev.
Most Common Value	0.5397	0.0002
Collision	0.2726	0.0007
Markov	0.4601	0.0004
Compression	0.0976	0.0000
t-Tuple	0.0759	0.0026
Longest Repeated Substring (LRS)	0.1052	0.0036
Multi Most Common in Window Prediction	0.5397	0.0002
Lag Prediction	0.0549	0.0001
MultiMMC Prediction	0.0549	0.0001
LZ78Y Prediction	0.5397	0.0004

distributions. However, LRS and lag prediction overestimate the entropy by approximately 50%.

The (LRS) estimator calculates the collision entropy rather than the min-entropy by identifying the frequency of repeated substrings. As collision entropy serves as an upper limit for min-entropy, the LRS estimator naturally produces overestimated results, the results are consistent with the findings of [26]. Similar results were obtained for 4-bit and 8-bit samples.

The expected entropy value for Markov sequences is 0.5155. It is observed that most of the estimates presented in Table 5 are significantly lower than the expected value. These results reveal that most of the estimators tend to produce substantially underestimated values for Markov sources. These findings are consistent with the results reported by [27].

4.2 Correlations of estimators

The Pearson and Spearman coefficients were used to measure the correlation between entropy estimators. To analyze correlation of the estimators mainly three different datasets are used in experiments:

1. **IID sequences with full entropy.** The datasets are generated using the Cipher Block Chaining (CBC) mode of the block cipher Advanced Encryption Standard (AES) [25]. The dataset contains 200 binary sequences of length 1 000 000 were generated. In these sequences, all outputs are assumed to have an equal probability of occurring and are independent. Hence, the outputs satisfy the IID assumption and have full entropy.
2. **IID sequences with entropy=0.5.** The dataset follows a biased binary distribution, where the probability of observing a 1 is 0.7, and the probability of observing a 0 is 0.3. For this dataset, 200 binary sequences of length 1 000 000 were generated. In these sequences, the expected entropy of a sequence is 0.5 per bit, and all terms are generated identically and independently, so sequences satisfy the IID assumption. This data is generated using the random number generator Mersenne Twister (MT19937) in C++.
3. **Non-IID sequences with entropy=0.875.** The dataset follows a biased binary distribution, where the elements of each sequence are generated as follows. Let $S = (s_1, s_2, s_3, \cdots)$ be a sequence of length 1 000 000, all terms of the sequence are generated by the random number generator Mersenne Twister (MT19937) in C++, however for each k, $s_{8k} = \sum_{i=1}^{7} s_{8k-i} \mod 2$; that is, $8k^{th}$ element of the sequence is sum of pre-

vious seven elements in mod 2. This modification reduces the entropy of the sequence in ratio $\frac{1}{8}$. The sequences in this dataset do not satisfy the IID-assumption. This dataset contains 200 binary sequences of length 1 000 000.

4.2.1 Correlation analysis with dataset 1: IID sequences with full entropy

The Pearson and Spearman coefficients were used to measure the correlation between entropy estimators. Using 200 binary sequences of length 1 000 000, Table 6 and Table 7 show the Pearson and Spearman correlations among different estimators, respectively. According to Table 6, a strong or moderate correlation was observed for the (MCV, Markov), (MultiMCW, MultiMMC) (MultiMMC, LZ78Y), and (MultiMCW, LZ78Y) estimators using Pearson's metric. When the same experiments were conducted using Spearman's metric, a correlation was still observed between (MCV, Markov). However, (MultiMMC, LZ78Y) and (MultiMCW, LZ78Y) correlations were no longer as strong. Additionally, the correlation between (Markov, LZ78Y) was observed to be strong for Spearman's metric.

Table 6 Pearson correlation among different estimators for IID sequences with full entropy

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	-0.053	**0.534**	-0.117	0.056	-0.051	0.054	-0.075	0.217	0.261
E2		1.000	0.132	-0.009	0.016	0.056	0.007	-0.028	-0.029	-0.086
E3			1.000	0.035	0.082	-0.016	0.026	-0.058	0.177	0.228
E4				1.000	-0.042	0.028	0.028	-0.001	0.109	0.076
E5					1.000	0.039	0.044	0.058	0.076	0.077
E6						1.000	-0.045	0.006	-0.056	-0.051
E7							1.000	-0.006	**0.470**	**0.806**
E8								1.000	-0.036	-0.028
E9									1.000	**0.469**
E10										1.000

