ETSI TR 104 171 V1.1.1 (2026-03)

TECHNICAL REPORT

Cyber Security (CYBER); Implementation Guidelines for Quantum Random Number Generators

Reference
DTR/CYBER-00163

Keywords
cyber security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

There have been many QRNGs introduced to the commercial market in recent years, each with its own particular set of advantages and limitations. The purpose of the present document is to present a set of reasonable guidelines for implementing Quantum Random Number Generators (QRNGs), and to give the user of such devices a survey of the various characteristics between different implementations. The present document examines each principal aspect of QRNG implementation and discusses in detail the various options and consequences of implementation choices.

Introduction

Random Number Generators (RNGs) are essential in applications requiring security, fairness, and unpredictability. Deterministic Random Number/Bit Generators (DRNGs/DRBGs), also called Pseudo-Random Number Generators (PRNGs), simulate randomness using deterministic algorithms. Physical True Random Number Generators (PTRNGs), on the other hand, rely on indeterministic physical processes to produce statistically random outcomes. Finally, Quantum Random Number Generators (QRNGs), a proper subclass of PTRNGs, exploit inherently quantum phenomena to produce outputs which are not only statistically random but also unpredictable given any prior side-information.

Following, certain key aspects and uses for QRNGs are discussed.

Use cases:

a) Cryptographic Security

Modern cryptography relies on random keys, nonces, and Initialization Vectors (IVs).

Unpredictable random numbers are necessary for:

1) Secure key generation in classical encryption protocols such as RSA, AES and ECC as well as Post-Quantum Cryptography algorithms [i.19] such as ML-DSA, ML-KEM and SLH-DSA.

2) State preparation and/or measurement basis choice in Quantum Key Distribution (QKD) protocols.

b) High-Security Applications

Industries like finance, healthcare, and government communications demand randomness that meets the highest integrity standards. QRNG devices, such as photon-based or vacuum fluctuation-based systems, ensure true unpredictability, essential for long-term security [i.2].

c) Scientific Research and Simulations

QRNGs can provide random sequences critical for high-precision applications, such as Monte Carlo simulations in physics, chemistry, and financial modelling.

Key aspects:

1) Cost and Hardware Requirements

One of the principal obstacles to scaling QRNG technology is the requirement for specialized hardware. Classical hardware-based RNGs are readily available and PRNGs can be implemented purely in software and run on virtually any standard computing device. Instead, QRNGs rely on quantum systems like photon detectors, beam splitters, or quantum vacuum fluctuation detectors to generate random numbers.

These quantum components are currently more expensive to produce and maintain than conventional hardware used for classical RNGs. The higher cost of manufacturing and maintaining QRNG systems limits their use primarily to high-security environments, such as government communications, financial institutions, and military applications, although the learning curve (the inverse-exponential relationship between the number of units built and the number of defects) is reducing the cost of integrated QRNG systems-on-a-chip as the industry goes forward. Today, there are several such commercially available devices being marketed as of 2025.

2) Speed and Throughput

In theory, QRNGs can generate entropy at a significantly higher rate than classical RNGs [i.25]. However, QRNGs have comparatively lower speed and throughput than classical RNGs used in conjunction with a PRNG, because QRNGs are bound by the rate at which quantum phenomena can be measured and processed. Although recent advancements have significantly increased the bit rates of QRNGs - reaching hundreds of megabits per second - they still fall short of the throughput achievable by the combination of classical RNGs and PRNGs, especially in applications that require large volumes of random numbers in real-time, such as large-scale simulations or high-frequency trading.

While significant advancements have improved their speed, QRNGs alone are still currently slower than the classical alternative in use today, High-throughput applications (e.g. simulations) may require hybrid QRNG-PRNG systems, which are in development at the present time.

3) Integration with Existing Systems

Current cryptographic infrastructure is optimized for entropy from classical RNGs that is expanded by PRNGs. Integrating QRNGs requires modifying Hardware Security Modules (HSMs), cryptographic libraries, and communication protocols to accommodate quantum randomness. HSMs are specialized hardware devices designed to safeguard and manage cryptographic keys. QRNGs are increasingly being incorporated into HSMs to improve the quality of the random numbers used for key generation and other cryptographic functions. By using QRNGs, HSMs can provide greater security assurances in environments where cryptographic operations are intended to be highly secure and resistant to attack, such as in financial services, healthcare, and critical infrastructure.

4) Scalability

Miniaturization of QRNG devices is improving, but scaling them for widespread consumer applications (e.g. IoT devices or mobile hardware) remains a challenge. Reducing power consumption and production costs is essential for broader adoption.

5) Lack of interface and control plane standards

There are no generally applicable standards for interface to quantum random generators, and this is a problem that can be solved by participation in a standards body such as ETSI, to make quantum random number generators accessible across a broad base of applications. The industry may want to develop a set of common interface standards or even a common command interface and set of common opcodes for controlling the quantum random number generator and a real-world deployment. Organizations like the National Institute of Standards and Technology (NIST) in the United States are already exploring ways to integrate quantum technologies into cryptographic standards as part of their post-quantum cryptography initiative.

1 Scope

The scope of the present document is limited to discussion of the implementation guidelines and characteristics of Quantum Random Number Generators (QRNGs), with an emphasis on what may require standardization and why. A set of informative annexes is also included in the present document to provide a survey of the current state of the art and some real-world examples and experimental results in order to provide some additional aspects that may be of help to the implementer.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Ma, X., Yuan, X., Cao, Z., Qi, B., & Zhang, Z. (2016): "Quantum random number generation", NPJ Quantum Information, 2, 16021.

[i.2] Pironio, S., Acin, A., Massar, S., de la Giroday, A. B., Matsukevich, D. N., Maunz, P. & Monroe, C. (2010): "Random numbers certified by Bell's theorem", Nature, 464(7291), 1021-1024.

[i.3] Sanguinetti, B., Martin, A., Zbinden, H., & Gisin, N. (2014): "Quantum Random Number Generation on a Mobile Phone", Physical Review X, 4(3), 031056.

[i.4] M. J. Applegate, O. Thomas, J. F. Dynes, Z. L. Yuan, D. A. Ritchie, A. J. Shields (2015): "Efficient and robust quantum random number generation by photon number detection", Applied Physics Letters, 107(7).

[i.5] Shen, Y., Tian, L., Zou, H. (2010): "Practical quantum random number generator based on measuring shot noise of vacuum states", Physical Review A, 81(6), 063814.

[i.6] Symul, T., Assad, S. M., & Lam, P. K. (2011): "Real time demonstration of high bitrate quantum random number generation with coherent laser light", Applied Physics Letters, 98(23), 231103.

[i.7] Jofre, M., Curty, M., Fernández, V., Martínez, A., Ortiz, J., Torres, J. P., & Pruneri, V. (2011): "True random numbers from amplified quantum vacuum", Optics Express, 19(21), 20665-20672.

[i.8] Abellán, C., Amaya, W., Mitrani, D., Pruneri, V., & Mitchell, M. W. (2014): "Ultra-fast quantum randomness generation by accelerated phase diffusion in a pulsed laser diode", Optics Express, 22(2), 1645-1654.

[i.9] Marangon, D. G., Vallone, G., & Villoresi, P. (2017): "Source-Device-Independent Ultrafast Quantum Random Number Generation", Physical Review Letters, 118(6), 060503.

[i.10] Frauchiger, D., Renner, R., & Troyer, M. (2013): "True randomness from realistic quantum devices", arXiv preprint arXiv:1311.4547.

[i.11] Müller-Quade, J., & Renner, R. (2009): "Composability in quantum cryptography", New Journal of Physics, 11(8), 085006.

[i.12] Tomamichel, M. (2015): "Quantum information processing with finite resources: mathematical foundations", vol. 5, Springer.

[i.13] Senno, G., Strohm, T., & Acín, A. (2023): "Quantifying the intrinsic randomness of quantum measurements", Physical Review Letters, 131(13), 130202.

[i.14] Meng, S., Curran, F., Senno, G., Wright, V. J., Farkas, M., Scarani, V., & Acín, A. (2024): "Maximal intrinsic randomness of a quantum state", Physical Review A, 110(1), L010403.

[i.15] Curran, F., Moradi, M., Senno, G., Stobinska, M., & Acín, A. (2025): "Maximal intrinsic randomness of noisy quantum measurements", arXiv preprint arXiv:2506.22294.

[i.16] Foreman, C., Yeung, R., Edgington, A., & Curchod, F. J. (2025): "Cryptomite: A versatile and user-friendly library of randomness extractors", Quantum, 9, 1584.

[i.17] Zhang, X., Nie, Y. Q., Liang, H., & Zhang, J. (June 2016): "FPGA implementation of Toeplitz hashing extractor for real time post-processing of raw random numbers", In 2016 IEEE™-NPSS Real Time Conference (RT) (pp. 1-5).

[i.18] Quside Technologies: "Quantum Random Number Generators (QRNGs)".

[i.19] NIST FIPS 204: "Module-Lattice-Based Digital Signature Standard", 13 August 2024.

[i.20] NIST SP 800-22 Rev. 1: "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications".

[i.21] ISO/IEC 18031:2025: "Information technology — Security techniques — Random bit generation".

[i.22] NIST SP 800-90A Rev.1: "Recommendation for RNG using Deterministic Random Bit Generators", June 2015.

[i.23] NIST SP 800-90B: "Recommendation for the Entropy Sources Used for Random Bit Generation", January 2018.

[i.24] NIST SP 800-90C: "Recommendation for Random Bit Generator (RBG) Constructions".

[i.25] United Kingdom National Cyber Security Centre (NCSC): "Quantum Networking Technologies". Version 1.0, 5 August 2025.

[i.26] ETSI TS 131 101 (V16.0.0): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; UICC-terminal interface; Physical and logical characteristics (3GPP TS 31.101 version 16.0.0 Release 16)".

[i.27] ETSI TS 131 102 (V18.4.0): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; Characteristics of the Universal Subscriber Identity Module (USIM) application (3GPP TS 31.102 version 18.4.0 Release 18)".

[i.28] ETSI TS 131 111 (V16.1.0): "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) (3GPP TS 31.111 version 16.1.0 Release 16)".

[i.29] Santha, M., & Vazirani, U. V. (1986): "Generating quasi-random sequences from semi-random sources", Journal of computer and system sciences, 33(1), 75-87.

[i.30] Van Griensven, Rosas, Pecen: "Quantum Proofing the Economy", Applied Quantum Technologies Institute (AQT) publications, 2025.

[i.31] Matthias Peter, Werner Schindler: "A Proposal for Functionality Classes for Random Number Generators", Version 3.0, 2024.

[i.32] ETSI TS 133 110 (V18.0.0): "Universal Mobile Telecommunications System (UMTS); LTE; Key establishment between a Universal Integrated Circuit Card (UICC) and a terminal (3GPP TS 33.110 version 18.0.0 Release 18)".

[i.33] GSMA™ SGP.25: "eUICC for Consumer and IoT Devices Protection Profile",Version 2.0, 19 December 2023.

[i.34] GSMA™ SGP.22: "RSP Technical Specification", Version 3.1 Final, 01 December 2023.

[i.35] FIPS Pub 140-3: "Security Requirements for Cryptographic Modules", 22 March 2019.

[i.36] BSI 20/31: "A Proposal for Functionality Classes for Random Number Generators", 10 September 2024.

[i.37] ISO/IEC 19790:2025: "Information security, cybersecurity and privacy protection — Security requirements for cryptographic modules", February2025.

[i.38] ISO/IEC 24759:2025: "Information technology — Security techniques — Test requirements for cryptographic modules", Fourth edition 2025.

[i.39] ISO/IEC 20543:2019: "Information technology — Security techniques — Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408", First edition 2019-10.

[i.40] ISO/IEC 15408-1:2022: "Information security, cybersecurity and privacy protection — Evaluation criteria for IT security", Fourth edition, 2022-08.

[i.41] ETSI GS QKD 014 (V1.1.1): "Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API".

[i.42] Recommendation ITU-T X.1702 (11/2019): "Quantum noise random number generator architecture".

[i.43] IETF RFC 4086: "Randomness requirements for security", June 2005.

[i.44] NIST SP 800-160v1r1: "Engineering Trustworthy Secure Systems".

[i.45] ITU-T Y.3800 series: "Quantum key distribution networks – Role in end-to-end cryptographic services with non-quantum cryptography", November 2023.

