

Guía de Seguridad de las TIC CCN-STIC 2110

Random Generators Evaluation Methodology Metodología de Evaluación de Generadores Aleatorios (MEGA)



Noviembre 2025





Catálogo de Publicaciones de la Administración General del Estado
cpage.mpr.gob.es

Edita:



Pº de la Castellana 109, 28046 Madrid
© Centro Criptológico Nacional, 2025
NIPO pendiente de asignación.



CENTRO CRIPTOLOGICO
NACIONAL
cn=CENTRO CRIPTOLOGICO
NACIONAL, 2.5.4.97=VATES-
S2800155J, ou=CENTRO
CRYPTOLOGICO NACIONAL,
o=CENTRO CRIPTOLOGICO
NACIONAL, c=ES
2025.11.21 12:59:59 +01'00'

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

TABLE OF CONTENTS

1 INTRODUCTION.....	8
1.1 OBJECTIVE	8
1.2 STRUCTURE OF THE DOCUMENT	8
1.3 EVALUATION PROCESS	10
1.4 GLOSSARY	11
2 PHYSICAL TRUE RANDOM NUMBER GENERATORS.....	26
2.1 OBJECTIVE	26
2.2 CCN-PTRNG DEFINITION	26
2.3 INPUTS.....	28
2.4 CCN-PTRNG EVALUATION TASKS	32
2.4.1 EVALUATION TESTS DEFINITION	35
3 HYBRID PHYSICAL TRUE RANDOM NUMBER GENERATORS	66
3.1 OBJECTIVE	66
3.2 CCN-HPTRNG DEFINITION	66
3.3 INPUTS.....	67
3.4 CCN-HPTRNG EVALUATION TASKS.....	68
3.4.1 EVALUATION TESTS DEFINITION	70
4 NON-PHYSICAL TRUE RANDOM NUMBER GENERATORS	76
4.1 OBJECTIVE	76
4.2 CCN-NPTRNG DEFINITION	76
4.3 INPUTS.....	78
4.4 CCN-NPTRNG EVALUATION TASKS.....	80
4.4.1 EVALUATION TESTS DEFINITION	83
5 DETERMINISTIC RANDOM NUMBER GENERATORS	97
5.1 OBJECTIVE	97
5.2 CCN-DRNG DEFINITION	97
5.3 INPUTS.....	98
5.4 CCN-DRNG EVALUATION TASKS.....	99
5.4.1 EVALUATION TESTS DEFINITION	102
6 ENHANCED DETERMINISTIC RANDOM NUMBER GENERATORS	116
6.1 OBJECTIVE	116
6.2 CCN-EDRNG DEFINITION	116
6.3 INPUTS.....	117
6.4 CCN-EDRNG EVALUATION TASKS.....	117
6.4.1 EVALUATION TESTS DEFINITION	120
7 ANNEX A: TRACEABILITY OF EVALUATION TASKS.....	131
7.1 CCN-PTRNG EVALUATION TASKS	131
7.2 CCN-HPTRNG EVALUATION TASKS.....	134
7.3 CCN-NPTRNG EVALUATION TASKS.....	135
7.4 CCN-DRNG EVALUATION TASKS.....	137
7.5 CCN-EDRNG EVALUATION TASKS.....	139

8	REFERENCES.....	141
9	ABBREVIATIONS.....	142

INDEX OF TABLES

Table 1. Evaluation Tests defined for CCN-PTRNG Requirements	34
Table 2. Functional Design Review – Notes.....	35
Table 3. Analog Stochastic Model – Notes	38
Table 4. Digital Stochastic Model – Notes	41
Table 5. Raw Entropy Estimation Method – Notes	45
Table 6. Parameter Sets Definition – Notes	46
Table 7. Validation of Assumptions – Notes.....	48
Table 8. Parameter Estimation – Notes.....	50
Table 9. Modeled Entropy Claim – Notes.....	52
Table 10. Internal Random Numbers Entropy – Notes	53
Table 11. Online Test – Notes.....	56
Table 12. Total Failure Test – Notes	58
Table 13. Start-Up Test – Notes.....	60
Table 14. Health Testing Alarms – Notes	61
Table 15. AIS-31 Testing – Notes	62
Table 16. SP 800-90B Test Suite– Notes.....	63
Table 17. Evaluation Tests defined for CCN-HPTRNG Requirements.....	69
Table 18. Composite Design – Notes.....	70
Table 19. PTRNG Evaluation – Notes.....	71
Table 20. Entropy Injection – Notes	73
Table 21. Evaluation Tests defined for CCN-NPTRNG Requirements	82
Table 22. Noise Sources – Notes	83
Table 23. Entropy Pool – Notes	84
Table 24. DRNG Component – Notes	86
Table 25. Heuristic Analysis – Notes.....	87
Table 26. Entropy Estimation – Notes	89
Table 27. Continuous Test – Notes.....	90
Table 28. Start-Up Test – Notes.....	92
Table 29. Health Testing Alarms – Notes	93
Table 30. SP 800-90B Test Suite– Notes.....	95
Table 31. Evaluation Tests defined for CCN-DRNG Requirements.....	102
Table 32. Cryptographic Evaluation – Notes	102
Table 33. Seeding Procedure – Notes	104
Table 34. Reseeding Procedure – Notes.....	105
Table 35. DRNG Formal Model – Notes.....	107
Table 36. Request Length – Notes.....	109
Table 37. Effective Internal State – Notes	109
Table 38. Forward Secrecy – Notes	110
Table 39. Backward Secrecy – Notes.....	111
Table 40. Enhanced Backward Secrecy – Notes	112
Table 41. Additional Input – Notes.....	114
Table 42. Statistical Inconspicuousness – Notes	114
Table 43. Evaluation Tests defined for CCN-EDRNG Requirements.....	119
Table 44. Cryptographic Evaluation – Notes	120