The bold entries in tables highlight correlations that are not negligible

Table 7 Spearman correlation among different estimators for IID sequences with full entropy

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	-0.043	**0.541**	-0.101	0.064	-0.032	-0.060	0.031	0.183	0.499
E2		1.000	0.122	0.028	0.025	0.004	0.014	0.001	0.002	-0.121
E3			1.000	0.049	0.095	-0.022	-0.045	0.051	0.178	**0.642**
E4				1.000	0.014	0.101	0.020	0.020	0.171	0.114
E5					1.000	0.071	-0.010	-0.079	0.032	0.058
E6						1.000	0.040	-0.064	0.019	0.001
E7							1.000	-0.059	0.078	-0.103
E8								1.000	0.018	0.139
E9									1.000	0.198
E10										1.000

The bold entries in tables highlight correlations that are not negligible

4.2.2 Correlation analysis with dataset 2: IID sequences with entropy 0.5

Experiments were repeated with the biased dataset to observe the relations of the estimators when sequences have not full entropy.Similarly, Pearson and Spearman coefficients were used to measure the correlation between entropy estimators. However, the number of highly correlated estimators is seen as the result of experiments. To make accurate observations Benjamini-Hochberg correction [24] applied to the P-values, $p < 0.01$ is assumed to be significant. Tables 14 and 15 in Appendix show p-values for correlation results.

When we interpret Pearson correlation results of estimators for biased binary sequences, we observe a strong correlation for (Markov,MCV), (Compression,MCV), and (Markov, Collision). There was a moderate correlation between the pairs (Collision, MCV) and (Compression, Markov).

According to Spearman's metric, there was a strong correlation between MCV and the estimators Markov, Compression, MultiMCW, MultiMMC, and LZ78Y. Similarly, Compression is highly correlated with MultiMCW, MultiMMC, and LZ78Y. As a result, the mutual correlations of MultiMCW, MultiMMC, and LZ78Y are very strong (Tables 8, 9, 10, and 11).

Table 8 Pearson correlation among different estimators for IID sequences with entropy=0.5

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	**0.490**	**0.838**	**0.758**	0.198	0.036	0.171	**0.206**	0.172	0.172
E2		1.000	**0.717**	**0.284**	0.162	0.043	0.060	0.156	0.061	0.061
E3			1.000	**0.589**	**0.224**	0.030	0.155	0.185	0.156	0.156
E4				1.000	0.156	0.049	0.140	0.128	0.140	0.140
E5					1.000	0.088	0.164	0.148	0.164	0.164
E6						1.000	**0.324**	-0.004	**0.324**	**0.324**
E7							1.000	0.011	**1.000**	**1.000**
E8								1.000	0.012	0.012
E9									1.000	**1.000**
E10										1.000

The bold entries in tables highlight correlations that are not negligible

Table 9 Spearman correlation among different estimators for IID sequences with entropy=0.5

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	**0.451**	**0.821**	**0.738**	**0.280**	0.164	**0.948**	**0.523**	**0.948**	**0.948**
E2		1.000	**0.696**	**0.268**	**0.199**	0.092	**0.415**	0.202	**0.415**	**0.416**
E3			1.000	**0.573**	**0.332**	0.135	**0.779**	**0.404**	**0.779**	**0.780**
E4				1.000	**0.238**	0.146	**0.717**	**0.392**	**0.718**	**0.718**
E5					1.000	0.129	**0.313**	0.160	**0.313**	**0.313**
E6						1.000	**0.196**	0.056	**0.196**	**0.195**
E7							1.000	**0.489**	**0.999**	**0.999**
E8								1.000	**0.489**	**0.489**
E9									1.000	**0.999**
E10										1.000

The bold entries in tables highlight correlations that are not negligible

4.2.3 Correlation analysis with dataset 2: Non-IID sequences with entropy 0.875

Experiments were repeated with simulated biased datasets to measure the relations of the estimators when sequences do not satisfy the IID assumption and do not have full entropy. Pearson and Spearman coefficients were used to measure the correlation between entropy estimators. To make accurate observations Benjamini-Hochberg correction [24] applied to the P-values, $p < 0.01$ is assumed to be significant. Tables 16 and 17 in Appendix show p-values for correlation results.

When we interpret Pearson correlation results of estimators for non-IID biased binary sequences, we observe a moderate correlation for (Markov,MCV) and (Markov,Collision).