[i.46] ISO/IEC 17025:2017: "General requirements for the competence of testing and calibration laboratories", Third edition, 2017-11.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACL	Access Control List
ADC	Analog-to-Digital Convertor
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AIS	Additional Information Sequence
ANSSI	Agence nationale de la sécurité des systèmes d'information (French National Agency for the Security of Information Systems)
API	Application Programming Interface
AXI	Advanced eXtensible Interface
BF	Bloom Filter

BSI	Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security)
CCCS	Canadian Centre for Cyber Security
CCN	Centro Criptológico Nacional
CPU	Central Processing Unit
DMA	Direct Memory Access
DRBG	Deterministic Random Bit Generator
DRNG	Deterministic Random Number Generator
DSP	Digital Signal Processing
EBF	Element-based sliding Bloom Filter
ECC	Elliptic Curve Cryptography
EMI	Electromagnetic Interference
ENT	Enterprise Network and Telecommunications
EU	European Union
eUICC	Embedded UICC
EZT	Entropy Zero Trust
FIPS	Federal Information Processing Standards (United States)
FPGA	Field-Programmable Gate Array
FPR	Floating Point Register
FRM	Fast Resource Management
FRMs	Faraday Rotator Mirrors
GS	Ground Station
GSMA	GSM Association
HSM	Hardware Security Module
HWRoT	Hardware Root of Trust
ID	Identity
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IID	Independent and Identically Distributed
IoT	Internet of Things
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LED	Light Emitting Diode
LFSR	Linear Feedback Shift Register
LO	Local Oscillator
ML-DSA	Module-Lattice-based Digital Signature standard
ML-KEM	Module-Lattice-based Key-Encapsulation Mechanism standard
NIST	National Institute of Standards and Technology
OS	Operating System
PBF	Partitioned sliding Bloom Filter
PCB	Printed Circuit Board
PCIe	Peripheral Component Interface express
PIN	Personal Identity Number
POVM	Positive Operator-Valued Measure
PPM	Prediction by Partial Matching
PQC	Post-Quantum Cryptography
PQU	Post Quantum Unit
PRNG	Pseudo-Random Number Generator
PSA	Protocol Service Availability
PTRNG	Physical True Random Number Generator
QES	Quantum Entropy Source
QKD	Quantum Key Distribution
QRNG	Quantum Random Number Generator
RBG	Random Bit Generator
RF	Radio Frequency
RFC	Request for Comments
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman

NOTE: A public-key encryption algorithm.

SLH-DSA	Stateless Hash-Based - Digital Signature Standard

SoC	System on Chip
SP	Special Purpose
SWaP	Size, Weight and Power
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TRNG	True Random Number Generator
UICC	Universal Integrated Circuit Card
UK	United Kingdom
URL	Universal Record Locator
USB	Universal Serial Bus
USB-C	Universal Serial Bus Type-C
VPN	Virtual Private Network

4 Theory of quantum random number generation

4.0 Introduction

The purpose of this clause is to show a theoretical basis for random number generation and what this might mean in practical implementations.

4.1 The definition of randomness

QRNGs are devices that use quantum mechanics' inherent unpredictability to produce *true random numbers*, i.e. numbers that are ϵ-close to being uniformly random and independent of all prior information [i.1] to [i.10].

Definition 1. A QRNG's outcome K is ϵ-*secure* (or ϵ-truly-random) if:

$$D\left(\rho_{KE}, \frac{I}{|K|} \otimes \rho_E\right) \leq \epsilon, \tag{1}$$

where:

1) $\rho_{KE} = \sum p(k)|k\rangle\langle k| \otimes \rho_E^k$ is the classical-quantum state describing the correlations between the classical random variable *K* and the (in general, quantum) state ρ_E of a potential eavesdropper's system *E*, where E is considered to be *side-information*.

2) $I/|K|$ is a uniform distribution over strings of length $|K|$ (usually, 2^n for some bit string length n).

3) $D(\sigma, \tau) \coloneqq \frac{1}{2}\|\sigma - \tau\|_1$ is the trace-distance between states σ and τ.

The role of system E, the side-information, in the definition of ϵ-security is to model all degrees of freedom that are, although relevant to the QRNG's operation, outside of the manufacturer (and, hence, of the honest user) control. In other words, these are untrusted sources of stochasticity (or, noise) which influence the QRNG's outcomes. The implicit quantification over all (the continuously) many states ρ_E might seem, at first, to turn this definition impractical. However, as shown next, the *physical modelling* of the device directly provides a way to compute the distance in Eq. (1) for the worst-case ρ_E.

Before moving on, there are two central aspects worth highlighting about the definition of randomness:

1) It is information-theoretical, i.e. without any computational assumption about the eavesdropper's power.

2) By being based on the trace distance, it satisfies the property of *universal composability* [i.11]. In short, this means that any cryptographic protocol (be it classical or quantum) which is ϵ'-secure when taking "ideal" random inputs remains $(\epsilon + \epsilon')$-secure when receiving "real" ϵ-secure inputs.

4.2 Main components of a QRNG

4.2.0 Introduction

QRNGs are comprised of two main modules: a Quantum Entropy Source (QES), where an inherently random process is probed to produce raw outcomes with some degree of unpredictability, and a classical randomness-extraction procedure which transforms the raw outcomes into a, usually shorter, sequence satisfying Eq. (1) for some prescribed value of ϵ.

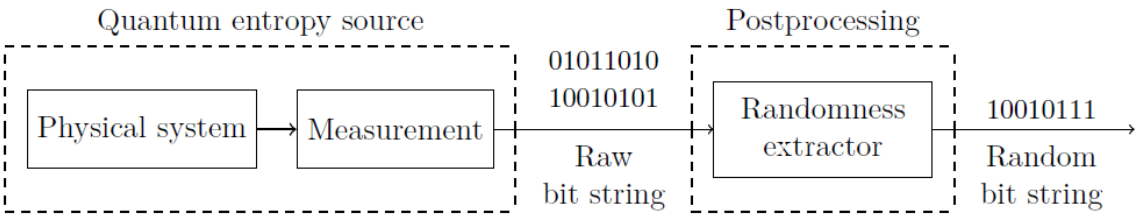


Figure 1: Components of a QRNG

4.2.1 Quantum entropy source

On every use of a QRNG, the action of its QES can be described as the preparation of a quantum system S in some state ρ_S followed by its measurement to produce some classical outcome A. The outcome statistics are described by some Positive Operator-Valued Measure (POVM) $\{M_S^a\}_a$ and they satisfy the Born rule:

$$\Pr[A = a] = \mathrm{Tr}[M_S^a \rho_S]. \tag{2}$$

A natural prerequisite of any QES is that there exists some outcome a such that $\Pr[A = a] \notin \{0,1\}$, i.e. that there is some degree of indeterminism in the measurement's outcomes. However, indeterminism does not imply unpredictability.

The reference to the concept of *entropy* in the denomination of this QRNG's component comes from the fact that the prevailing way to extract ϵ-secure numbers from it is via the use of min-entropy extractors (see clause 4.2.2). In short, these are extraction procedures that target all sources with an assumed lower bound on the *conditional min-entropy* $H_{\min}(A|E)$ of their outcomes A given the side-information E.

Definition 2. Let a quantum system S be in a state ρ_S and subject to a measurement described by a POVM $\{M_S^a\}_a$. The *conditional min-entropy* $H_{\min}(A|E)$ is given by [i.13]:

$$H_{\min}(A|E, \langle \rho_S, \{M_S^x\}_x \rangle) := -\log_2 P_{guess}(A|E, \langle \rho_S, \{M_S^x\}_x \rangle) \tag{3}$$

with

$$P_{guess}(A|E, \langle \rho_S, \{M_S^x\}_x \rangle) := \max_{|\psi\rangle_{SME}, \{\Pi_{SM}^a\}_a, \{M_E^a\}_a} \sum_a \langle \psi | \Pi_{SM}^a \otimes M_E^a | \psi \rangle \tag{4}$$

subject to:

$$\mathrm{Tr}_{ME}[|\psi\rangle\langle\psi|] = \rho_S \tag{5}$$

$$\mathrm{Tr}_M[\Pi_{SM}^a (I \otimes \mathrm{Tr}_{SE}[|\psi\rangle\langle\psi|])] = M_S^x \tag{6}$$

$$\mathrm{Tr}[(\Pi_{SM}^a \otimes I)|\psi\rangle\langle\psi|] = \mathrm{Tr}[M_S^x \rho_S] \tag{7}$$

$H_{\min}(A|E)$ quantifies how *unpredictable* the QES's outcome A is for an eavesdropper holding side-information E about them. Next, this notion is shown through a series of toy examples.

NOTE: In fact, the amount of ϵ-secure extractable from a min-entropy source is characterized, up to second order, by the smooth min entropy, but this goes outside of the scope of the present document. The interested reader is referred to [i.12].

Example no. 1: Consider a QES that prepares a qubit S in a uniform superposition of computational basis states and then measures it in that same basis. That is:

$$\rho_S = |+\rangle\langle+|, \text{ with } |+\rangle_S = \frac{|0\rangle+|1\rangle}{\sqrt{2}}. \tag{8}$$

$$M_s^a = |a\rangle\langle a|_S \tag{9}$$

The state ρ_S and the measurement $\{M_s^a\}_a$ are "noiseless" (formally, extremal elements of the respective convex sets). For this ideal situation, quantum mechanics not only says that the measurement outcome A is uniformly random but also that it is uncorrelated with the result of any other measurement made on any other system, i.e. completely unpredictable. Hence,

$$H_{\min}(A|E) = H_{\min}(A) = -\log_2 \max_a |\langle+|a\rangle|^2 = 1. \tag{10}$$

In fact, for this particular case, a stronger fact holds: A is maximally ϵ-secure (i.e. with $\epsilon = 0$), so no extraction procedure is needed (or, equivalently, the identity extractor applies).

Example no. 2: Now, eliminate the assumption of an ideal (i.e. pure) state preparation, and consider that the preparation of S is affected by uncoloured noise. Specifically, assume that every time the QES attempts to prepare a qubit S in the state $|+\rangle$, there is an unavoidable and uncontrolled source of noise that shrinks the Bloch sphere uniformly along x, y, and z to a radius of $(1-p)$. Hence, for some $p \in [0,1]$,

$$\rho_S = (1-p)|+\rangle\langle+| + p\frac{I}{2}. \tag{11}$$

First of all, notice that, in spite of the depolarizing noise process, the measurement outcomes remain uniformly random independently of p, i.e.

$$\Pr[A=0] = \mathrm{Tr}[\rho_S|0\rangle\langle 0|] = (1-p)\times 1/2 + p\times 1/2 = 1/2. \tag{12}$$

Therefore, this toy QES is compatible with a stochastic model of an unbiased coin (even independent and identically distributed in sequential uses) and with, therefore, a claim of full min-entropy

$$H_{\min}(A) = 1. \tag{13}$$

However, for cryptography, one should not assume that a noise process that is uncontrollable by a QES manufacturer is also so by a potential eavesdropper. This rationale is captured [i.14] by the fact that:

$$H_{\min}(A|E) = 1 - 2\log_2\left(\sqrt{1-\frac{p}{2}} + \sqrt{\frac{p}{2}}\right). \tag{14}$$

Notice how $H_{\min}(A|E)$ goes to 0 as the amount of noise p goes 1. Contrasting this with the fact that $H_{\min}(A) = 1$, shows how the notions of uniformity and unpredictability can be completely divergent.

Example no. 3: Finally, eliminate the assumption that the measurement is ideal and consider that itself is also affected by uncoloured noise (which, for simplicity, it is considered of equal degree p). Hence,

$$\rho_S = (1-p)|+\rangle\langle+| + p\frac{I}{2}, \tag{15}$$

$$M_s = (1-p)|0\rangle\langle 0| + p\frac{I}{2}. \tag{16}$$

For this case, it can be shown [i.15] that, while $H_{\min}(A) = 1$ for all p, $H_{\min}(A|E)$=0 for $p \geq 1 - \frac{1}{\sqrt{2}} \approx 0.29$, making the contrast between uniformity and unpredictability even starker.

It is important to notice that the depolarizing channel, although pervasive in quantum experiments, is a worst-case type of noise model. Therefore, in a more detailed modelling of the untrusted noise process, the difference between the unconditional and conditional min-entropies need not be this large. These examples just go to show the pitfalls of considering the former over the latter when evaluating QRNGs.

In practice, the state of the quantum system and the measurement with which it is probed will depend on the values of certain parameters, usually varying with time. Moreover, the subset of experimentally observable parameters $\vec{p}$ might only restrict the actual state and measurement to belong to some set $\mathcal{M}(\vec{p})$ (rather than completely determining them). This fact can be shown with the concept of a *physical model*.

Definition 3. A *physical model* of a QES consists of a finite number m of parameters and a map:

$$\vec{p} = p_1, \ldots, p_m \in \mathbb{R}^m \longmapsto \mathcal{M}(\vec{p}) \subseteq States \times POVMs \tag{17}$$

such that, at any given point in time, if the parameters take the values $\vec{p}$, then the actual QES's state and measurement belong to $\mathcal{M}(\vec{p})$.

The *physical model* of a QES encapsulates the QRNG's manufacturer's assumptions about its inner workings. **The claim of ϵ-security will directly rest on these assumptions**.

In QRNGs using min-entropy extractors, together with the physical model, the QRNG's manufacturer will usually provide lower bounds:

$$H_{min}\left(A\middle|E,\vec{p}_{typ}\right) \geq \min_{\langle\rho_S,\{M_S^x\}_x\rangle\in\mathcal{M}(\vec{p})} H_{\min}(A|E,\langle\rho_S,\{M_S^x\}_x\rangle) \tag{18}$$

for $\vec{p} = \vec{p}_{typ}, \vec{p}_{wc}$ typical and worst-case values of the model's parameters (potentially for different ranges of the QRNG's operation conditions, such as, e.g. temperature).

4.2.2 Randomness extractor

The randomness extractor component of a QRNG is in charge of *extracting* ϵ-secure strings from the QES's outcomes. In this clause, a succinct description of the theory is shown. For an in-depth review of the subject in the context of quantum information theory, the reader is referred to [i.16].