Table 45. Seeding Procedure – Notes	121
Table 46. Reseeding Procedure – Notes.....	122
Table 47. DRNG Formal Model – Notes.....	124
Table 48. Request Length – Notes.....	125
Table 49. Effective Internal State – Notes	125
Table 50. Forward Secrecy – Notes	126
Table 51. Backward Secrecy – Notes	126
Table 52. Enhanced Backward Secrecy – Notes	127
Table 53. Additional Input – Notes.....	128
Table 54. Enhanced Forward Secrecy – Notes	129
Table 55. Statistical Inconspicuousness – Notes	129
Table 56. Summary of CCN-PTRNG Evaluation Tasks.....	133
Table 57. Summary of CCN-HPTRNG Evaluation Tasks	134
Table 58. Summary of CCN-NPTRNG Evaluation Tasks	136
Table 59. Summary of CCN-DRNG Evaluation Tasks	138
Table 60. Summary of CCN-EDRNG Evaluation Tasks	140

INDEX OF FIGURES

Figure 1. CCN-PTRNG Design Scheme	26
Figure 2. CCN-HPTRNG Design Scheme	66
Figure 3. CCN-NPTRNG Design Scheme	77
Figure 4. CCN-DRNG Design Scheme	97
Figure 5. CCN-EDRNG Design Scheme	116

1 INTRODUCTION

1. The generation of cryptographically secure random numbers is a foundational pillar of modern cryptography. Without reliable randomness, the security guarantees offered by cryptographic mechanisms—such as key generation, encryption or digital signatures—collapse. Many of these mechanisms are formally proven secure only under the assumption that the random inputs they rely on are truly unpredictable and random.
2. Therefore, the evaluation of random number generators (RNGs) is a critical step in certifying the overall security of technological products. Weaknesses in entropy generation can lead to catastrophic cryptographic failures, which are undetectable through standard functional testing.

1.1 OBJECTIVE

3. The objective of this document is to define the CCN-MEGA methodology, a common methodology for the evaluation of Random Number Generators (RNGs) integrated in technological products undergoing evaluation within the Spanish certification scheme.
4. In particular, this methodology supports the Cryptographic Mechanisms Evaluation Methodology [CCN-MEMeC], which, when random number generators are used, mandates that they belong to the functionality classes recommended by the CCN and have been successfully evaluated in accordance with the present methodology.
5. The present methodology establishes requirements and evaluation criteria aligned with both the German AIS 20/31 standards—developed by the Bundesamt für Sicherheit in der Informationstechnik (BSI)—and the random bit generation standards of the U.S. National Institute of Standards and Technology (NIST), the Special Publications [SP 800-90A], [SP 800-90B], and [SP 800-90C]. As such, the methodology adds value to ongoing efforts aimed at harmonizing global frameworks for the evaluation of random number generators.

1.2 STRUCTURE OF THE DOCUMENT

6. This methodology applies to different CCN-approved functionality classes for random number generators. A generator belongs to a given functionality class if it is successfully evaluated according to the specific set of tasks defined in this methodology for that class.
7. The supported functionality classes are:
 - **CCN-PTRNG:** Physical True Random Number Generator. RNGs that extract entropy from dedicated hardware components exploiting physical noise phenomena.