According to Spearman's metric; similar to Pearson's metric, there was a moderate correlation for Markov and the MCV and Collision. Moreover, moderate correlations for the pairs (LZ78Y,MCV) and (LZ78Y,Markov) were observed.

Table 10 Pearson correlation among different estimators for Non-IID sequences with entropy=0.875

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	0.130	**0.572**	0.013	-0.027	-0.073	-0.003	-0.042	0.039	0.091
E2		1.000	**0.323**	-0.071	-0.113	0.030	0.076	0.043	0.059	0.127
E3			1.000	-0.018	-0.096	-0.027	-0.057	-0.069	0.067	0.075
E4				1.000	0.112	0.040	0.103	-0.165	0.033	**-0.253**
E5					1.000	-0.066	0.060	0.092	-0.050	-0.060
E6						1.000	0.039	-0.004	0.055	-0.046
E7							1.000	-0.017	-0.014	0.107
E8								1.000	-0.005	0.064
E9									1.000	-0.017
E10										1.000

The bold entries in tables highlight correlations that are not negligible

Table 11 Spearman correlation among different estimators for Non-IID sequences with entropy=0.875

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**
E1	1.000	0.114	**0.521**	-0.043	-0.038	-0.070	-0.041	-0.013	0.105	**0.473**
E2		1.000	**0.273**	-0.045	-0.076	0.062	0.112	-0.037	0.066	0.063
E3			1.000	-0.047	-0.094	-0.035	-0.029	-0.059	0.046	**0.687**
E4				1.000	0.113	0.112	0.036	0.046	0.030	-0.012
E5					1.000	-0.000	0.093	0.053	0.058	-0.113
E6						1.000	-0.011	-0.024	0.053	-0.083
E7							1.000	-0.025	0.051	0.102
E8								1.000	0.004	-0.063
E9									1.000	0.025
E10										1.000

The bold entries in tables highlight correlations that are not negligible

4.3 Impact of the transformations

For this experiment, 200 uniformly distributed sequences of length 1 000 000 with full entropy were used. These sequences were transformed using a reversing, binary derivative and t-rotation for $t = 16, 64, 128, 1024$. Entropy estimates for the original and transformed sequences were compared, and their Pearson and Spearman correlation coefficients are listed in Tables 12 and 13, respectively.

Effect of reversing and rotating the input sequences One of the results of these experiments shows that, for certain entropy estimators including MCV, collision, Markov, t-tuple, and LRS, reversing or rotating the input sequences did not lead to any changes in the estimated entropy values. This result suggests that these estimators are insensitive to reversal, which could be an indication of their robustness.

Effect of binary derivation The binary derivation transformation, which involves XORing consecutive bits to generate a new sequence, effectively impacts local dependencies between adjacent bits. The experimental results show that, for all estimators, the entropy estimates changed after applying this transformation. This can be due to the fact that taking binary

Table 12 Pearson Correlation according to the estimation results of transformed sequences

	Original	Reversed	Bin. Drv.	16-r.	64-r.	128-r.	1024-r.
MCV	1.0000	1.0000	-0.0289	1.0000	1.0000	1.0000	1.0000
Collision	1.0000	1.0000	-0.0160	1.0000	1.0000	1.0000	1.0000
Markov	1.0000	1.0000	0.4586	1.0000	1.0000	1.0000	1.0000
Compress.	1.0000	0.3334	0.4887	0.3379	0.3374	0.3927	0.3368
t-Tuple	1.0000	1.0000	0.1144	1.0000	1.0000	1.0000	1.0000
LRS	1.0000	1.0000	0.7013	1.0000	1.0000	1.0000	1.0000
MultiMCW	1.0000	0.1301	0.8455	0.9999	0.9998	0.9997	0.9994
Lag Pre.	1.0000	0.1492	0.0037	0.9983	0.9971	0.9962	0.9915
MultiMMC	1.0000	0.0564	-0.0189	0.9977	0.9962	0.9962	0.8329
LZ78Y	1.0000	0.0598	0.1510	0.9961	0.9927	0.9918	0.9738

Table 13 Spearman Correlation according to the estimation results of transformed sequences