In simple terms, a randomness extractor is needed in a QRNG because the raw quantum output is not perfectly random yet in a practical implementation. This is because a QRNG relies on a quantum process, like photon detection, that is fundamentally unpredictable but is affected by real-world imperfections:

- Electronic noise
- Drift of detector characteristics over time
- Detector bias, causing one outcome to happen slightly more often, or spurious detector response due to photons arriving too quickly before the detector recovers from the previous photon.

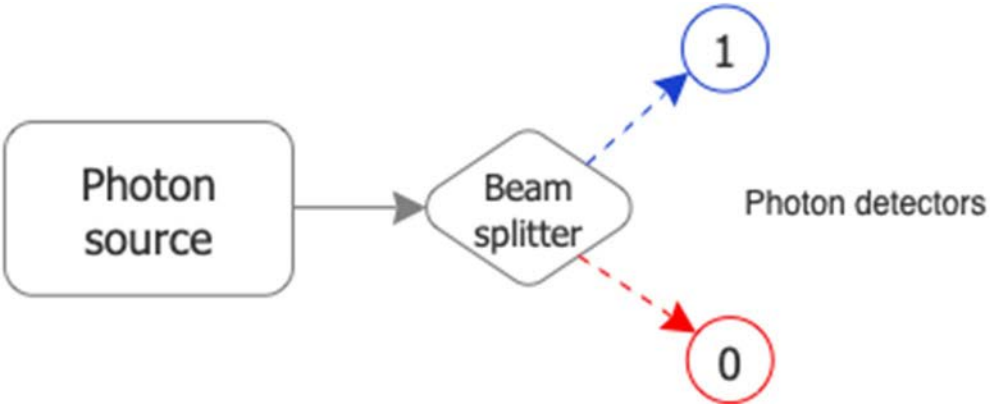


Figure 2: Simplified photonic random number generator

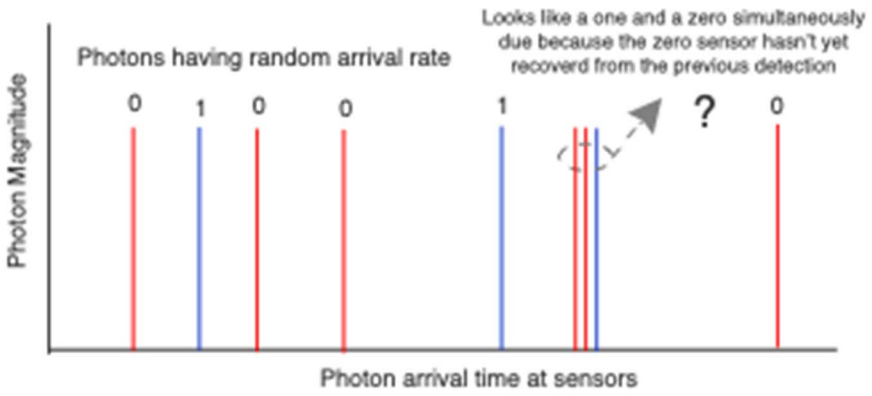


Figure 3: Example of photon arrival times at sensors

For example, in some situations, even though the arrival rate of the ones and zeros is random, a sensor that is not yet recovered from the last detection is one way that can bias the output to produce recognizable patterns, even if the underlying photon generation is a random quantum phenomenon. In this example, both detectors think there is a simultaneous one and zero detected.

So the raw output may look like this:

- 01001100...

Therefore, "unpredictable" is not the same as "uniform" [i.18], and although Quantum physics guarantees unpredictability, many applications such as cryptography, simulations, etc. require equiprobability. This means that each bit would be 50/50, with no correlations between bits. Therefore, raw QRNG data may be unpredictable yet biased, for example:

- 1 appears 51 % of the time, 0 appears 49 %

A case like this is not acceptable for cryptography, even though it came from a quantum source.

In summary, the extractor cleans the randomness by taking biased, imperfect quantum data and outputs fewer bits that are as close as possible to uniformly distributed and independent of untrusted noise sources.

Without an extractor, an attacker could exploit small biases, the output might fail statistical testing for uniformity and cryptographic security proofs break. There are three types of extractors:

1) *Deterministic.* As its name indicates, these are simply functions $\mathrm{EXT}: \{0,1\}^* \to \{0,1\}^*$. This type of extractor can only exist for specific families of entropy sources (be them quantum or classical). In fact, in a celebrated result [i.29], Santha and Vazirani showed that for a natural family of entropy sources defined by the property:

$$\frac{1}{2} - \epsilon \leq p(x_i | x_{i-1}, \ldots, x_1) \leq \frac{1}{2} + \epsilon, \tag{19}$$

no deterministic extractor exists.

2) *Min-entropy extractors*. These are procedures that target QESs X for which a lower bound on the conditional min-entropy $H_{\min}(X|E) \geq k$ is assumed. There are two types of min-entropy extractors:

   a) *Seeded extractors*. These are procedures $\mathrm{EXT}: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ that, in addition to the QES X, take a *seed* S (to act as catalyst for the extraction process) which should be ϵ'-close to uniformly random and independent of, both, X and the side-information, i.e.

$$D\left(\rho_{XSE}, \frac{I}{|S|} \otimes \rho_{XE}\right) \leq \epsilon'. \tag{20}$$

   Where S is an ϵ'-seed. If $\Im = \{f_s: \{0,1\}^n \to \{0,1\}^m\}_s$ is a two-universal family of hash functions, the *leftover hashing lemma* shows that the function $\mathrm{EXT}(x, s) = f_s(x)$ is a seeded extractor, i.e.

$$D\left(\rho_{EXT(X,S)SE}, \frac{I}{2^m} \otimes \rho_{SE}\right) \leq \epsilon + \epsilon'. \tag{21}$$

as long as:

$$m \leq H_{\min}(X|E) - 2\log_2(1/\epsilon). \quad (22)$$

Notice that the output of extractor is (close to) independent not only of the side-information but also of the seed S, implying that the latter can be reused. Extractors with this property are termed *strong*.

NOTE: $\Im$ is two-universal iff for all $x \neq y$ have that $\Pr_{f\sim\Im}[f(x) = f(y)] \leq 1/2^m$.

b) *Two-source extractors:* Instead of ideal randomness, these extractors $\mathrm{EXT}: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ require the additional entropy source Y to have sufficient conditional min-entropy $H_{\min}(Y|E) \geq k_a$ and that some type of independence holds between X, Y and E, the most realistic being that they constitute a quantum Markov chain, i.e. $I(X:Y|E) = 0$.

In [i.16], the reader may find Python implementations of different (quantum-proof) randomness extractors. For FPGA implementations, the reader may look at one for the Toeplitz extractor given in [i.17].

Before moving on to the following clause, it is important to realize that the security definitions given above are relevant when a QRNG is intended for cryptographic applications. If, on the other hand, one is only interested in generating randomness for numerical simulations, then these definitions, although sufficient, are not necessary. Namely, one only needs good statistical properties (i.e. small deviations from uniformity) but does not need unpredictability. In fact, the completely predictable source given in Example no.3 is perfect for this task. Even if the source exhibits some bias, one can apply postprocessing techniques to remove it (see Annex C).

5 Implementation Guidelines for QRNGs

5.0 Introduction

The purpose of the following clauses is to point out some of the practical implications of building quantum random number generators and to identify some possible areas in which normative specifications might be applicable.

5.1 Quantum entropy sources

5.1.1 Quantum integrity

The claim that the outcomes of a purported QRNG are ϵ-secure should be based on a physical model of its QES, as discussed in the preceding clause. Devices should not use marketing terminology such as "quantum-enhanced" unless accompanied by supporting evidence and alignment with the EZT model (see clause 5.1.6).

5.1.2 Online conditional min-entropy estimation

As stated in Definition 3, the physical model of a QRNG will, in general, depend on some set of parameters. In general, the parameters' values for which the claim of ϵ-security holds (for some fixed target ϵ) will only be a proper subset of the set of all values.

Therefore, a QRNG should provide an online method to determine the model parameters' values. For QRNGs using min-entropy extractors, this directly implies the ability for *conditional min-entropy estimation* accessible in runtime. In particular, the possibility to determine if the amount of conditional min-entropy being generated by the QES is sufficient for the extractor to reach the target security parameter ϵ (see e.g. Eq. (22)).

NOTE: Restricted to the two-source case for simplicity, but the definition straightforwardly generalizes to an arbitrary number of sources.

5.1.3 Statistical monitoring

Definition 1 implies that the probability of distinguishing a QRNG's output from a uniformly random (and, independent of the side information) string is bounded above by $(1 + \epsilon)/2$, which, for typical values of $\epsilon \approx 10^{-15}$, implies indistinguishability for all practical purposes.

Nevertheless, statistical testing of the said outcomes should still be in place, to safeguard against an undetected failure either in the QES or in the randomness extraction procedure (the latter being much less likely). When one of these components fail, the generated outputs are likely to be distinguishable from uniform. Notice that if the ϵ-security verification mechanism discussed in the preceding clause is performed on all generated outputs, a QES failure will be detected by it (provided the assumptions on which the physical model relies continue to hold).

Statistical monitoring would generally be conducted at a minimum sampling rate of 1 Hz, with higher rates recommended for high-throughput systems or critical applications.

The monitoring process should include:

a) **Sampling of Raw Output**: The physical model of the QES will determine a set of expected distributions for the raw bits. Computing statistical estimators tailored for such family of distributions might help detect malfunctioning of the entropy source.

At least one sample per second would typically be taken from the raw entropy stream for analysis. The system should support configurable sampling intervals to adapt to performance requirements. More samples over similar periods can be used to customize the confidence intervals based on accuracy requirements. The final min-entropy estimate for the noise source is typically the minimum of the estimates obtained from all applicable statistical tests.

a) **Short-Term Statistical Analysis**: Over rolling windows (e.g. 1-second, 10-second, and 1-minute intervals), compute summary statistics such as:

   i) Mean and variance of bit values

   ii) Frequency counts (bitwise and symbol-wise)

   iii) Probability density estimation

   iv) Unconditional entropy estimation (e.g. unconditional min-entropy)

b) **Anomaly Detection**: Establish acceptable thresholds for statistical parameters. If the monitored metrics fall outside of predefined or configurable bounds (e.g. mean ≠ 0,5 ± tolerance, entropy drops below expected threshold), trigger an exception.

c) Exception Handling and Response: In the event of an anomaly:

   i) **Flag the condition** in the system log.

   ii) **Isolate or pause** the affected output stream if risk to randomness quality is detected.

   iii) **Raise an alert** to connected systems or administrators.

   iv) **Optionally switch to a failsafe entropy source** or previously validated randomness cache, depending on application criticality.

d) **Logging and Audit**: All monitoring data and exceptions should be timestamped, logged securely, and made available for audit and compliance verification.

The Universal Integrated Circuit Card (UICC) specification also present security concepts that may be applicable to QRNGs. The principal UICC standards defined by 3GPP are ETSI TS 131 101 [i.26], ETSI TS 131 102 [i.27] and ETSI TS 131 111 [i.28] which define some security features and mechanisms to protect the UICC and the sensitive data it holds, which contributes to detecting or preventing unauthorized access that could be considered tampering.

Following is a summary of how UICC specifications address security:

- The UICC is implemented as a tamper-resistant device. This means the physical chip is built to resist attempts to physically access or extract its contents. While such implementations do not generally detect tampering, it is a fundamental security measure against hardware-level tampering because it would not be likely to open the device without destroying the ability to extract much meaningful information from it.

- Specifications like ETSI TS 133 110 [i.32] describe mechanisms for establishing a secure channel between the UICC and a terminal. This secure channel is critical for protecting sensitive information exchanged between the two entities. If a terminal attempts to interact with the UICC in an unauthorized or unexpected way, the secure channel's integrity or authentication mechanisms might fail, indicating a potential breach attempt.

- The UICC also manages user credentials and performs cryptographic operations. Access to sensitive applications and data on the UICC is protected by authentication methods, often relying on PINs or other credentials. Unauthorized attempts to bypass these controls could be seen as a form of tampering, and the UICC's security policies could prevent access.

- In the case of embedded UICCs (eUICCs), specifications from GSMA, like SGP.25 [i.33] and SGP.22 [i.34], define the operation of secure remote provisioning of profiles. In particular, these specifications define the implementation of secure communication channels for the eUICC. Attacks that exploit weaknesses in these specifications could be related to tampering, such as memory exhaustion attacks or malicious modifications of management messages. The underlying assumption is that the eUICC should maintain its integrity during these operations.

- The principles set forth by the UICC specifications might be applied to the QRNGs as well. Continuous monitoring ensures that the entropy source maintains its quantum properties and helps detect hardware degradation, tampering, or external interference. These mechanisms are critical to uphold the reliability and trustworthiness of the QRNG.

5.1.4 Shielding and Side-Channel attacks protection

5.1.4.1 Introduction

The quantum source should be adequately shielded from environmental noise (shock, vibration, electromagnetic fields, temperature fluctuations) that could introduce bias or predictability.

5.1.4.2 Environmental and Physical Vulnerabilities in QRNGs

Electromagnetic Interference (EMI)

QRNGs that utilize photonic components, such as laser diodes and photodetectors [i.4], [i.5], can be vulnerable to electromagnetic interference. An out-of-band electromagnetic injection attack can manipulate the QRNG's output, forcing predictable patterns or known sequences, thereby compromising the randomness quality. These attacks may comprise the placement of a Radio Frequency (RF) probe having a signal with variable frequency and amplitude from a signal generator or other RF source.

