Observations on NIST SP 800-90B Entropy Estimators

Melis Aslan
Middle East Technical University
Ankara, TURKEY
melisa@metu.edu.tr

Ali Doğanaksoy
Middle East Technical University
Ankara, TURKEY
aldoks@metu.edu.tr

Zülfükar Saygı
TOBB ETU
Ankara, TURKEY
zsaygi@etu.edu.tr

Meltem Sönmez Turan
National Institute of Standards and Technology
Gaithersburg, MD
meltem.turan@nist.gov

Fatih Sulak
Atilim University
Ankara, TURKEY
fatih.sulak@atilim.edu.tr

Abstract

Random numbers play a crucial role in cryptography, since security of cryptographic protocols relies on the assumption of availability of uniformly distributed and unpredictable random numbers to generate secret keys, nonces, 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 provide 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

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, 4, 5, 6], the International Organization for Standardization (ISO) [7, 8, 9, 10], and Bundesamt für Sicherheit in der Informationstechnik (BSI) [11, 12, 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.

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.

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 discussion.

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 the following chapters of this paper, *entropy* refers to *min-entropy*.

2.2 Entropy Estimation Based on SP 800-90B

SP 800-90B [4] describes an *entropy source* model, that is composed of a noise source, health tests, and an optional conditioning function. The standard also provides guidelines for the generation of 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, a number of 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 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.

Table 1: Entropy estimators of NIST SP 800-90B

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

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 normal distribution.

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.

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 important 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 and used 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 a number of deterministic transformations that are 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 $S = (\texttt{10110001110010})$ is $S' = (\texttt{01001110001101})$.

- **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 $S = (\texttt{10110001110010})$ is $S' = (\texttt{11010010010111})$.

- ***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 $S = (\texttt{101100011100 10})$ is $S' = (\texttt{11000111001010})$.

4 Experimental Results

4.1 Simulated Datasets

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) [24]. 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 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 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++`.

4.2 Accuracy of Entropy Estimators

Table 2 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.

Table 2: Mean and standard deviation of entropy estimators for binary, 4-bit, and 8-bit sources with full entropy

	1-bit		4-bit			8-bit		
	Mean	**Std. Dev.**	**Mean**	**Mean/bit**	**Std. Dev.**	**Mean**	**Mean/bit**	**Std. Dev.**
MCV	0.9951	0.0009	3.9514	0.9879	0.0056	7.6736	0.9592	0.0222
Collision	0.9141	0.0194	*	*	*	*	*	*
Markov	0.9982	0.0011	*	*	*	*	*	*
Compression	0.8535	0.0287	*	*	*	*	*	*
t-Tuple	0.9294	0.0104	3.7799	0.9450	0.0149	7.6736	0.9592	0.0222
LRS	0.9785	0.0262	3.8928	0.9732	0.1131	7.7468	0.9683	0.1878
Multi MCW	0.9954	0.0114	3.9635	0.9909	0.0662	7.8169	0.9771	0.1315
Lag Prediction	0.9957	0.0072	3.9677	0.9919	0.0416	7.8116	0.9764	0.1679
MultiMMC	0.9951	0.0129	3.9616	0.9904	0.0778	7.8197	0.9775	0.1302
LZ78Y	0.9956	0.0096	3.9616	0.9904	0.0778	7.8198	0.9775	0.1302

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 3. Similar to uniform distribution, the compression estimate underestimates entropy for biased distributions. However, LRS and lag prediction overestimate the entropy by approximately 50%. Similar results were obtained for 4-bit and 8-bit samples.

