Cryptographic Standards for Random Number Generation

Cryptographic Standards for Random Number Generation

Leonardo Errati[1]
Joint work with Roberto La Scala[2] and Mauro Patano[3] ⋆

[1] Politecnico di Torino, Italy,
[2] Università di Bari, Italy,
[3] INFN Bari, Italy

Abstract. Random number generators (RNGs) are an often overlooked cornerstone for secure communications, cryptographic protocols, and scientific simulations. This workshop opens with an overview of the NIST framework for RNGs - from entropy sources to filters and test suites. An important conclusion is that the construction of secure, standardised Quantum RNGs is a multidisciplinary effort; this will be exemplified by an ongoing integration project for distributed IT infrastructures.

Keywords: NIST SP 800-90 · NIST SP 800-22 · Quantum RNGs.

1 Introduction

Cryptographic protocols often delegate the generation of perfect and unbiased randomness to the notation $x \leftarrow\!\$\, S$, meaning "$x$ is chosen uniformly at random in the set S". In the real world, this task is entrusted to *random number generators*. We will explore their construction from high-entropy sources, entropy extractors, and cryptographic primitives according to the framework of the US National Institute of Standards and Technology (NIST), showcasing the multidisciplinary endeavour required to study and deploy them.

Extracting entropy. Access to an entropy source does not directly yield full-entropy random bits. For instance, raw noise measurements may be biased or correlated. A source is modelled as a random variable X, and its unpredictability is measured by the *min-entropy of* X,

$$H_\infty(X) = -\log_2 \max_x \Pr[X = x],$$

De Cifris Koine – CIFRIS25 ACTA – https://doi.org/10.69091/koine/vol-7-W39
⋆ The work of the third author was co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE - DigitAl lifelong pRevEntion" initiative, code PNC0000002, CUP: B53C22006480001.

which intuitively quantifies the worst-case predictability of any outcome x of X. This predictability is relaxed by *accumulation* of many samples and *amplification* via conditioning components. The NIST 800-90 framework is built with the goal of transforming adequate sources into reliable cryptographic randomness.

The NIST framework. Depending on context and taste, random number generators are divided into many classes: PRNGs, DRNGs TRNGs, QRNGs, and many others. The NIST framework focuses on bit generators (RBGs) and classifies them into *Nondeterministic* (NRBGs), which deliver fresh entropy each time, and *Deterministic* (DRBGs), which expand the entropy of an initial seed using cryptographic algorithms. The latter being more practical, we aim at building efficient and secure DRBGs emulating the nice properties of NRBGs.
The NIST Special Publication 800-90 series defines the reference framework for RBGs, each of the three documents addressing a distinct layer of the problem.

2 SP 800-90A: DRBG constructions

Released in 2006 and revised in 2015, *Special Publication 800-90A* [2] specifies approved and FIPS-validated DRBG mechanisms. Their purpose is expanding the initial entropy output via well-established and standardised symmetric primitives: *Hash_DRBG*, *HMAC_DRBG*, and *CTR_DRBG*. Earlier versions included *Dual_EC_DRBG*, an elliptic-curve-based generator withdrawn due to widespread concerns about the existence of a backdoor [4].

Functional model. A DRBG operates as a state machine with well-defined interfaces, outlined in Figure 1. During *instantiate*, the internal state is initialized by consuming entropy input to derive the initial seed. The *generate* function produces output bits while updating the internal state. The *reseed* function incorporates fresh entropy to renew the seed. Finally, the *uninstantiate* interface allows secure deletion of the state. The delicate balance between deterministic seed expansion and entropy injection sustains the security of DRBGs.

Security requisites. The security of a DRBG ultimately depends on the secrecy and quality of its entropy input. In case of state compromise, correct implementations are designed to prevent the disclosure of past outputs (*backtracking resistance*), and approved mechanisms guarantee the unpredictability of future outputs after reseeding (*prediction resistance*). The use of keyed primitives improves the security of the DRBG.

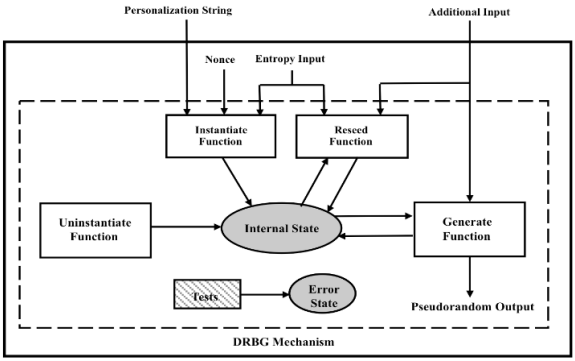

Fig. 1: The functional model of a DRBG. Its underlying purpose is seed expansion for the removal of bias in the entropy input - either inherent to the noise source or artificially introduced. Image by NIST.

3 SP 800-90B: Entropy sources

Special Publication 800-90B [8], finalized in 2018, specifies requirements and validation procedures for entropy sources. Its focus is on the *quality* of the initial entropy provided by physical processes.