The effects of such attacks may range from completely disabling the QRNG because of saturating amplification and switching stages of the device but can also be extremely subtle and may not be easily detectable through standard statistical tests [i.6]. A useful countermeasure can be the implementation of a self-monitoring sub-system that disables and logs the relevant information upon detection of the effects of such an attack.

External Magnetic Fields

QRNGs employing components like Faraday Rotator Mirrors (FRMs) in unbalanced Michelson interferometers are sensitive to external magnetic fields. These fields can alter the rotation angle of the FRM, affecting the polarization state of light and, consequently, the interference pattern used for randomness generation. Such deviations can lead to a reduction in the extractable randomness and open potential security loopholes.

Attacks Using Atomic Radiation

Radiation can cause problems for conventional electronics but may also attack the quantum source used in a QRNG, skewing the distribution. In the case of photon-based QRNGs, the source is usually a LED or laser which may be highly sensitive to radiation [i.8], [i.9]. Some of the earliest QRNGs used radioactive decay as a source of entropy, which is inherently complex to radiation harden, as external radiation can interfere with the detection of the decay particles, as external radiation is detected by the internal detector without the ability to discern whether the particle was from the QRNG's own radiation source or something external. Some QRNGs use quantum vacuum fluctuations [i.7] as a source of randomness, and these are less susceptible to radiation than a single photon source or atomic decay approach. It would also be recommended to use detectors and amplifiers specifically designed to differentiate between a genuine event and a radiation-induced false signal, as radiation can introduce false signals as they decaying particles pass through semiconductors.

Optical Injection Attacks

QRNGs based on photodetection can be susceptible to optical injection attacks. Injecting high-energy photons, such as laser pulses, into the photodetector can generate additional photo-carriers, leading to erroneous signals and compromising the integrity of the randomness source. This "blinding" attack can be executed without direct physical contact, making it a significant security concern. A powerful light source such as a flash tube is used to saturate or "blind" the QRNG's single-photon detectors. This can force the detectors into a deterministic or predictable state, where they no longer register the quantum events that are the source of randomness. The attacker can use this knowledge to predict the key and/or disable the key generation depending on system architecture.

Power Analysis/Tampering Attacks

This is not unique to QRNG attacks but is worth mentioning, as this is a fundamental approach to side-channel attack against cryptosystems in general. In the case of a power analysis attack, the attacker monitors and analyses the power consumption of the QRNG device, and in some cases can detect subtle variations that correlate with the random bits being generated. This type of attack usually uses an oscilloscope and/or a logic analyser to either directly measure the voltage drop or current consumption of the entire device, or of certain points in the QRNG subsystem.

On the other hand, a power tampering attack may be effectuated by changing the power supply voltage or regulation of all or part of the QRNG subsystem. Attacks such as this certainly have impact on the operation of various components, such as detectors, etc., but also may have negative impacts on the laser in an optical implementation of a QRNG. If the laser's supply voltage is changed sufficiently that the voltage regulation is unable to compensate, the distribution of the random number output may be corrupted. This is why it is important to ensure that self-monitoring is incorporated into QRNG implementations, where if e.g. the voltage to the laser and other key components is either too far above or below certain threshold limits, the QRNG is disabled, and a warning is issued.

Timing Analysis Attacks

Using an oscilloscope and/or a logic analyser, the voltage and/or current drain of various electronic components is performed. For example, simultaneous monitoring of the voltage on the DSP chip's interrupt pin with the output level of the QRNG may indicate some kind of correspondence between the two relative to the statistics of the random number output distribution, which can provide clues as to how the internals of the QRNG operate. In addition, an attacker can analyse the exact timing of events in the QRNG, such as the time between photon detections. If these timings contain any non-random or exploitable patterns, they might be used to gain a partial or full understanding of the key.

Attacks on Post-Processing

A QRNG's raw quantum entropy data often contains statistical biases that are removed through post-processing algorithms. If the algorithms to remove these biases are poorly implemented, an attacker can exploit them. For example, a weak extraction function could fail to remove all non-random elements, leaving a predictable pattern that can be leveraged to compromise the key. Or, as previously mentioned, an overly aggressive function to remove repeated patterns may destroy the desired output distribution for cryptographic purposes.

5.1.4.3 Mitigation Strategies

To enhance the resilience of QRNGs against environmental and physical interferences, the following measures are recommended, depending on system requirements:

a) **Electromagnetic Shielding**: Implementing shielding techniques to protect sensitive components from EMI can prevent unauthorized manipulation of the QRNG output.

b) **Magnetic Field Monitoring**: Incorporating sensors to detect external magnetic fields can help in identifying potential threats and initiating countermeasures promptly.

c) **Optical Isolation**: Using optical isolators and filters can protect photodetectors from unauthorized light injections, preserving the integrity of the randomness source.

d) **Continuous Monitoring**: Implementing real-time monitoring of the QRNG's output for statistical anomalies can aid in the early detection of potential interferences or attacks.

e) **Physical Security Measures**: Ensuring that QRNG devices are housed in secure environments can reduce the risk of physical tampering or exposure to harmful environmental conditions.

5.1.5 AI Driven Attacks and potential mitigation techniques

Artificial intelligence and machine learning are quickly becoming both the greatest threat and potentially the most powerful defense in cybersecurity [i.30]. Unfortunately, attackers are also leveraging AI. They use it to automate and scale attacks, create more sophisticated malware that can adapt to certain defenses, and craft highly convincing social engineering and phishing campaigns with tools like deepfakes. There already exists a constant "security arms race" between attackers and defenders of secure systems, but the introduction of AI attacks and defense may take this to yet another level, with AI on both sides of the attack/defense spectrum.

AI can be used to attack QRNGs, but not by directly predicting the quantum process itself. The randomness of a QRNG is rooted in the unpredictable nature of quantum mechanics, which is fundamentally immune to classical prediction, even from advanced AI, but the real vulnerabilities of QRNGs lie in their practical implementation and the classical components of the device. This is where AI and machine learning come in.

AI attacks against QRNGs are a form of side-channel attack or predictive analysis. They do not try to break the quantum laws of physics but rather exploit the flow of information between the quantum source and the final output.

Exploiting Classical Noise: Every QRNG system has classical hardware components, like sensors, power supplies, and signal processors. These components introduce small amounts of classical noise, which may be predictable in some cases. If these untrusted noise sources have not been included in the QRNG's physical model (see Definition 3), an attacker with a sophisticated AI model may be able to analyse the output of the QRNG and, over time, learn to identify the predictable patterns caused by it. Then, once the distribution of the classical noise is known, the attacker can now slightly improve its ability to predict the QRNG's outcome, thereby compromising the security of the generated random numbers. Even for an advanced AI, this can be a fairly difficult and time-consuming procedure.

Side-Channel Attacks: These involve an attacker monitoring side channels of the device, such as power consumption, electromagnetic radiation, etc. AI can analyse extremely large amounts of data fairly quickly and potentially find correlations between these side-channel signals and the output of the QRNG. This allows an attacker to gain information about the output without directly attacking the quantum source. For example, a QRNG might show a subtle, non-random power spike when it generates a specific type of number, a pattern that an AI could learn and exploit. This is not unique to QRNG attacks and may be applied in a similar manner to other types of random number generators as well.

To counter these threats, developers of QRNGs are focusing on a few key strategies:

- **Robustness against Classical Noise:** By improving the hardware design and shielding of QRNG devices, manufacturers can minimize the impact of classical noise.
- **Advanced Randomness Extraction:** Researchers are developing more sophisticated and verifiable randomness extraction algorithms.
- **Continuous Monitoring:** As previously discussed in clause 5.1.4.3, continuous monitoring systems that constantly check the QRNG's output for even the slightest biases or correlations may allow for real-time adjustments or alerts.

5.1.6 Entropy Zero Trust (EZT) - ETSI EZT Profile for QRNG Security

To address evolving post-quantum threats and entropy forgery scenarios, QRNGs deployed in regulated or high-security domains should implement the Entropy Zero Trust (EZT) model, which incorporates layered assurance beyond the entropy source.

A QRNG system compliant with EZT should:

a) **Layer 1 - Provable Quantum Entropy Source**: Entropy should derive from quantum phenomena with verifiable physics (e.g. photon path, vacuum fluctuation) and source-level attestation.

b) **Layer 2 - Continuous Health Monitoring**: Entropy output should be monitored in real time (e.g. min 1 Hz) for drift, bias, or failure, with automatic logging, isolation, and fallback.

c) **Layer 3 - Hardware Root of Trust**: The platform should include TPM/eFuse/HWRoT for cryptographic attestation, secure boot, and anti-rollback firmware protections.

d) **Layer 4 - Secure Integration Path**: Output should traverse authenticated and auditable channels (e.g. PCIe/AXI DMA with ACLs, hardened drivers, container interfaces).

e) **Layer 5 - Virtualized Entropy Pool Management**: Multitenant environments should support logical QRNG instances with independent logging, error boundaries, and memory isolation.

This model is aligned with zero trust cryptographic architecture's best practices.

5.1.7 Entropy Provenance and Usability Assurance

To ensure that the entropy output of a QRNG is both *quantum-origin* and *usable for critical applications*, the following layered guarantees should be enforced:

a) **Source-Specific Cryptographic Proof-of-Origin**: Entropy values should be tagged or traceable back to an attested quantum source, using TPM-bound cryptographic signatures or similar hardware root-of-trust.

b) **Run-Time Provenance Audit**: Devices should log entropy generation metadata (e.g. timestamp, firmware ID, entropy pipeline identifier) for each output block or session.

c) **Entropy Usability Interface**: Entropy output should be provided via hardened interface layers that include secure memory boundaries, transport authentication, and usage binding. This ensures that the quantum entropy is not only present, but functionally available to cryptographic systems without tampering or downgrade attacks.

5.2 Security of Implementation

5.2.1 Tamper Resistance

The QRNG device should be designed to be tamper-evident and possibly have continuous self-diagnostics, as per clause 5.1.4.3 above, so that any attempts to manipulate it are detectable and that appropriate countermeasures may be taken.

5.3 Classification of QRNGs

5.3.1 What to classify

The objective of classification is to provide both the manufacturer and user with a common way of thinking about the domain of quantum random number generators along with their intended application. There are characteristics that can easily be classified, such as throughput rates and power consumption, but also size in terms of device volume and weight of the device. This is important when considering the construction of a certain type of product, [i.31].

Consider the requirements for building a practical surveillance drone, having the ability to acquire and send images while in operation. A key requirement for such a product might be the security of the signalling channels in both uplink and downlink direction, making if difficult for an intruder to take control of the device remotely by reverse-engineering the signalling channels and capturing the drone's receiver using higher-power transmission. A solution for such a signalling channel might be a non-reusable one-time pad generated by a QRNG, so that even if an attacker can transmit information on the drone's uplink receive channel, the drone would not receive meaningful information, as the attacker would not have the one-time pad. In such an implementation, the QRNG that generates the one-time pad might reside in the drone itself and could share the key with the ground station control device using a hard-wired interface, such as USB-C, before flight.

In this case, the requirements of the QRNG should be consistent with the physical size and weight constraints, power requirements and also throughput requirements of the system. For example, the maximum volume of a QRNG in this case might be 100 mm^2 with a maximum weight of 10 grams, maximum power dissipation of 100 mW and minimum throughput of 10 kb/second.

Another type of product might be used by a cryptographic certificate authority, with the requirement that the system generates several million certificates per minute. In this case, the throughput of the QRNG should be such that the minimum number of certificates can indeed be generated within the allocated time period, but power consumption and size constraints may be much relaxed compared to the drone example.

The following guidelines are proposed:

Table 1

Trust Level	Definition	Minimum Requirements
TL-0	Component-level entropy source	No entropy provenance, no attestation, raw data access only
TL-1	Secure QRNG subsystem	Device includes self-tests, but does **not guarantee** runtime quantum-origin traceability
TL-2	EZT-compliant entropy platform	Meets criteria in clause 5.1.4.3. Includes cryptographic attestation, monitored entropy integrity, and hardened interface compliance
TL-3	Military/Critical infrastructure grade	Includes dual entropy sources with appropriate certification, secure boot with quantum entropy sealed identity

5.3.2 Throughput

Various techniques for implementing QRNGs mean that the rate of number generation may be drastically different between implementations. Because the QRNG is designed to meet the required throughput requirements for the intended application, it may be desirable to describe QRNG capabilities as having a certain "throughput class". This provides manufacturers and/or users of QRNG devices a reasonable basis for comparison, and a more-or-less "common language" for the discussion of QRNG types at a high-level.