	Original	Reversed	Bin. Drv.	16-r.	64-r.	128-r.	1024-r.
MCV	1.0000	1.0000	-0.0432	1.0000	1.0000	1.0000	1.0000
Collision	1.0000	1.0000	0.0565	1.0000	1.0000	1.0000	1.0000
Markov	1.0000	1.0000	0.4030	1.0000	1.0000	1.0000	1.0000
Compress.	1.0000	0.3090	0.5283	0.3053	0.3053	0.3685	0.3094
t-Tuple	1.0000	1.0000	0.0964	1.0000	1.0000	1.0000	1.0000
LRS	1.0000	1.0000	0.5425	1.0000	1.0000	1.0000	1.0000
MultiMCW	1.0000	0.8795	0.0170	0.9975	0.9954	0.9947	0.9869
Lag Pre.	1.0000	0.3607	-0.0282	0.9822	0.9717	0.9603	0.9219
MultiMMC	1.0000	0.3762	0.2872	0.9162	0.8772	0.8770	0.6943
LZ78Y	1.0000	0.6069	0.3580	0.9941	0.9884	0.9867	0.9530

derivation may increase entropy for sequences with periodic or structured patterns, as it introduces more randomness. Conversely, for highly random sequences, the transformation can introduce some structure, possibly leading to a decrease in entropy. Our results highlight that applying binary derivation as a conditioning component can significantly impact the entropy estimates, emphasizing the importance of considering such transformations when designing random number generators.

5 Discussion and future directions

In this paper, we examined the black-box entropy estimators outlined in NIST SP 800-90B. We observed that compression and collision estimates tend to underestimate the entropy for uniform and biased distributions, which is consistent with the findings of Zhu et al. [18] and Kim et al. [19]. When focusing on the accuracy of compression estimates, various insights can be drawn. Entropy is inherently a global property of a probability distribution, whereas compression algorithms typically operate on specific sequences, focusing on local patterns. This distinction might be critical, as it suggests that the inherent differences between global and local approaches can significantly impact entropy estimation. Future research could investigate whether the underestimation of entropy by compression algorithms represents a potential vulnerability that could be exploited in predicting or attacking sequences. Alternatively, studies could focus on compression estimate to determine whether it should be reconsidered entirely, emphasizing that accurate entropy estimation might only be achieved through global approaches.

It is also important to note that prediction-based estimators, such as multi-most common in window, lag, or multiMMC methods, are specifically designed to detect weaknesses when the estimation is low.

We observed that the remaining estimates are close to the true entropy for the uniform distribution. However, LRS and lag prediction overestimate entropy for binary, 4-bit, and 8-bit sequences for biased distributions. For prediction-based estimates, overestimations are expected when the underlying model does not fit the distribution of the sequence.

Our experiments also reveal a strong correlation between Markov and MCV tests for uniform distributions. When analyzing correlations in biased datasets that satisfy the IID assumption, we observed an increase in the number of correlated estimators, particularly between MultiMCW, MultiMMC, and LZ78Y are very strong. Additionally, MCV was highly correlated with the estimators including Markov, compression, MultiMCW, MultiMMC, and LZ78Y. The most significant negative correlation found was between MCV and compression, indicating that these methods employ fundamentally different approaches for estimating entropy. Compression, which focuses on local patterns, differs from the other estimators, while MCV provides accurate estimates for IID sequences. This difference explains the observed negative correlation for IID sequences.

On the other hand, when analyzing estimtors for biased datasets that do not satisfy the IID assumption, our experiments show that moderate correlation between (Markov, MCV), (Markov, collision), (LZ78Y,MCV) and (LZ78Y,Markov) estimators.

If efficiency is a priority, selecting one of the highly correlated tests to obtain prediction results is statistically meaningful. However, for detailed analysis or detecting unusual cases, evaluating the results of all estimators is more reliable. Additionally, these moderate or high correlations can be interpreted as an indication that the estimators are working consistently with one another.

Another significant contribution of this study is the emphasis on the role of conditioning components, designed as deterministic transformations, in entropy estimation, particularly when designing random number generators. For future work, it would be valuable to explore the effects of additional deterministic transformations, particularly the ones used in real-world designs. This could include, for example, lagged derivatives of the form $s_i \oplus s_{i+L}$ (in addition to the special case of $L = 1$ in this paper) or the application of linear transformations that can be represented as full-rank linear functions.

We anticipate that the insights provided by this paper will contribute to improving the accuracy of NIST's entropy estimation strategy and promote future studies that consider the impacts of commonly used conditioning or post-processing functions.