Table 3: Mean and standard deviation of entropy estimators of datasets for biased binary, 4-bit near-uniform, and 8-bit near-uniform distributions

	Biased Binary Dist.		**4-bit Near-uniform**			**8-bit Near-uniform**		
	Mean	**Std. Dev.**	**Mean**	**Mean/bit**	**Std. Dev.**	**Mean**	**Mean/bit**	**Std. Dev.**
MCV	0.5122	0.0009	1.9872	0.4968	0.0050	4.0169	0.5021	0.0160
Collision	0.5095	0.0020	*	*	*	*	*	*
Markov	0.5146	0.0011	*	*	*	*	*	*
Compression	0.3224	0.0009	*	*	*	*	*	*
t-Tuple	0.5031	0.0116	1.9710	0.4928	0.0197	3.9993	0.4999	0.0380
LRS	0.7692	0.0205	3.2364	0.8091	0.0954	6.9466	0.8683	0.1884
Multi MCW	0.5121	0.0055	1.9860	0.4965	0.0200	4.0063	0.5008	0.0738
Lag Prediction	0.7756	0.0263	3.2812	0.8203	0.0923	6.9558	0.8695	0.2984
MultiMMC	0.5118	0.0055	1.9861	0.4965	0.0200	4.1557	0.5195	0.1028
LZ78Y	0.5118	0.0055	1.9860	0.4965	0.0200	4.1556	0.5194	0.1027

4.3 Correlations of Estimators

The Pearson and Spearman coefficients were used to measure the correlation between entropy estimators. Using 200 binary sequences of length 1 000 000, Table 4 and Table 5 show the Pearson and Spearman correlations among different estimators, respectively. According to Table 4, 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 4: Pearson correlation among different estimators for uniform distribution with full entropy

	MCV	**Collision**	**Markov**	**Compression**	**t-Tuple**	**LRS**	**MultiMCW**	**Lag Prediction**	**MultiMMC**	**LZ78Y**
MCV	1.0000	-0.0531	**0.5338**	-0.1170	0.0564	-0.0506	0.0535	-0.0745	0.2174	0.2610
Collision		1.0000	0.1315	-0.0092	0.0163	0.0563	0.0071	-0.0281	-0.0286	-0.0856
Markov			1.0000	0.0347	0.0821	-0.0158	0.0261	-0.0581	0.1767	0.2278
Compression				1.0000	-0.0422	0.0284	0.0281	-0.0011	0.1094	0.0756
t-Tuple					1.0000	0.0388	0.0444	0.0583	0.0760	0.0765
LRS						1.0000	-0.0449	0.0059	-0.0557	-0.0505
MultiMCW							1.0000	-0.0063	**0.4702**	**0.8063**
Lag Prediction								1.0000	-0.0363	-0.0281
MultiMMC									1.0000	**0.4693**
LZ78Y										1.0000

Table 5: Spearman correlation among different estimators for uniform distribution with full entropy

	MCV	**Collision**	**Markov**	**Compression**	**t-Tuple**	**LRS**	**MultiMCW**	**Lag Prediction**	**MultiMMC**	**LZ78Y**
MCV	1.0000	-0.0426	**0.5410**	-0.1012	0.0636	-0.0317	-0.0601	0.0314	0.1825	0.4991
Collision		1.0000	0.1224	0.0282	0.0254	0.0035	0.0140	0.0009	0.0017	-0.1207
Markov			1.0000	0.0491	0.0954	-0.0215	-0.0454	0.0510	0.1784	**0.6420**
Compression				1.0000	0.0138	0.1014	0.0202	0.0200	0.1711	0.1143
t-Tuple					1.0000	0.0714	-0.0104	-0.0789	0.0316	0.0575
LRS						1.0000	0.0396	-0.0641	0.0187	0.0008
MultiMCW							1.0000	-0.0593	0.0784	-0.1028
Lag Prediction								1.0000	0.0178	0.1391
MultiMMC									1.0000	0.1982
LZ78Y										1.0000

4.4 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 the Table 6 and Table 7, respectively. Reversing and rotating the input sequences did not have any impact on its entropy estimation for the MCV, collision, Markov, t-tuple, and LRS estimators (hence, the same estimate is obtained) for either of the correlation metrics. Among different transformations, binary derivative seems to have the highest impact on the prediction based estimates, namely multiMCW, Lag, multiMMC and LZ78Y.