Examples of throughput classes follow:

Table 2

Class	Throughput Range	Typical Use Cases
Class I	≤ 100 kbps	Embedded systems, low-rate IoT, sensor entropy sources
Class II	100 kbps - 10 Mbps	Key generation for cryptographic libraries, mobile security modules
Class III	10 Mbps - 100 Mbps	Secure communication appliances, high-assurance key generation
Class IV	100 Mbps - 1 Gbps	Enterprise HSMs, VPN appliances, bulk key generation
Class V	≥ 1 Gbps	High-performance computing, cloud data centers, real-time QKD systems

5.3.3 Power Consumption

Another criterion may be the implementation of power consumption classes, providing another means of comparison between devices and their applications. For example:

Table 3

Class	Power Range	Typical Deployment
Class A	≤ 100 mW	Low-power QRNGs for embedded cryptographic coprocessors and PCB/modules
Class B	100 mW-500 mW	Discrete QRNG modules in mobile devices or lightweight security appliances
Class C	500 mW - 2 W	PCIe cards, USB-attached QRNGs, or compact modules with post-processing hardware
Class D	2 W - 10 W	Rack-mounted entropy appliances or systems with integrated DRBGs and shielding electronics
Class E	> 10 W	Unlikely in realistic QRNG deployments; not typical or recommended

5.3.4 Volume

QRNG solutions should specify volume constraints for target environments, particularly in embedded and mobile systems. Standardized classifications could be as follows:

a) Class S: < 100 mm^3 (e.g. M.2 modules for embedded systems)

b) Class M: 100 mm^3 - 1 000 mm^3 (compact modules, PCIe)

c) Class L: > 1 000 mm^3 (rack-mount or appliance-based systems).

5.3.5 Weight

QRNGs intended for deployment in airborne, space, or mobile platforms should define operational weight constraints. Suggested classification:

a) Light: ≤ 20 g

b) Medium: 20 g - 200 g

c) Heavy: > 200 g

The classification provides a practical basis for integration and compliance with Size-Weight-Power (SWaP) constraints in critical systems.

5.3.6 Interface Specifications

The commercial implementation of QRNGs is still relatively new at this date, to the point that manufacturers tend to use proprietary interfaces. This fact severely limits interoperability among manufacturers and even between product lines within a single manufacturer, in addition to requiring expensive and time-consuming activities for manufacturers in order to protect their supply chain. For example, in a case of a manufacturer who can use multiple chip vendors having common interfaces, the product manufacturer benefits by having multiple sources of chips having common interfaces, but also the chip vendors benefit because adoption is likely to increase at volume for those with common interfaces.

Some of the items that require consideration when defining a common interface standard are as follows:

Application Programming Interface (API) interfaces

These types of interfaces usually take the form of software libraries, that allow programmers to perform operations such as setting parameters, extracting certain amounts of random information, setting up monitoring, enabling/disabling logging and/or the entire device, etc. A typical API would support function calls, with or without parameters and would return usually a success or failure indication and optionally fill in the address of a variable with the desired information. It is important that any such library is implemented in such a way so as to prevent side channel attacks on the QRNG, entropy source and/or conditioning routines by, e.g. sending unknown parameters, out of range parameters, running addressing pointers past boundaries, etc.

Message-driven interfaces

A message-driven interface may be implemented across an opaque data channel, which could be a secure network connection, data bus or other such bi-directional channel. Typically, the device could be controlled by sending an op-code with optional parameter(s) across the channel. The device would then optionally send back a response message containing a success or failure code and possibly the string of random information requested. A message-driven interface may also be set up to be interrupt driven, operating the device asynchronously, where the device raises an interrupt pin bound to a function to call to process the information.

Programmability of interfaces and operational modes

It is common for hardware devices to be programmable, and for the QRNG it is envisioned that it would be possible to set the device up for a certain mode of operation, e.g. free-running, where the device continues to output a stream of bits at a certain data rate until it is told to stop, or one-shot, where the device is told to output a certain number of random bits or octets and then stops until it receives another command.

The following represents a potential set of operational commands (op-codes) and modes:

1) Enable device: device would not perform any operations until this command is received.
2) Disable device: device would stop performing any operations until re-enabled.
3) Reset device: device would perform an internal reset, returning to a predictable internal state.
4) Enable continuous monitoring: enables continuous monitoring of probability density of output data.
5) Define default actions of continuous monitoring: could range from no action to sending special message to alert system operators that the distribution of output data is at risk of being predictable, plus automatic self-destruct if a certain level of predictability is noted.
6) Destroy device: self-destruct to prevent reverse-engineering if tampering is detected.
7) Set logging level: sets the verbosity of device state logging, if needed, where a default logging level of zero could also mean that the device does not send any logging information.
8) Set logging channel: tells the device where to send its logging information.
9) Set output channel: this could set the address of a variable at which output information is written, and the boundaries of the address space. It could also set a device address, such as a physical address to which output data are sent.
10) Set output data rate: sets the speed at which the random bit stream is delivered.
11) Set synchronous (one-shot) mode and the number of desired bit would appear at the output.
12) Set asynchronous (free-running) mode: the device would continuously deliver the maximum number of output bits continuously.
13) Set absolute maximum output length: would set the maximum number of random bits to output.

The present document just presents possibilities for op-codes and modes, but the actual standards would be defined in a Technical Specification (TS) which is a normative standard. Much more coordination and discussion among the industry is required to help decide which op-codes and modes would be mandatory or optional. It is certainly possible to standardize all four of the above types of interfaces, and even to support all interface standards by a single manufacturer, which can provide versatility.

5.3.7 Scalability

It is important to consider the scalability of the QRNG design to meet potential future demands. For example, a firm who generates cryptographic certificates may find it challenging to continue to meet the demand for cryptographic keys as its customer base grows. With scalability also comes the notion of extensibility, which is the ability for the architecture to anticipate the potential scaling of the services.

For example, consider how straightforward it is to add another QRNG module or group of modules to a system. If a system is intended to be scalable, an extensible architecture should be constructed to allow for rapid and relatively risk-free update of systems for adding additional QRNG capacity, and each additional QRNG module should not impact the existing ones already deployed. The notion of redundancy is also a possibility, where a QRNG in group of QRNGs in a module may automatically be replaced in the event of self-destruction to prevent tampering.

5.4 Compliance and Certification

5.4.1 Industry Standards

QRNG development and deployment should adhere to relevant industry standards and best practices to ensure security, reliability, and interoperability. These standards serve as benchmarks for design assurance, entropy quality, cryptographic integration, and compliance validation.

Key references include:

a) **FIPS 140-3 [i.35]**
Defines the security requirements for cryptographic modules, including physical security, tamper resistance, and entropy source validation. QRNGs integrated into cryptographic systems should align with the applicable security level (typically Level 3 or 4 for tamper resistance).

b) **NIST SP 800-90 Series**:

i) SP 800-90A: Approved Deterministic Random Bit Generators (DRBGs) [i.20]

ii) SP 800-90B: Recommendations for entropy sources used in random bit generation (critical for QRNG validation) [i.22], [i.23]

iii) SP 800-90C: Construction and combination of entropy sources and DRBGs [i.24].

c) BSI AIS 20/31 [i.36]

Evaluation criteria for different functionality classes comprising deterministic and true random number generators.

d) ISO/IEC 19790 [i.37], ISO/IEC 18031 [i.21**] and ISO/IEC 24759 [i.38]**

International equivalents and test requirements related to FIPS 140-3 [i.35], commonly used in jurisdictions outside the United States.

e) ISO/IEC 20543 [i.39]

Test and analysis methods for random bit generators within ISO/IEC 19790 [i.37] and ISO/IEC 15408-1 [i.40].

f) **ETSI GS QKD 014 [i.41]**
Though focused on QKD systems, this document establishes the guidelines for exchanging keys using quantum communication methods and specifies how data should be formatted and transmitted within a QKD network.

g) Recommendation ITU-T X.1702 [i.42]

h) **IETF RFC 4086 [i.43]**
Offers practical guidance for cryptographic applications that consume randomness, including entropy pool handling and mixing functions.

In addition to compliance with these standards, QRNG designers should implement lifecycle practices aligned with **NIST SP 800-160** [i.44] (Systems Security Engineering) to ensure long-term trustworthiness and resilience against evolving threats.

The development of QRNG systems should also consider adherence to any sector-specific regulations (e.g. health, finance, defense) where certified cryptographic strength and verified entropy sources are mandated.

5.4.2 Certifications

To ensure trustworthiness, compliance, and broad market acceptance, it is critical that QRNG systems obtain certifications from reputable and recognized authorities. These certifications validate that the QRNG's entropy source, processing methods, physical implementation, and integration practices conform to established security and performance standards.

Key Certification Pathways:

a) **FIPS 140-3 [i.35] Certification**
Issued by the U.S. National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS), this certification is required for cryptographic modules used in U.S. federal systems. QRNGs integrated into such modules should comply with the physical security, entropy, and cryptographic requirements outlined in FIPS 140-3 [i.35].

b) **NIST Entropy Source Validation (per SP 800-90B [i.23])**
For QRNGs used to feed Deterministic Random Bit Generators (DRBGs), entropy source validation is critical. Certification involves rigorous testing of entropy generation, statistical quality, and robustness against failure or manipulation.

c) **BSI AIS 20/31 [i.36]**
An evaluation framework for deterministic and true random number generators by the German Bundesamt für Sicherheit in der Informationstechnik (BSI).

d) **ISO/IEC 19790 [i.37] & ISO/IEC 24759 [i.38] Compliance**
International equivalents to FIPS 140-3 [i.35], used for evaluating cryptographic modules outside of North America. Ensures global interoperability and recognition.

e) **ETSI & ITU-T Compliance**
Emerging certifications from bodies such as ETSI (e.g. ETSI GS QKD 014 [i.41]) and ITU-T (e.g. ITU-T Y.3800 series [i.45] for quantum technologies) may be applicable to QRNGs, particularly those used in QKD or telecom applications.

f) **National and Regional Cybersecurity Agencies**
In some jurisdictions, additional certifications may be required by national standards agencies (e.g. ANSSI in France, CCN in Spain) for use in classified or regulated environments.

Other Assurance Practices:

i) **Third-Party Penetration Testing**:
Independent evaluation of tamper resistance, side-channel resilience, and environmental robustness.

ii) **Independent Lab Testing**:
Certification bodies may require or accept entropy and randomness testing performed by ISO/IEC 17025 [i.46]-accredited laboratories.

iii) **Compliance Labels and Seals**:
QRNG vendors may also seek recognition through industry alliances and compliance programs (e.g. TCG, PSA Certified, or EU Cybersecurity Certification Framework).

6 Conclusions

Quantum Random Number Generators (QRNGs) represent a critical building block in quantum-resilient and post-quantum cryptographic infrastructures. To effectively transition from classical to **provably quantum entropy** platforms, implementers should go beyond mere claims of quantum sourcing and instead establish layered, verifiable, and auditable implementations.

The present document outlines a set of implementation guidelines that distinguish genuine QRNGs from enhanced TRNGs or entropy emulators, based on Entropy Zero Trust (EZT) principles, cryptographic attestation mechanisms, and real-time entropy monitoring on which to build reasonable implementation plans.

It is essential that QRNG vendors and integrators avoid overstated claims of "true randomness" without proof of isolation, origin validation, and entropy usability in critical systems. Procurement and evaluation frameworks should begin to demand provable provenance, runtime attestation, and interface integrity.

Future standardization efforts should prioritize:

a) ϵ-security certification models beyond current statistical testing (e.g. runtime EZT compliance).

b) Mandatory logging and attestation protocols for entropy source trust.

c) Integration guidelines for hybrid PQC + QRNG systems.

Annex A:
QRNGs - the current state of the art

A.1 Photon-Based QRNGs

Photon-based QRNGs are one of the most commonly implemented and studied forms of quantum random number generators. These systems use the inherent randomness of quantum phenomena like photon polarization or photon paths through a beam splitter to generate truly random numbers. These are the main characteristics of this type of this QRNG technology:

a) In a typical photon-based QRNG, a light source sends photons through a beam splitter, where each photon has an equal probability of being transmitted or reflected. Detectors are placed to capture the photon's path, with the outcome being mapped to a binary 0 or 1. The randomness arises from the fact that the behaviour of individual photons at the quantum level is unpredictable.

b) Early photon-based QRNG systems were limited by their relatively low throughput, but recent developments have increased the speed of these devices significantly. Researchers have improved the detection and data acquisition systems, allowing for bit rates of up to hundreds of megabits per second. High-throughput photon-based QRNGs are now suitable for real-time cryptographic applications where secure keys or random numbers are generated at high speeds.

c) One of the biggest advances in photon-based QRNGs has been the development of miniaturized, chip-scale QRNG devices. These devices integrate the quantum optics and detectors onto a single chip, making them small enough to be embedded in mobile devices [i.3], laptops, and Internet of Things (IoT) hardware. The miniaturization of QRNGs increases their accessibility for consumer applications, allowing devices like smartphones to generate true random numbers for secure communications or mobile payment transactions.

A.2 Quantum Vacuum Fluctuation-Based QRNGs

A QRNG based on quantum vacuum fluctuations generates random numbers by measuring the intrinsic, unpredictable energy fluctuations of a vacuum. According to quantum mechanics, even an empty space like a vacuum is not really empty. The space is filled with virtual particles and fields that constantly appear and disappear. This is a fundamental phenomenon guaranteed by the Heisenberg Uncertainty Principle.

The process of generating randomness based on vacuum fluctuations typically uses a technique called homodyne detection to measure these fluctuations. Homodyne detection is a technique used in quantum optics to measure the phase and amplitude of quantum states, particularly coherent states of light. This method can be effectively utilized in vacuum-based quantum random number generators to produce truly random numbers. A coherent state of light is characterized by a well-defined phase and amplitude. When using homodyne detection, two coherent states are involved: the quantum signal state from a vacuum state and a Local Oscillator (LO) state that serves as a reference. The quantum signal and the local oscillator are combined at a beam splitter. This creates interference between the two fields, allowing measurement of the relative phase and amplitude of the quantum state.