Appendix - Supplementary Material

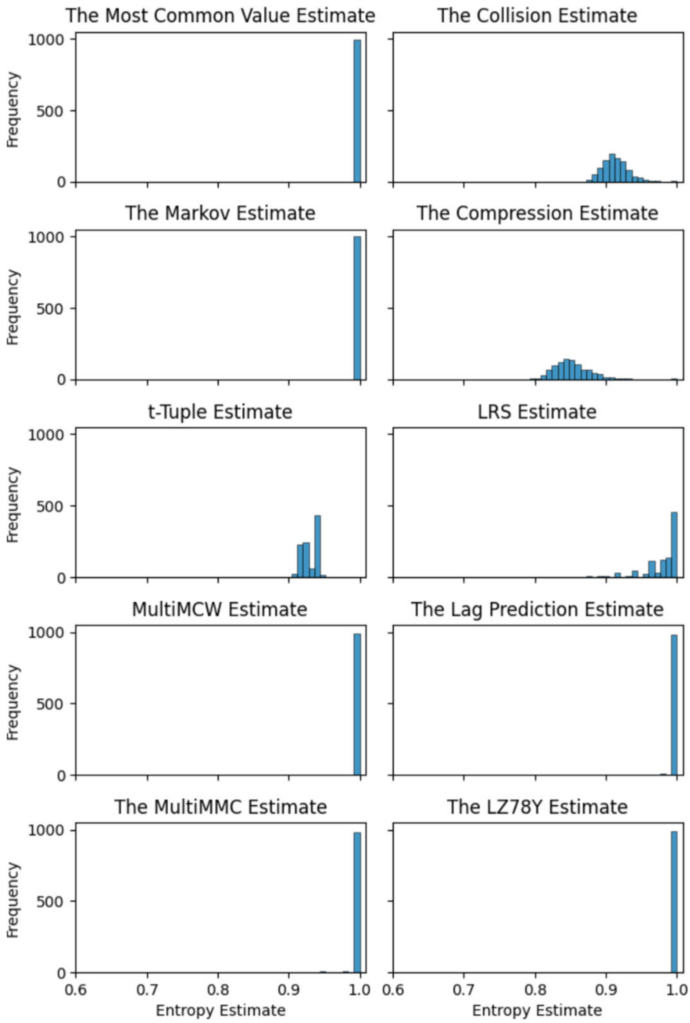

Fig. 1 Distribution of entropy estimates for full-entropy binary inputs

Table 14 P-values of Pearson correlation among different estimators for IID sequences with entropy=0.5

P-values	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	0.000	0.000	0.000	0.000	0.001	0.057	0.002	0.000	0.002	0.002
E2	**0.000**	0.000	0.000	0.000	0.003	0.051	0.037	0.004	0.037	0.037
E3	**0.000**	**0.000**	0.000	0.000	0.000	0.063	0.003	0.001	0.003	0.003
E4	**0.000**	**0.000**	**0.000**	0.000	0.003	0.046	0.006	0.008	0.006	0.006
E5	0.001	0.003	**0.000**	0.003	0.000	0.016	0.003	0.004	0.003	0.003
E6	0.057	0.051	0.063	0.046	0.016	0.000	0.000	0.086	0.000	0.000
E7	0.002	0.037	0.003	0.006	0.003	**0.000**	0.000	0.089	0.000	0.000
E8	**0.000**	0.004	0.001	0.008	0.004	0.086	0.089	0.000	0.089	0.089
E9	0.002	0.037	0.003	0.006	0.003	**0.000**	**0.000**	0.089	**0.000**	0.000
E10	0.002	0.037	0.003	0.006	0.003	**0.000**	**0.000**	0.089	**0.000**	0.000

The bold entries in tables highlight correlations that are not negligible

Table 15 P-values of Spearman correlation among different estimators for IID sequences with entropy=0.5

P-values	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000
E2	**0.000**	0.000	0.000	0.000	0.000	0.020	0.000	0.001	0.000	0.000
E3	**0.000**	**0.000**	0.000	0.000	0.000	0.006	0.000	0.000	0.000	0.000
E4	**0.000**	**0.000**	**0.000**	0.000	0.000	0.004	0.000	0.000	0.000	0.000
E5	**0.000**	**0.001**	**0.000**	**0.000**	0.000	0.007	0.000	0.003	0.000	0.000
E6	0.002	0.020	0.006	0.004	0.007	0.000	0.001	0.042	0.001	0.001
E7	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	0.000	0.000	0.000	0.000
E8	**0.000**	0.001	**0.000**	**0.000**	0.003	0.042	**0.000**	0.000	0.000	0.000
E9	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**	**0.000**	0.000	0.000
E10	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**	**0.000**	**0.000**	0.000