Table 6: Pearson Correlation according to the estimation results of transformed sequences

	Original	**Reversed**	**Bin. Drv.**	**16-rot.**	**64-rot.**	**128-rot.**	**1024-rot.**
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
Compression	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
Multi MCW	1.0000	0.1301	0.8455	0.9999	0.9998	0.9997	0.9994
Lag Prediction	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 7: Spearman Correlation according to the estimation results of transformed sequences

	Original	**Reversed**	**Bin. Drv.**	**16-rot.**	**64-rot.**	**128-rot.**	**1024-rot.**
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
Compression	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
Multi MCW	1.0000	0.8795	0.0170	0.9975	0.9954	0.9947	0.9869
Lag Prediction	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

5 Discussion

In this paper, we studied the black-box entropy estimators described in NIST SP 800-90B. We observed that compression and collision estimates both underestimate the entropy both for uniform and biased distributions, which is consistent with the findings of Zhu

et al. [18] and Kim et al. [19]. 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. Understanding the reasons for this gap based on the details of the estimators is planned for future work.

These experiments show a strong correlation between the Markov and MCV tests for uniform distribution. Additionally, we observed that taking binary derivation significantly changes the entropy estimates, especially for prediction-based estimators.

We expect the provided results to help improve the accuracy of NIST's entropy estimation strategy and promote similar studies to consider the impacts of commonly used conditioning or post-processing functions.

Acknowledgements

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

References

[1] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 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, page 35, USA, 2012. USENIX Association.

[2] Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring rsa keys from certified smart cards: Coppersmith in the wild. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 341–360, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[3] Elaine B. Barker and John M. Kelsey. SP 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Technical report, National Institute of Standards and Technology, June 2015.

[4] Meltem Sönmez Turan, Elaine B. Barker, John M. Kelsey, Kerry A. McKay, Mary L. Baish, and Michael Boyle. SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation. Technical report, National Institute of Standards and Technology, January 2018.

[5] Elaine B. Barker, John M. Kelsey, Kerry A. McKay, Allen Roginsky, and Meltem Sönmez Turan. SP 800 90C Recommendation for Random Bit Generator (RBG) Constructions (3rd Draft). Technical report, National Institute of Standards and Technology, September 2022.

[6] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, N. Heckert, James Dray, San

Vo, and Lawrence Bassham. 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.

[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.

[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.

[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.

[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.

[11] AIS 20: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren (Version 3). Report, Bundesamt für Sicherheit in der Informationstechnik (BSI), May 2013.

[12] AIS 31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren (Version 3). Report, Bundesamt für Sicherheit in der Informationstechnik (BSI), May 2013.

[13] Matthias Peter and Werner Schindler. A Proposal for Functionality Classes for Random Number Generators (Version 2.35, DRAFT) . Report, Bundesamt für Sicherheit in der Informationstechnik (BSI), September 2022.

[14] P. L'Ecuyer and R. Simard. Testu01: A c library for empirical testing of random number generators, 2007.

[15] G. Marsaglia. The marsaglia random number cdrom including the diehard battery of tests of randomness, 1996.

[16] R. G. Brown. Dieharder: A random number test suite, 2013.

[17] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, M. L. Stefan Leigh, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudo random number generators for cryptographic applications, 2001.

[18] Shuangyi Zhu, Yuan Ma, Tianyu Chen, Jingqiang Lin, and 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] Yongjune Kim, Cyril Guyot, and Young-Sik Kim. On the efficient estimation of min-entropy. *IEEE Transactions on Information Forensics and Security*, 16:3013–3025, 2021.

[20] Joshua E. Hill. SP 800-90B Refinements: Validation Process, Estimator Confidence Intervals, and Assessment Stability. ICMC, 2020.

[21] M. Sönmez Turan, A. Doganaksoy, and S. Boztas. On independence and sensitivity of statistical randomness tests. *In International Conference on Sequences and Their Applications (SETA)*, 2008.

[22] K. Pearson and Galton Laboratory for National Eugenics. *"Note on Regression and Inheritance in the Case of Two Parents".* Proceedings of the Royal Society. Royal Society, 1895.