The homodyne detection setup involves two photodetectors positioned to measure the output of the beam splitter. The difference in the detection results from both photodetectors gives information about the quantum state. The two beams interfere with each other, and the resulting interference pattern is measured by a balanced photodetector that subtracts the signals from two photodiodes, effectively cancelling out the noise from the laser itself and isolating the small, quantum-induced fluctuations. The output from the photodetector is a continuous analog signal representing the measured quantum fluctuations. This signal, which is intrinsically random, is then digitized using an Analog-to-Digital Converter (ADC). The raw digital data are then subjected to post-processing algorithms to remove any residual biases or non-randomness caused by classical noise, like thermal noise from the electronics, and to ensure a statistically uniform output.

A.3 Entanglement-Based QRNGs

Entanglement-based QRNGs exploit the phenomenon of quantum entanglement, where two or more particles are linked in such a way that the measurement of one instantaneously affects the state of the other, no matter the distance between them. The main characteristics of Entanglement-based QRNGs are presented below.

a) The outcomes of judiciously chosen measurements over distinct parts of an entangled system exhibit correlations that no local hidden variable theory can reproduce. These correlations are termed *nonlocal*.

b) In an entanglement-based QRNG (also known as device-independent QRNG in the literature), the observation of these nonlocal correlations allows for the quantification of the amount of conditional min-entropy in their raw outputs, solely from the assumption of independence between the measurements of the distinct parts of the composite system.

c) Entanglement-based QRNGs are still largely experimental but are an exciting area of research because of the reduction in the number of assumptions that back their physical model.

d) One challenge with entanglement-based QRNGs is verifying the entanglement and scaling the system for practical cryptographic use. Ongoing research is focused on building more reliable entanglement sources and improving the scalability of these systems so they can generate random numbers at rates suitable for real-world applications.

A.4 Physical TRNGs vs "QRNGs"

Some industry experts claim that a QRNG is really a type of TRNG; the only difference would be that a QRNG uses a quantum entropy source. Some devices marketed as "quantum" RNGs rely on amplified classical chaos, semiconductor instabilities, or thermal noise, none of which are quantum phenomena. While such sources may contain stochastic contributions, they may not meet the minimum criteria for certified quantum entropy unless the randomness contribution is:

a) measurable;

b) isolated from classical noise;

c) attested through quantum integrity protocols.

Manufacturers and integrators should avoid confusing classical physics-based TRNGs with true QRNGs unless independently validated. Implementers and buyers should request proof of quantum entropy validation in procurement processes. Reference implementation tests or certifications should accompany claims of QRNG status.

A.5 PQC + QRNG Architectures

The highest levels of cryptographic assurance are achieved when systems integrate:

a) Provably quantum entropy from certified QRNGs (see clause 4.2),

b) Post-Quantum Cryptographic (PQC) algorithms that are not reliant on QKD,

c) Runtime attestation of entropy generation and consumption (see clause 5.1.3),

d) Scalable, modular deployment.

Table A.1

Element	PQC-only	PQC + QRNG
Entropy Integrity	Software-seeded	Certified Quantum Source
Quantum Threat Resilience	Partial	Strong
Supply Chain Tamper Resilience	Low	Medium-High (with attestation)

These layered approach help enforce *zero-trust entropy processing pipelines*, distinguishing QRNG platforms from simple entropy sources.

Architectures that combine PQC and QRNG should define not only the entropy source but also its **integration domain**. Examples include:

a) QRNG-backed secure boot with sealed firmware hashes

b) TPM-fused entropy measurement for per-device identity keys

c) Certificate authorities with QRNG-seeded keypair derivation

A key requirement in hybrid PQC + QRNG architectures is the elimination of any deterministic fallback or seed leakage that would allow an attacker to predict key generation even with post-quantum computational resources.

Zero-trust entropy pipelines should assume all intermediate components (CPU, memory, OS) are potentially compromisable and thus enforce **hardware-rooted isolation**, entropy path cryptographic signing, and multi-tenant logical separation when applicable (e.g. cloud or HSM deployments).

Annex B:
EZT Implementation Blueprint

This annex provides a non-binding checklist and architecture suggestion for vendors and integrators aiming to implement EZT-compliant QRNG systems.

Minimum Capabilities:

a) Quantum entropy source attested by physics-backed measurement (see clause 5.1.3)

b) Secure boot with entropy sealing (TPM/eFuse required)

c) Health monitoring module with statistical deviation detection (1 Hz min)

d) Authenticated entropy interface (PCIe, DMA with ACL)

e) Entropy provenance log tied to entropy pool state

Implementation Example (abstract):

i) FPGA or SoC controls QES and randomness extractor based on two-universal hashing (e.g. Toeplitz hashing) and entropy monitor.

ii) Hardware module signs entropy packets with internal key

iii) OS driver exposes /dev/quantum_rng with ioctl-based access

iv) Entropy is injected into system entropy pool or HSM entropy interface

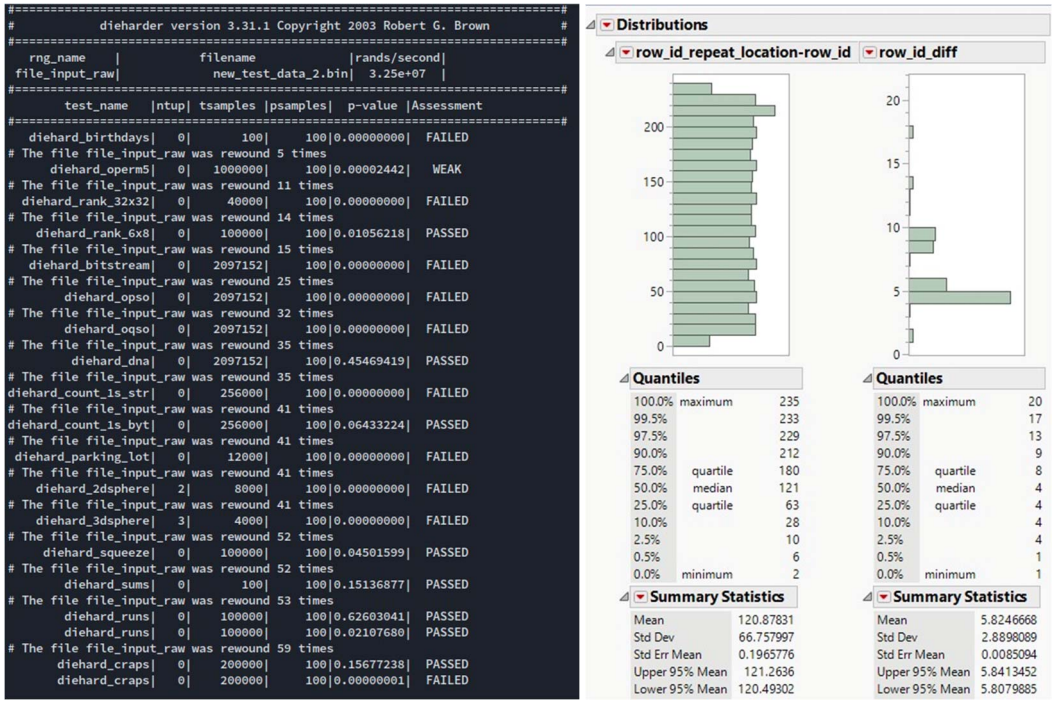
This annex does not prescribe specific vendors or implementations but rather supports ETSI members and readers in achieving an auditable level of zero-trust entropy generation.

Annex C:
Filtering options to address statistical bias

C.0 Example of addressing bias in a QES intended for stochastic simulations

The output of certain QESs was identified as not being entirely uniform and possessing a systematic bias that hindered its scoring and compliance against several randomness tests. After testing several solution options, the optimal path was determined to be the use of a probabilistic data structure and logic implemented after the QES output stage, which would filter and post-process the raw output to make it compliant and errorless. This logic also serves as a tool for self-certification and validation of the quality and health status of the unit.

The randomness output from the QES suite demonstrated systematic bias during tests with Diehard, using both Accelerated and Non-Accelerated hardware in direct mode (without system entropy enhancement). The issue manifests as frequent failures in several randomness tests, indicating non-random behaviour. To further investigate, a custom utility was developed to extract 128-bit phrases from the QES's output. Each phrase was added sequentially to an online filter that tracked repetitions and recorded their positions. A sample of 27×106 128-bit phrases was analysed to detect potential patterns or irregularities in the output.



```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|            new_test_data_2.bin|  3.25e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 5 times
      diehard_operm5|   0|   1000000|     100|0.00002442|   WEAK
# The file file_input_raw was rewound 11 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 14 times
    diehard_rank_6x8|   0|    100000|     100|0.01056218|  PASSED
# The file file_input_raw was rewound 15 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 25 times
        diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 32 times
        diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 35 times
         diehard_dna|   0|   2097152|     100|0.45469419|  PASSED
# The file file_input_raw was rewound 35 times
diehard_count_1s_str|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 41 times
diehard_count_1s_byt|   0|    256000|     100|0.06433224|  PASSED
# The file file_input_raw was rewound 41 times
 diehard_parking_lot|   0|     12000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 41 times
    diehard_2dsphere|   2|      8000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 41 times
    diehard_3dsphere|   3|      4000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 52 times
     diehard_squeeze|   0|    100000|     100|0.04501599|  PASSED
# The file file_input_raw was rewound 52 times
        diehard_sums|   0|       100|     100|0.15136877|  PASSED
# The file file_input_raw was rewound 53 times
        diehard_runs|   0|    100000|     100|0.62603041|  PASSED
        diehard_runs|   0|    100000|     100|0.02107680|  PASSED
# The file file_input_raw was rewound 59 times
      diehard_craps|   0|    200000|     100|0.15677238|  PASSED
      diehard_craps|   0|    200000|     100|0.00000001|  FAILED
```


Figure C.1: Diehard test result on raw QRNG output (left) and Distribution of phrase repetition (right)

C.1 Implications of Findings

Diehard test failures: Figure C.1 shows the results of the Diehard test on the raw QES output. In this case, many randomness failures were observed.

Phrase repetitions: Repeated phrases were observed after fewer than 1 000 rows, with subsequent repetitions occurring at intervals of 40 to 240 rows. This pattern suggests an internal glitch within the QRNG itself, potentially related to an overflow condition in an 8-bit register. The mean repetition rate was calculated as 1 in every 5,5 phrases, corresponding to a repetition factor of approximately 20 %. Such a high repetition rate is well beyond acceptable limits for secure randomness, adversely affecting the quality of the RNG output.

The findings reveal a critical flaw in the QES output, making it unsuitable for secure RNG applications without additional post-processing. Note also that the experimental method used to identify these issues is not viable for production environments due to its high resource demands and incompatibility with continuous data flow typical of QESs in operation.

Exploration of potential solutions

To address the identified problem and improve the quality of randomness, a hardware-based solution leveraging Bloom Filters (BF) was developed. This approach focuses on enhancing randomness quality by efficiently detecting and filtering repeated patterns while maintaining system throughput and integrity.

Bloom Filters are highly efficient, probabilistic data structures designed to determine whether an element belongs to a set. They excel in scenarios where quick lookups and minimal memory usage are critical, as they use a bit array and multiple hash functions to represent the presence of elements. The key space-time advantages of Bloom Filters make them ideal for real-time hardware applications:

a) Space Efficiency: Bloom Filters offer a compact representation of sets, significantly reducing memory requirements compared to traditional data structures like hash tables. This space efficiency makes them suitable for hardware implementations where memory resources are limited, such as FPGA-based systems.

b) Time Efficiency: Bloom Filters perform membership checks in constant time, specifically O(k), where k is the number of hash functions used. This constant-time complexity ensures that queries and insertions can be performed rapidly, making them ideal for high-frequency data streams that require real-time processing.

c) Trade-offs: While Bloom Filters are highly efficient, they trade off memory savings for the possibility of Type I errors (false positives). This means they may incorrectly indicate that an element is present when it is not. However, they guarantee no false negatives, meaning if the filter says an element is absent, it is definitely not in the set. This probabilistic nature makes Bloom Filters especially useful for scenarios where absolute accuracy is less critical than speed and space, such as in filtering repeated patterns in randomness data.

Given these advantages, Bloom Filters are well-suited to operate as real-time filters in the PQU units, detecting and eliminating repetitions from the randomness output with minimal resource consumption and latency. Several alternative solutions were explored, as follow:

Option 1: Basic Bloom Filter (BF)

A standard Bloom Filter uses multiple hash functions to determine the positions in a bit array corresponding to a given element. If all positions are set to 1, the element is considered part of the set. This mechanism is simple and efficient, making it an ideal candidate for hardware implementation within the practical limitations of certain product deployments.

Strengths: Space-efficient and fast, providing real-time checks with minimal resource consumption. Guarantees no false negatives, which is critical for filtering out repeated patterns in the randomness output.

Option 2: Element-based Sliding Bloom Filter (EBF)

An Element-based Sliding Bloom Filter (EBF) extends the basic Bloom Filter by associating each bit in the array with a timestamp or counter. This additional information allows the filter to track when bits were set, introducing a temporal dimension to the filtering process. During a query, if all relevant bits are set and their associated timestamps fall within a valid time window, the filter considers the element as present.