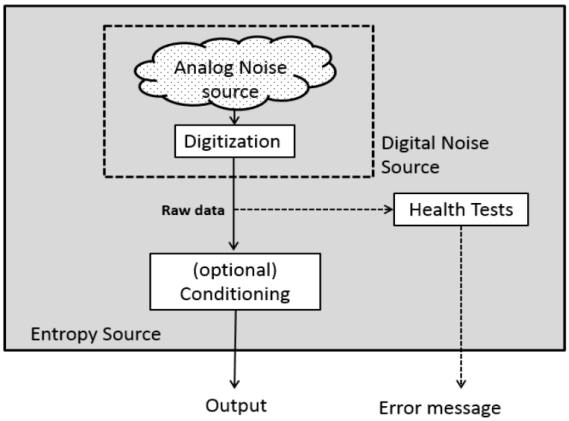

Fig. 2: An entropy source, returning (conditioned) digitalised noise from the underlying noise source. Image by NIST.

Noise sources. Consider the diagram in Figure 2. Noise sources contain the process that is ultimately responsible for the entropy of the output bitstream. They can be *physical* (usuaully dedicated hardware processes, e.g. thermal noise, quantum tunnelling) or *non-physical* (system data, e.g. human inputs). The output distribution of the noise source must be stationary, and its state protected to prevent adversarial knowledge or influence. Testing requires large datasets of raw samples, often at least 10^6, to estimate min-entropy. IID or non-IID assumptions affect which statistical estimators are applied.

Conditioning components. Optional deterministic functions can be used to reduce bias and/or increase the output entropy rate. Vetted conditioning components can be *keyed* (HMAC, CMAC, AES CBC-MAC) or *unkeyed* (approved hash functions, Hash_df, AES Block_Cipher_df). The latter must justify, both theoretically and statistically, how entropy is preserved.

Testing and validation. Since IID noise sources are easier to handle, samples can undergo recommended *IID tests*. More generally, NIST requires *health tests*: startup, restart, monitoring for catastrophic failures, bias tests, statistical tests. These are usually tailored and technology-specific. Validation is performed by NVLAP-accredited laboratories, and results are reviewed via CAVP/CMVP to ensure claimed entropy rates are met under documented conditions.

4 SP 800-90C: Combined Constructions

Special Publication 800-90C [1], published as a draft in July 2024, specifies how DRBG mechanisms defined in SP 800-90A can be securely combined with validated entropy sources compliant with SP 800-90B. These constructions are then suitable for FIPS 140-2 validation.

Constructions. Four main classes are specified:

- **RBG1:** A DRBG without continuous access to an entropy source, instantiated once via an external RBG2(P) or RBG3. It does not support reseeding, prediction resistance, or full-entropy output. Typical in constrained devices.
- **RBG2:** A DRBG with continuous access to validated entropy sources. Optional reseeding, it cannot provide full-entropy output.
- **RBG3:** Includes one or more physical entropy sources and guarantees full-entropy outputs with prediction resistance. Two types: *RBG3(XOR)* XORs source output with a DRBG, *RBG3(RS)* continuously reseeds the DRBG.
- **RBGC:** A chain of RBGs within the same computing platform. The root RBG accesses an initial entropy source and may provide prediction resistance; subsequent RBGCs are seeded from their parent.

Entropy sources. Admissible entropy sources are those validated under SP 800-90B. SP 800-90C prescribes the requirements on their entropy estimates, conditioning functions, health tests, to allow integration.

5 SP 800-22: Statistical Tests for Randomness

Special Publication 800-22 [3] provides a suite of statistical tests designed to evaluate the quality of random sequences. The suite performs empirical assessment by identifying significant deviations from ideal randomness. However, some deviations can be expected even in truly random sequences, and apparent failures may arise from generator anomalies or natural statistical fluctuations. SP 800-22 is intended as a necessary but not sufficient step for evaluating RBGs, and should be complemented by rigorous cryptographic analysis.

Methodology. Each test requires sufficiently long output sequences $u_1, \ldots, u_n$ (typically $n \sim 10^6$) to compute a test statistic Y. The outputs may be individual bits, or real numbers in $[0, 1]$. In the latter case, depending on the test, the values might be discretized. Some tests in the well-known *TestU01* suite by L'Ecuyer and Simard [5] require and assess such discretization.
The null hypothesis H_0 states that the sequence $\{u_i\}_i$ is independent and identically distributed according to the uniform distribution. The p-value will indicate the probability that the observed deviation from H_0 is due to chance. Extremely small or extremely large p-values provide evidence against H_0; the supposed inadequacy of the suggested non-rejection regions was already addressed in a previous edition of this conference [7].
The NIST suite contains fifteen families of statistical tests, each designed to detect specific types of bias: deviations from uniformity, temporal correlations, repetitive structures, etc. Together, all their instances can amount to hundreds of concurrent tests. We present a representative selection of these families.

Frequency (Monobit) Test. Evaluates an imbalance between the number of 0s and 1s in the bitstream. They are mapped to -1 or $+1$ respectively, and their sum $Y = \sum_{i=1}^{n} X_i$ is computed. The null hypothesis H_0 assumes that each bit is IID with equal probability of being 0 or 1. Under H_0, Y follows the Central Limit Theorem and is approximated by a normal distribution, which is then used to calculate the p-value for the test.