The bold entries in tables highlight correlations that are not negligible

Table 16 P-values of Pearson correlation among different estimators for Non-IID sequences with entropy=0.875

P-values	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	0.000	0.330	0.000	0.915	0.844	0.704	0.970	0.771	0.771	0.550
E2	0.330	0.000	0.000	0.704	0.438	0.840	0.704	0.771	0.704	0.332
E3	**0.000**	**0.000**	0.000	0.906	0.550	0.844	0.704	0.704	0.704	0.704
E4	0.915	0.704	0.906	0.000	0.438	0.771	0.496	0.111	0.831	0.002
E5	0.844	0.438	0.550	0.438	0.000	0.704	0.704	0.550	0.757	0.704
E6	0.704	0.840	0.844	0.771	0.704	0.000	0.771	0.970	0.715	0.771
E7	0.970	0.704	0.704	0.496	0.704	0.771	0.000	0.906	0.914	0.465
E8	0.771	0.771	0.704	0.111	0.550	0.970	0.906	0.000	0.970	0.704
E9	0.771	0.704	0.704	0.831	0.757	0.715	0.914	0.970	0.000	0.906
E10	0.550	0.332	0.704	**0.001**	0.704	0.771	0.465	0.704	0.906	0.000

The bold entries in tables highlight correlations that are not negligible

Table 17 P-values of Spearman correlation among different estimators for Non-IID sequences with entropy=0.875

P-values	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	0.000	0.412	0.000	0.774	0.774	0.770	0.774	0.915	0.466	0.000
E2	0.412	0.000	0.000	0.770	0.709	0.770	0.412	0.774	0.770	0.770
E3	**0.000**	**0.000**	0.000	0.770	0.530	0.774	0.812	0.770	0.770	0.000
E4	0.774	0.770	0.770	0.000	0.412	0.412	0.774	0.770	0.812	0.915
E5	0.774	0.709	0.530	0.412	0.000	0.999	0.530	0.770	0.770	0.412
E6	0.770	0.770	0.774	0.412	0.999	0.000	0.915	0.821	0.770	0.634
E7	0.774	0.412	0.812	0.774	0.530	0.915	0.000	0.821	0.770	0.466
E8	0.915	0.774	0.770	0.770	0.770	0.821	0.821	0.000	0.979	0.770
E9	0.466	0.770	0.770	0.812	0.770	0.770	0.770	0.979	0.000	0.821
E10	**0.000**	0.770	**0.000**	0.914	0.412	0.634	0.466	0.770	0.821	0.000

The bold entries in tables highlight correlations that are not negligible

Acknowledgements The authors thank Sevim Seda Odacıoğlu for her contributions to the implementations of the estimators.

Author Contributions The authors conducted the work collaboratively, with the experiments carried out by M.A. The analyses were performed in conjunction with all authors, and the manuscript was written collaboratively. All authors contributed equally.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests The authors declare no competing interests.

References

1. Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J.A.: Mining your ps and qs: detection of widespread weak keys in network devices. In: Proceedings of the 21st USENIX Conference on Security Symposium. Security'12, p. 35. USENIX Association, USA (2012)
2. Bernstein, D.J., Chang, Y., Cheng, C., Chou, L., Heninger, N., Lange, T., Someren, N.: Factoring rsa keys from certified smart cards: Coppersmith in the wild. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013, pp. 341–360. Springer, Berlin, Heidelberg (2013)