Strengths: Adds the ability to distinguish between recent and older elements, improving the filter's effectiveness in high-frequency data streams where repeated patterns could reappear after a short period.

Option 3: Partitioned Sliding Bloom Filter (PBF)

A Partitioned Sliding Bloom Filter (PBF) operates by dividing the bit array into multiple partitions. Each partition functions as a distinct Bloom Filter, and when an element is inserted, it is added to the currently active partition. Queries are performed by combining all partitions using a bitwise OR operation. Over time, the active partition is switched to the next one, while older partitions are cleared, effectively removing older elements.

Strengths: Provides a systematic and scalable method for managing older elements, ensuring that older patterns do not accumulate indefinitely. The partitioned approach allows for coarse-grained control over the age of elements retained in the filter.

Implementation Considerations: The size of each partition and the number of partitions should be chosen carefully to align with the expected data throughput and analysis window. The window / partitions should exceed the actual analysis window to ensure accurate filtering.

Experimental determination of most appropriate filter implementation

To identify the best solution for filtering repeated patterns in the QES output, experiments were conducted on three Bloom Filter variants, focusing on accuracy and system efficiency.

The filters were configured as follows:

i) Bloom Filter (BF). 6 230 000-bit array, 5 hash functions

ii) Element-based Sliding Bloom Filter (EBF). 10 000-bit array, 5 hash functions, 800-element window, 8-bit counters

iii) Partitioned Sliding Bloom Filter (PBF). 10 000-bit array, 5 hash functions, 1 000-element window, 4 partitions

Performance Summary

The experimental results highlighted key differences in accuracy, where PBF offers Superior accuracy, consistently above 0,927, thanks to its partitioned approach that efficiently handles older data. Recall and F1 Score to select the optimal parameters, as Recall prioritizes minimizing false negatives, which is crucial given the heavier consequences of missing instances in the positive class (i.e. 'repeated' patterns). The F1 Score provides a balanced measure by considering both false negatives and false positives, ensuring overall classification effectiveness.

A set of 12 experiments were run using different Window element sizes as well as array bit sizes, with the objective of testing different configurations that are deemed feasible to be implemented as well as close to the theoretical bloom filter optimal points.

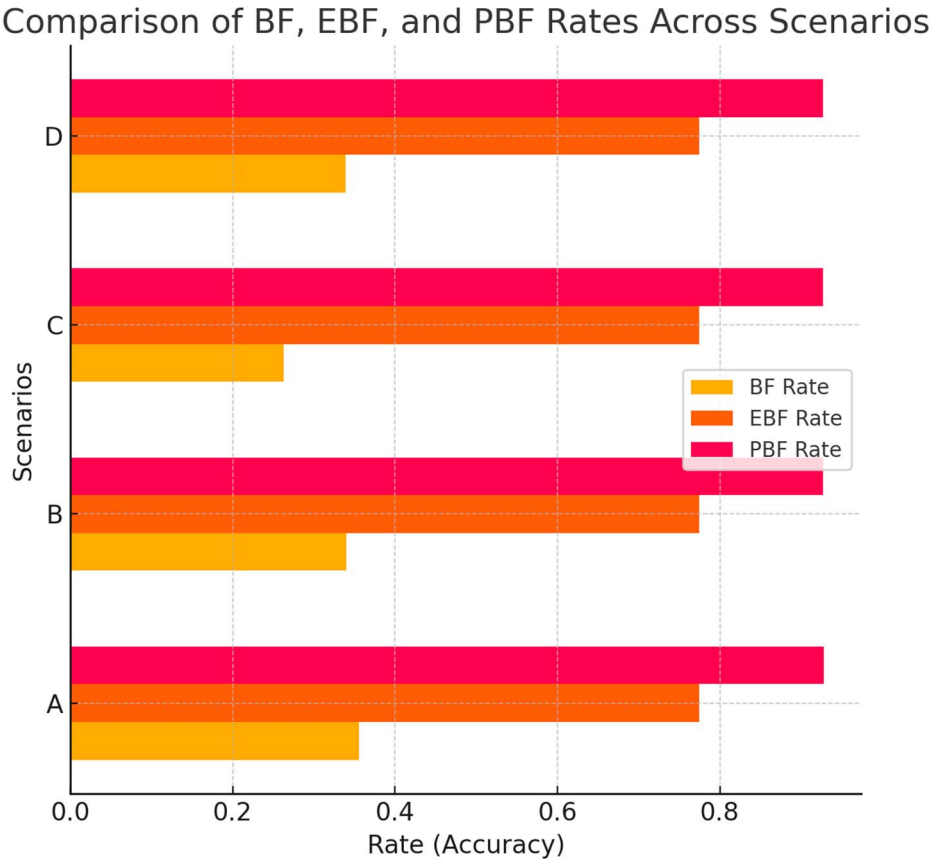


Figure C.2: Classification accuracy comparison of base filter options

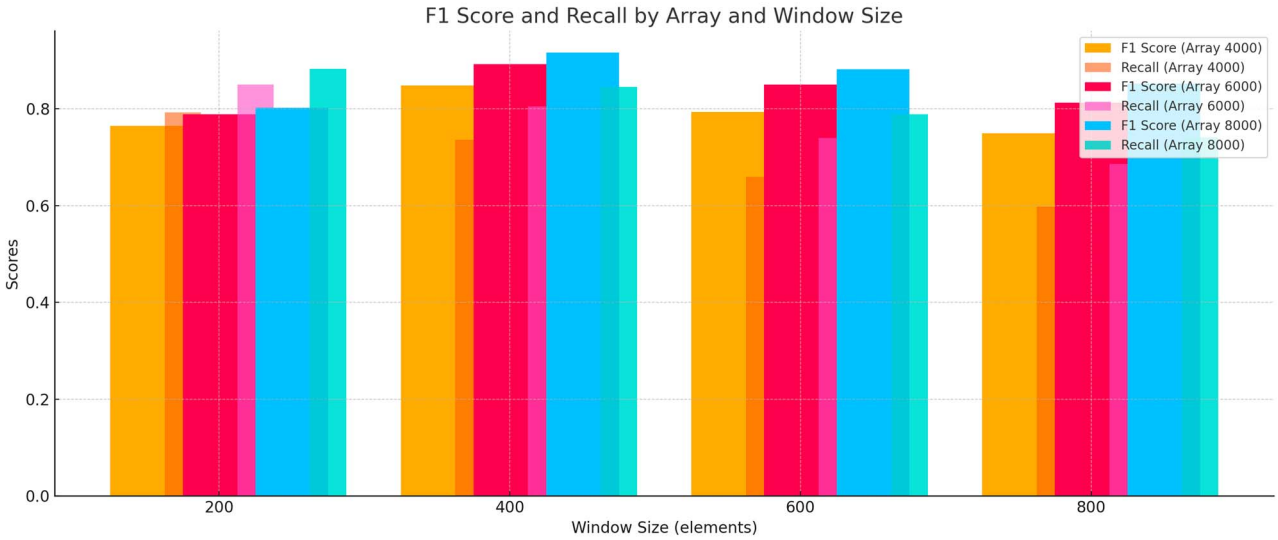


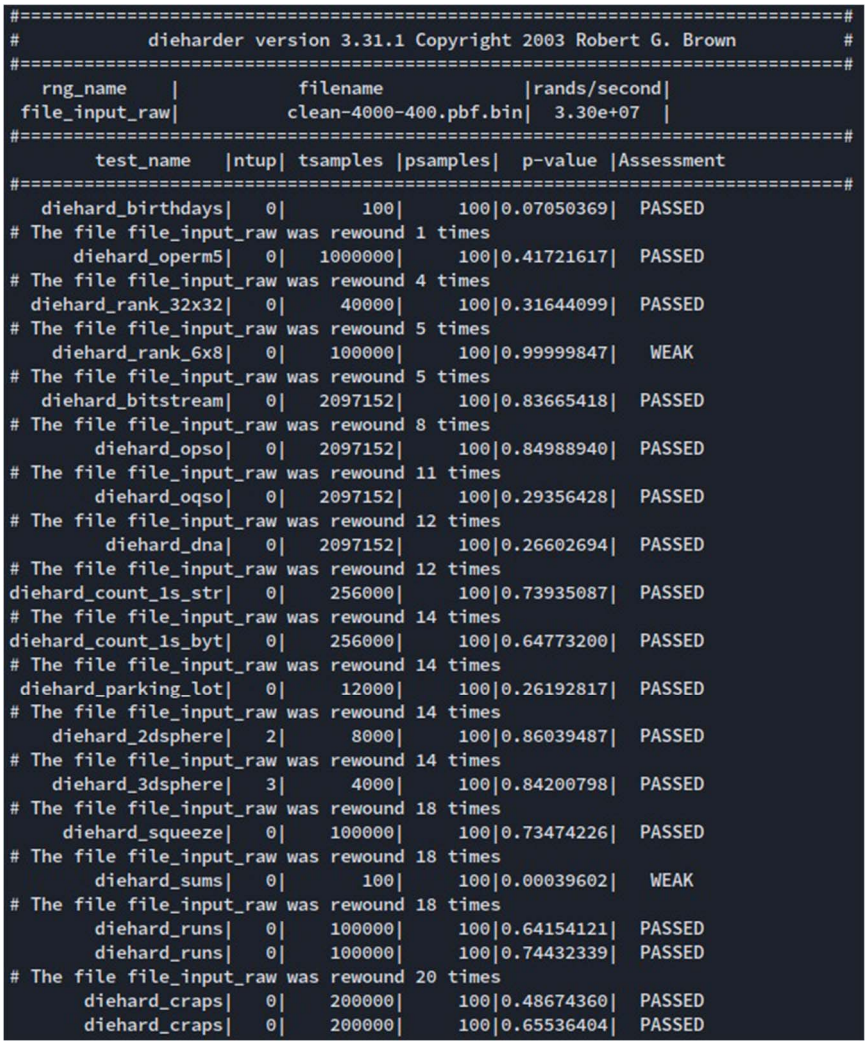
Figure C.3: PBF classification performance comparison of different parameters

According to the experimental results, the optimal parameters are a window size of 400 elements and an array size of 8 000 bits. These parameters provided the highest combined Recall and F1 Score, with a Recall of 0,845 and an F1 Score of 0,916, making this configuration the best for detecting repeated patterns while maintaining classification accuracy.

Additionally, the False-Positive Rate (FPR) for this configuration is approximately 3,8 %, which is considered excellent for probabilistic filters. False-negative rate is 0 %.

Diehard Tests on Post-Processed QES Output

Diehard tests were performed on a sample of random output obtained from a QES that was post-processed using the optimal Partitioned Sliding Bloom Filter (PBF) configuration (8 000-bit array, 400-element window). The goal of this test was to evaluate the effectiveness of the PBF in improving the uniformity of the QES output.



```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|          clean-4000-400.pbf.bin|  3.30e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.07050369|  PASSED
# The file file_input_raw was rewound 1 times
      diehard_operm5|   0|   1000000|     100|0.41721617|  PASSED
# The file file_input_raw was rewound 4 times
  diehard_rank_32x32|   0|     40000|     100|0.31644099|  PASSED
# The file file_input_raw was rewound 5 times
    diehard_rank_6x8|   0|    100000|     100|0.99999847|   WEAK
# The file file_input_raw was rewound 5 times
   diehard_bitstream|   0|   2097152|     100|0.83665418|  PASSED
# The file file_input_raw was rewound 8 times
        diehard_opso|   0|   2097152|     100|0.84988940|  PASSED
# The file file_input_raw was rewound 11 times
        diehard_oqso|   0|   2097152|     100|0.29356428|  PASSED
# The file file_input_raw was rewound 12 times
         diehard_dna|   0|   2097152|     100|0.26602694|  PASSED
# The file file_input_raw was rewound 12 times
diehard_count_1s_str|   0|    256000|     100|0.73935087|  PASSED
# The file file_input_raw was rewound 14 times
diehard_count_1s_byt|   0|    256000|     100|0.64773200|  PASSED
# The file file_input_raw was rewound 14 times
 diehard_parking_lot|   0|     12000|     100|0.26192817|  PASSED
# The file file_input_raw was rewound 14 times
    diehard_2dsphere|   2|      8000|     100|0.86039487|  PASSED
# The file file_input_raw was rewound 14 times
    diehard_3dsphere|   3|      4000|     100|0.84200798|  PASSED
# The file file_input_raw was rewound 18 times
     diehard_squeeze|   0|    100000|     100|0.73474226|  PASSED
# The file file_input_raw was rewound 18 times
        diehard_sums|   0|       100|     100|0.00039602|   WEAK
# The file file_input_raw was rewound 18 times
        diehard_runs|   0|    100000|     100|0.64154121|  PASSED
        diehard_runs|   0|    100000|     100|0.74432339|  PASSED
# The file file_input_raw was rewound 20 times
       diehard_craps|   0|    200000|     100|0.48674360|  PASSED
       diehard_craps|   0|    200000|     100|0.65536404|  PASSED
```

Figure C.4: Diehard results on PBF-treated RNG output

The Diehard test suite, which consists of 19 statistical tests for randomness, showed a significant improvement compared to previous results without post-processing. Out of the 19 tests, 17 tests passed successfully with no issues. The remaining 2 tests were marked as WEAK, but this result was attributed to the relatively small sample size used in these specific tests, rather than any significant deviation from uniformity. This outcome demonstrates the robustness of the PBF solution in enhancing the uniformity output. With larger sample sizes, these weak results can be mitigated, leading to a fully compliant output for all tests in the Dieharder suite.