Random Excursions (Graph Path) Test. Focuses on the topological properties of the graph sequence. After the bitstream is mapped to $-1/+1$ as above, the cumulative sums $Y_k = \sum_{i=1}^{k} X_i$ form a random walk, whose returns to zero define a cycle; the test counts the number of visits to states $\pm 1, \pm 2, \ldots, \pm 4$ within each cycle. Under H_0, their number follows a discrete-state Markov distribution.

Maurer's Universal Statistical Test [6]. This test evaluates the compressibility of the sequence, under the principle that truly random data should be incompressible. The bitstream is divided into fixed-size blocks of length ℓ, and for each i-th block the distance A_i to its previous occurrence is recorded. After an initialization phase of Q blocks, the next K blocks are used to compute

$$F_K = \frac{1}{K} \sum_{i=1}^{K} \log_2(A_i)$$

which under H_0 has a known expected value μ_ℓ and variance σ_ℓ^2/K, therefore its standardized score

$$Y = \frac{\sqrt{K}(F_K - \mu_\ell)}{\sigma_\ell}$$

is approximately normal.

6 Reliable Randomness in the Wild: a Joint Effort

A Multidisciplinary Collaboration. Classical entropy sources often struggle to provide sufficient randomness for the scale and complexity of modern IT infrastructures. We present our ongoing project, the deployment of a quantum random number generator for INFN computing facilities, which is but an example of the transdisciplinarity of distributing high-quality, high-quantity randomness in real-world infrastructures. Each role is essential for the construction of a validated mechanism (e.g. one following FIPS 140-2 validation), demonstrating how meaningful progress can arise through coordinated, structured, cross-domain effort.

Physics: Quantum Noise Sources. Physicists can leverage quantum processes, such as photon path measurements through beam splitters, laser phase noise, and vacuum fluctuations, to generate high-quality entropy. These noise sources are carefully digitized and conditioned for use by downstream generators. Key tasks include ensuring stability, minimizing bias, and characterizing the statistical properties of the raw signal.

Cryptography: Secure Extraction and Expansion. Cryptographers design the primitives required to convert raw quantum measurements into usable, high-assurance entropy. This involves extraction, deterministic expansion, and integration to guarantee unpredictability, backtracking resistance, and prediction resistance, in line with standards such as SP 800-90.

Mathematics: Statistical Validation. Mathematicians can rigorously assess the quality of randomness. They can produce or assess min-entropy estimates, apply IID and non-IID statistical tests, and verify compliance with standards like SP 800-22, ensuring uniformity, independence, and unpredictability of raw and processed sequences.

Systems Engineering: Distribution and Integration. System architects and IT specialists manage the collection, distribution, and secure transmission of entropy across large-scale infrastructures. They address latency, throughput, and reliability, designing robust, real-time mechanisms that preserve entropy and integrity, and enable seamless integration into operational environments.

7 Conclusion

The emerging area of quantum randomness provides a promising source of high-assurance entropy. Projects such as ours show how diverse areas, such as physics, cryptography, mathematics, and systems engineering, elegantly converge to address real-world security challenges.

References

1. Barker, E.: Recommendation for Random Bit Generator (RBG) Constructions (2024). https://doi.org/10.6028/nist.sp.800-90c.4pd
2. Barker, E.B., Kelsey, J.M.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Jun 2015). https://doi.org/10.6028/nist.sp.800-90ar1
3. Bassham, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., Heckert, N.A., Dray, J.F., Vo, S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications (2010). https://doi.org/10.6028/nist.sp.800-22r1a
4. Bernstein, D.J., Lange, T., Niederhagen, R.: Dual EC: A standardized back door. Cryptology ePrint Archive, Paper 2015/767 (2015), https://eprint.iacr.org/2015/767
5. L'Ecuyer, P., Simard, R.: Testu01: Ac library for empirical testing of random number generators. ACM Transactions on Mathematical Software (TOMS) **33**(4), 1–40 (2007)
6. Maurer, U.M.: A universal statistical test for random bit generators. Journal of cryptology **5**(2), 89–105 (1992)
7. Morgari, G., Bagini, V., Bazzanella, D., Giacchetto, A.: A note on the p-values distribution in nist sp 800-22 rev. 1a statistical tests. In: CIFRIS24 ACTA, De Cifris Koine, vol. 5, pp. 129–131. De Cifris Press (February 2025). https://doi.org/10.69091/koine/vol-5-W24, guglielmo Morgari, 0000-0003-3071-8999; Vittorio Bagini; Danilo Bazzanella, 0000-0001-8837-5736; Alessandro Giacchetto, 0000-0001-8415-5066
8. Turan, M.S., Barker, E., Kelsey, J., McKay, K.A., Baish, M.L., Boyle, M.: Recommendation for the entropy sources used for random bit generation (Jan 2018). https://doi.org/10.6028/nist.sp.800-90b