3. Barker, E.B., Kelsey, J.M.: SP 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Technical report, National Institute of Standards and Technology (June 2015). https://doi.org/10.6028/NIST.SP.800-90Ar1
4. Sönmez Turan, M., Barker, E.B., Kelsey, J.M., McKay, K.A., Baish, M.L., Boyle, M.: SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation. Technical report, National Institute of Standards and Technology (January 2018https://doi.org/10.6028/NIST.SP.800-90B
5. Barker, E.B., Kelsey, J.M., McKay, K.A., Roginsky, A., Sönmez Turan, M.: SP 800 90C Recommendation for Random Bit Generator (RBG) Constructions (3rd Draft). Technical report, National Institute of Standards and Technology (September 2022https://doi.org/10.6028/NIST.SP.800-90C.3pd
6. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, N., Dray, J., Vo, S., Bassham, L.: SP 800-22 Rev. 1a A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Technical report, National Institute of Standards and Technology (2010). https://doi.org/10.6028/NIST.SP.800-22r1a
7. ISO Central Secretary: ISO/IEC 19790:2012 Information technology – Security techniques – Security requirements for cryptographic modules. Standard ISO/IEC 19790:2012, International Organization for Standardization, Geneva, CH (2012). https://www.iso.org/standard/52906.html
8. ISO Central Secretary: ISO/IEC 15408-1:2009 Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model. Standard ISO/IEC 15408-1:2009, International Organization for Standardization, Geneva, CH (2015). https://www.iso.org/standard/50341.html
9. ISO Central Secretary: ISO/IEC 18031:2011 Information technology – Security techniques – Random bit generation. Standard ISO/IEC 18031:2011, International Organization for Standardization, Geneva, CH (2011). https://www.iso.org/standard/54945.html
10. ISO Central Secretary: Information technology – Security techniques – Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408. Standard ISO/IEC 20543:2019, International Organization for Standardization, Geneva, CH (2019). https://www.iso.org/standard/68296.html
11. AIS 20: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren (Version 3). Report, Bundesamt für Sicherheit in der Informationstechnik (BSI) (May 2013). https://www.bsi.bund.de/dok/6618284
12. AIS 31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren (Version 3). Report, Bundesamt für Sicherheit in der Informationstechnik (BSI) (May 2013). https://www.bsi.bund.de/dok/6618252
13. Peter, M., Schindler, W.: A Proposal for Functionality Classes for Random Number Generators (Version 2.35, DRAFT) . Report, Bundesamt für Sicherheit in der Informationstechnik (BSI) (September 2022). https://www.bsi.bund.de/dok/ais-20-31-appx-2022
14. L'Ecuyer, P., Simard, R.: Testu01: A C library for empirical testing of random number generators (2007)
15. Marsaglia., G.: The marsaglia random number cdrom including the diehard battery of tests of randomness (1996)
16. Brown, R.G.: Dieharder: A random number test suite (2013)
17. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, M.L.S., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for random and pseudo random number generators for cryptographic applications (2001)
18. Zhu, S., Ma, Y., Chen, T., Lin, J.: JiwuJing: Analysis and improvement of entropy estimatorsin nist sp 800–90b for non-iid entropysources. IACR Transactions on Symmetric Cryptology. **2017**(3), 151–168 (2017)
19. Kim, Y., Guyot, C., Kim, Y.: On the efficient estimation of min-entropy. IEEE Trans. Inf. Forensics Secur. **16**, 3013–3025 (2021)
20. Hill, J.E.: SP 800-90B Refinements: Validation Process, Estimator Confidence Intervals, and Assessment Stability. ICMC (2020)
21. Turan, M.S., Doganaksoy, A., Boztas, S.: On independence and sensitivity of statistical randomness tests. In International Conference on Sequences and Their Applications (SETA). (2008)
22. Pearson, K., National Eugenics, G.L.: "Note on Regression and Inheritance in the Case of Two Parents". Proceedings of the Royal Society. Royal Society, (1895). https://books.google.com/books?id=xst6GwAACAAJ
23. Spearman, C.: The proof and measurement of association between two things. Am. J. Psychol. **15**, 88–103 (1904)
24. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. Roy. Stat. Soc.: Ser. B (Methodol.) **57**(1), 289–300 (1995)

25. Dworkin, M., Mouha, N., Turan, M.S.: Advanced Encryption Standard (AES). Federal Inf. Process. Stds. (NIST FIPS) 197, National Institute of Standards and Technology, Gaithersburg, MD. (2001 (updated 2023))
26. Woo, J., Yoo, C., Kim, Y., Cassuto, Y., Kim, Y.: Generalized lrs estimator for min-entropy estimation. IEEE Trans. Inf. Forensics Secur. **18**, 3305–3317 (2023). https://doi.org/10.1109/TIFS.2023.3280745
27. Kelsey, J., McKay, K.A., Turan, M.S.: Predictive models for min-entropy estimation. Cryptology ePrint Archive. **Report 2015/600** (2015). Accessed: 2025-01-11