While more work would be required to ensure acceptable commercial-grade or military-grade performance, the preceding experiments and results show that it is indeed possible to improve randomness of numbers generated by a QRNG by implementation of Bloom Filters.

Bloom filters and related probabilistic logic, when implemented at the hardware or FPGA level can thus improve the uniformity of QESs' outputs by real-time filtering of repeat patterns or buffer echo effects. This logic should be made part of FIPS boundary where applicable, and its state should be cryptographically signed on boot or update.

Annex D: NIST SP800-90B: Unconditional/Statistical entropy estimation

NIST SP 800-90B [i.23] provides a comprehensive methodology for estimating and evaluating the amount of **unconditional** entropy in a purported entropy source and how to submit it for evaluation. Given it focus on the observed entropy $H_{\min}(A)$, it does not quantify unpredictability. As shown in clause 4, one can design devices whose outcomes are uniformly distributed but are, yet, completely predictable by an adversary with the right side-information.

Following is a summary of the NIST SP 800-90B [i.23] standard.

1) Understand the noise source:

   Detailed documentation is required. The submitter of the entropy source should provide thorough documentation explaining the physical and/or computational processes that generate the noise. This includes the theory of operation, the underlying principles, and why it is expected to produce entropy.

   The submitter should also make a claim and justification concerning the amount of min-entropy, which is a conservative measure of randomness, per output sample of the noise source. This claim should ideally be supported by a technical argument based on the design and operation of the source.

   The submitter should also state whether they believe the noise source output to be Independent and Identically Distributed (IID). This assumption would affect the types of statistical tests used for evaluation.

2) Data Collection Requirements:

   NIST SP 800-90B [i.23] specifies the collection of two primary datasets from the raw noise source output, before any conditioning, as per the following.

   Sequential Dataset: At least 1 000 000 consecutive samples from the noise source are collected during normal operation.

   Restart Dataset: The entropy source is restarted 1 000 times, and after each restart, 1 000 consecutive samples are collected. This helps detect issues that might only appear across multiple power cycles or resets.

   Optionally, if a non-vetted conditioning component is used, a separate dataset of 1 000 000 samples from the conditioner output might be required for evaluation.

3) Entropy Estimation Techniques:

   NIST SP 800-90B [i.23] outlines a suite of statistical tests to estimate the unconditional min-entropy of the collected data. These tests are applied to the raw noise source data. The tests are categorized for both IID and Non-IID assumptions.

   Assessment of whether the source is IID: If the submitter claims the source is IID, and this is not statistically refuted, a set of tests is applied to estimate the min-entropy based on the distribution of symbols or patterns within the sequential data. These tests look at the frequency of the most common outcomes and include the following:

   a) Frequency estimate based on the most frequently occurring symbol.

   b) Collision estimate based on the probability of repeated symbols.

   c) Longest repeated substring test that identifies unusually long repeating sequences.

   d) Most common value estimate, that considers the frequencies of the most frequent values.

   e) Lempel-Ziv78Y Estimate evaluates the entropy based on the compressibility of the data.

In the case that the source is determined to be Non-IID, a different set of tests is applied that are designed to detect dependencies and predictability in the sequence. These tests often look at patterns across the data as follows:

i) Markov Test: Checks for dependencies between consecutive symbols.

ii) Prediction by Partial Matching (PPM) Test: Estimates predictability based on prior symbols.

iii) Multi Permutation Test: Looks for biases in the ordering of symbols.

iv) Lag Prediction Test: Checks for predictability based on values at specific lags.

v) Tuple Prediction Test: Examines predictability based on short sequences of symbols.

vi) Restart Tests: These tests are applied to the restart dataset to assess the consistency of the source across restarts. They look for biases or predictable behaviour that might emerge after a reset.

The final min-entropy estimate for the noise source is typically the minimum of the estimates obtained from all applicable statistical tests and the submitter's claim, if the claim is statistically supported.

4) Health Testing:

Beyond entropy estimation, NIST SP 800-90B [i.23] mandates health tests to be performed on the raw noise source output during operation. These tests are designed to detect catastrophic failures or significant reductions in entropy. The specification defines two core health tests:

- Repetition count tests detect if the noise source "freezes" and produces the same output value repeatedly for an extended period.

- Adaptive proportion tests monitor the proportion of zeros and ones of the underlying binary source and flags significant deviations from the expected proportions, which could indicate a loss of entropy.

5) Conditioning (Optional):

NIST SP 800-90B [i.23] allows for optional conditioning of the raw noise source output to improve its statistical properties or increase the entropy rate. It specifies a set of "vetted" conditioning functions based on well-understood cryptographic primitives. If a vetted conditioning function is used with a noise source of sufficient min-entropy, the output of the conditioner can, under certain circumstances, be claimed to have full entropy. Non-vetted conditioning functions require careful analysis, and the entropy of their output needs to be estimated.

6) Validation Process:

The overall validation process involves testing by an accredited laboratory against the requirements of NIST SP 800-90B [i.23]. This includes performing the statistical tests on the collected data and verifying the implementation of the health tests. The results are then reviewed, and if the entropy source meets the requirements, it can be listed as a validated entropy source.

To conclude, estimating the entropy of a quantum entropy source is not an exact science, and requires a multi-disciplinary approach that combines a deep understanding of the underlying quantum physics, careful data acquisition, the use of appropriate entropy estimation techniques, statistical analysis and rigorous validation. NIST SP 800-90B [i.23] is a valuable resource, especially on the statistical aspects in this domain.

Annex E:
Standardized Tests for Deviations from Uniformity

There are relatively few standards for testing whether the output of a random number generator is truly random, as opposed to displaying something other than a uniform distribution of data over a large number of samples. A most relevant such test at the present time is the NIST SP 800-22 specification [i.20], which applies a suite of statistical tests to a sequence of generated numbers. The assumption is that a truly random sequence should exhibit certain statistical properties, and that deviations from these properties suggest a lack of randomness or the presence of possibly repeating patterns or predictable distribution.

Each test in the NIST SP 800-22 [i.20] suite uses the principle of hypothesis testing. The null hypothesis is that the sequence of numbers being tested is random. The statistical test then calculates a p-value, which is the probability of that the null hypothesis was true. A high p-value, typically above a chosen significance level such as 0,01, suggests that the observed result is likely to occur even with a truly random sequence. In such a case, the null hypothesis, i.e. randomness, is not rejected. This means that there would be no strong statistical evidence to suggest that the generator is non-random based on a specific test. On the other hand, a low p-value would suggest that the observed result is unlikely to occur by chance. In this case, the null hypothesis of randomness is rejected. This provides statistical evidence that the generator might not be producing truly random numbers according to that specific test.

There are a variety of parameters that are tested by the NIST SP 800-22 [i.20] suite, each designed to detect different types of non-randomness or patterns:

a) **Frequency tests** check that the number of occurrences of zeros and ones in the sequence are approximately equal, as would be expected in a truly random sequence.

b) **Run tests** examine the number and length of consecutive sequences of identical bits, "runs" of zeros or ones. A truly random sequence should have a predictable distribution of run lengths.

c) **Longest Run of Ones in a Block tests** evaluate the longest consecutive sequence of ones within fixed-size blocks of the overall sequence, like taking snapshots of information of the output data. Deviations from expected lengths may indicate non-randomness.

d) **Binary Matrix Rank Tests** evaluate the linear dependence among fixed-length substrings of an output sequence. Random sequences would exhibit a certain rank in these matrices.

e) **Discrete Fourier Transform Tests** analyse the spectral frequency components of the sequence, where a truly random sequence would have a flat spectrum with no dominant repeating patterns.

f) **Non-overlapping Template Matching Tests** search for specific pre-defined patterns within an output sequence. So rather than testing for ones and zeros repetition, these tests check for the recurrence of specific sequences, or "templates". The number of times these templates appear should be consistent with randomness.

g) **Overlapping Template Matching Tests** are similar to the non-overlapping template test, but allows templates to overlap in certain cases, providing additional sensitivity to different types of patterns.

h) **Maurer's "Universal Statistical" Tests** detect whether the sequence can be significantly compressed. Highly compressible sequences are not likely to be random.

i) **Linear Complexity Tests** measure the length of the shortest Linear Feedback Shift Register (LFSR) that is able to generate a specific sequence, where truly random sequences would have a high linear complexity.

j) **Serial Tests** analyse the frequency of all possible overlapping patterns of a certain length.

k) **Approximate Entropy Tests** quantify the irregularity and unpredictability of a number sequence.

l) **Cumulative Sums (Cusum) Tests** analyses the cumulative sum of the deviations of the number of ones from their expected proportion in a large population of data. Large deviations from expected values may indicate non-randomness.

m) **Random Excursions Tests** analyse the number of times a random walk, derived from the sequence, visits or crosses a specific state. A large number of crossings may indicate that a pattern exists in the data.

n) **Random Excursions Variant Tests** are similar to random excursion tests, but focuses on the number of cycles having a specific total number of visits to a state, where a high number of visits to a specific state may indicate non-randomness.

A random number generator is considered to have passed the NIST SP 800-22 [i.20] tests if the p-values for all the individual tests exceed the chosen significance level. This indicates that there is no strong statistical evidence to reject the null hypothesis that the generated sequence is believed to be random, based on the various aspects of randomness that each test evaluates.

While the NIST SP 800-22 specification [i.20] does not prove unequivocally that a random number generator is perfectly random, it provides a rigorous statistical framework to assess whether a generator produces sequences that are statistically indistinguishable from a truly random sequence according to a comprehensive set of criteria. Failing even one of these tests raises concerns about the generator's suitability for applications requiring high-quality randomness, such as the generation of cryptographic passwords or certificates.

Other NIST specifications on this topic include the NIST SP 800-90 Series, which aims to improve the overall quality of the generators by specifying design principles, but do not provide full solutions:

i) NIST SP 800-90A [i.22]. At present NIST is working on a revision of SP 800-90A.

ii) NIST SP 800-90B [i.23].

iii) NIST SP 800-90C [i.24].

There also exist other statistical test suites which are not formal standards in the same way as NIST, but are widely used in industry in the valuation of random number generators. These include the following:

a) Diehard Battery of Tests: A collection of statistical tests developed by George Marsaglia, known for its high degree of rigor.

b) TestU01: A library of statistical tests for RNGs developed by Pierre L'Ecuyer and Richard Simard, offering various test batteries with different degrees of computational intensity.

c) PractRand: Another widely used test suite by David Blackman, known for its sensitivity in detecting certain types of non-randomness.

d) ENT: A simple utility by John Walker that performs several basic statistical tests on a stream of data.

e) Note that at this time, NIST SP 800-90C [i.24] has not yet published any certification path, and there are no certifications available, only self-claimed compliance.

Annex F: Bibliography

- S. Majidy, C. Wilson, R. Laflamme: "Building Quantum Computers: A Practical Introduction", Cambridge University Press, 2024.
- James R. Carr: "Simple random number generation", Department of Geological Sciences, Mackay School of Mines, University of Nevada, Mail Stop 172, Reno, NV 89557-0138, USA, 7 November 2002.
- Herrero-Collantes, M., & Garcia-Escartin, J. C. (2017): "Quantum random number generators", Reviews of Modern Physics, 89(1), 015004.
- Bera, M. N., Acín, A., Kuś, M., Mitchell, M. W., & Lewenstein, M. (2017): "Randomness in quantum mechanics: philosophy, physics and technology", Reports on progress in physics, 80(12), 124001.
- Mannalatha, V., Mishra, S., & Pathak, A. (2023): "A comprehensive review of quantum random number generators: concepts, classification and the origin of randomness", In Quant. Inf. Proc. (Vol. 22, No. arXiv: 2203.00261, p. 439).
- Gabriel, C., Wittmann, C., Sych, D., Dong, R., Mauerer, W., Andersen, U. L., & Leuchs, G. (2010): "A generator for unique quantum random numbers based on vacuum states", Nature Photonics, 4(10), 711-715.
- Uchida, A., Amano, K., Inoue, M., Hirano, K., Naito, S., Someya, H., Davis, P., et al. (2008): "Fast physical random bit generation with chaotic semiconductor lasers", Nature Photonics, 2(12), 728-732.
- T. Ferreira da Silva, G. B. Xavier, G. C. Amaral, G. P. Temporão, J. P. von der Weid (2016), "Quantum random number generation enhanced by weak-coherent states interference", arXiv:1608.05155.
- Qi, B., Chi, Y. M., Lo, H. K., & Qian, L. (2010): "High-speed quantum random number generation by measuring phase noise of a single-mode laser", Optics Letters, 35(3), 312-314.

Annex G:
Change history

Date	Version	Information about changes
April 2025	Draft 01	Started WI, Scoping statements, basic outline clauses and inserted certain content
August 2025	Draft 02	Added some content - awaiting other contributions
November 2025	Draft 03	Added content as per feedback from ETSI members, added examples to each their own Annex, updated references
January 2026	Draft 04	Cleaned up editorial changes, small corrections to terms
January 2026	Draft 05	Fixed hyper-links, small corrections
February 2026	Draft 06	Additional small corrections

History

Version	Date	Status
V1.1.1	March 2026	Publication