[23] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103, 1904.

[24] Morris Dworkin, Nicky Mouha, and Meltem Sönmez Turan. Advanced Encryption Standard (AES). *Federal Inf. Process. Stds. (NIST FIPS) 197, National Institute of Standards and Technology, Gaithersburg, MD*, 2001 (updated 2023).

Appendix

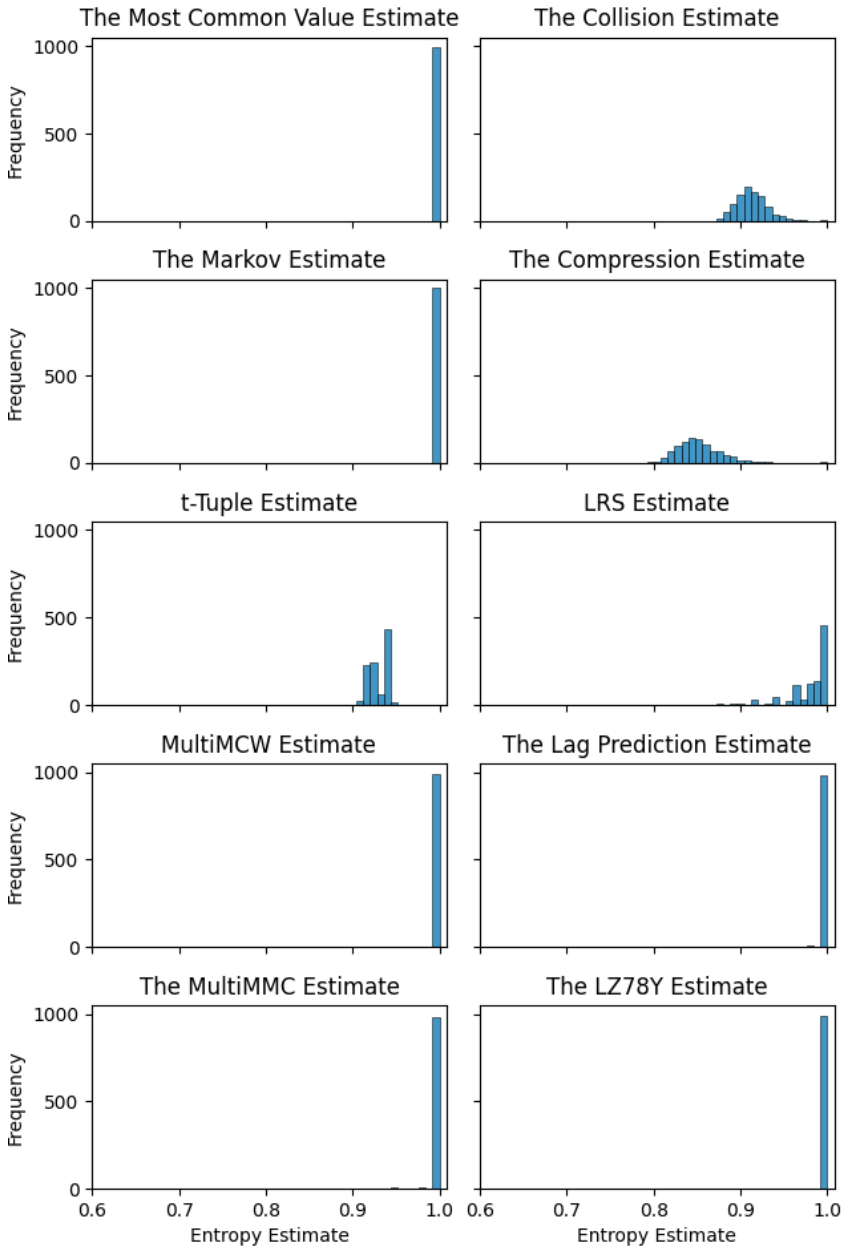

Figure 1: Distribution of entropy estimates for full-entropy binary inputs