- **CCN-NPTRNG**: Non-Physical True Random Number Generator. RNGs based on non-dedicated or system-level sources.
 - **CCN-HPTRNG**: Hybrid True Random Number Generator. RNGs that combine a CCN-PTRNG and a CCN-DRNG as the post-processing component.
 - **CCN-DRNG**: Deterministic Random Number Generator. RNGs that use deterministic cryptographic constructions seeded with sufficient entropy.
 - **CCN-EDRNG**: Deterministic RNG with enhanced assurance. Deterministic RNGs that comply with additional requirements.
8. The **vendor shall** declare the functionality class for the TOE. This declaration determines the applicable set of evaluation tasks within this methodology.
9. The definition of the requirements to be met by the TOE and the evaluation tasks to be performed by the tester have been made following this structure:
- The requirements start with the words “The **TOE shall** [XRNG.YY]”, (where “XRNG” corresponds to the declared functionality class, and “YY” is a numerical tag) followed by the description of the requirement to be met by the TOE.
 - The evaluation tasks are defined in a table which specifies an evaluation task ID and a brief description of the tasks to be performed by the tester, which begins with the words “The **tester shall**” followed by a brief description and the required inputs by the tester to carry out the evaluation. The task ID is denoted using bold typography with the following format **CCN-XRNG-NAME**, where “NAME” is the name assigned to the task. The format of an evaluation task is as detailed below:

CCN-XRNG EVALUATION TASK	CCN-XRNG-NAME
The tester shall ...	
[Evaluation Task Definition]	

10. The tests associated with each evaluation task are defined in a table divided into two columns. The first column contains the identifier (ID) of the test and a brief description of the activities to be performed by the tester. The second column specifies the inputs required to perform the test. The ID is written in bold typography following the format **CCN-XRNG-NAME/TestName**, where “CCN-XRNG-NAME” is the ID of the evaluation task associated with the test, and “TestName” is the specific name of the cryptographic test to be performed:

CCN-XRNG-NAME/TestName	Inputs
[Evaluation Test Definition]	

1.3 EVALUATION PROCESS

11. This section defines the procedure to be followed by the tester once the required evaluation inputs described have been received from the vendor.
12. In addition to the inputs required for each functionality class, the **vendor shall** provide the tester with the “*CCN-MEGA Evidence Matrix*” document ([CCN-MEGA-EV]), a document containing traceability tables for all the requirements the TOE shall meet in order to comply with each functionality class defined in this methodology. This document is not considered an input of the methodology itself. In it, the **vendor shall** declare the claimed functionality class for the TOE and, for each applicable requirement, provide an explicit reference to the relevant section of the evaluation input containing the supporting evidence.
13. The **tester shall** use this document as a reference index to navigate the evidence submitted by the vendor across the various inputs required by the methodology.
14. The sections of this methodology have been organized to group, for each functionality class, the evaluation tasks that apply to the TOE, along with the corresponding definitions and required evaluation inputs. This structure allows the tester to locate and execute the applicable tasks efficiently, according to the declared functionality class.
15. The **tester shall** perform the evaluation tasks defined in the section corresponding to the functionality class to which the TOE is claimed to belong. Each evaluation task is further divided into one or more specific evaluation tests.
16. Considering that each evaluation task is composed of one or several tests, the **tester shall** execute those categorized as mandatory (see the note below).and those categorized as Implementation-Dependent (Impl-Dep), only in case the TOE supports the associated functionality.
17. At the beginning of each section, a summary table is provided listing all the evaluation tests associated with each task, following the format detailed below. The first column indicates the evaluation task and the second column the test name, and the last column the test categorization.

Note: In the absence of test categorization, all tests are to be considered as mandatory.
18. For improved traceability throughout the evaluation process, the tester shall make use of the summary tables included in "Annex A: Traceability of Evaluation Tasks".

CCN-XRNG Evaluation Task	Evaluation Test	Test Categorization
CCN-XRNG-NAME	CCN-XRNG-NAME/TestName	Mandatory or Impl-Dep

1.4 GLOSSARY

Additional Input

19. Any data that is input into a DRNG by the state transition function (i.e., not by the seeding procedure or reseeding procedures). This data can be provided by an internal or external entropy source (PTRNG, HPTRNG or NPTRNG); they can or may not contain entropy; they may be provided by a reliable source or be under the control of an attacker.

Analog Noise Source

20. In the context of the CCN-PTRNG functionality class, the component that harnesses the physical phenomenon which generates its inherent unpredictability.
21. This component shall not perform any digitization; its output is the raw random signal.

Backward Secrecy

22. In the context of the DRNG functionality classes, backward secrecy ensures that knowledge about previous output values cannot be derived with practical computational effort from the knowledge of current or subsequent output values. Together with the forward secrecy property, this ensures that a DRNG remains unpredictable.
23. "Deriving knowledge" means gaining significant advantage over blind guessing.

Compression Rate

24. Ratio between the average input bit length of the cryptographic post-processing algorithm and the bit length of the resulting internal random numbers per (short) time interval; ideally holds for each internal random number. This term is specific to the CCN-HPTRNG functionality class.
25. To increase the entropy per bit, the compression rate must be > 1 .

Computational Effort

26. Computational effort refers to the practical workload or cost (in terms of time and resources) needed to carry out an attack or derive information, often measured relative to what is feasible with current or foreseeable technology.

Computational Power

27. Computational power refers to the practical ability of an entity (e.g., an attacker or a system) to perform a large number of operations or calculations within a feasible amount of time, given real-world resource constraints.

Computational Security

28. Since the generation mechanism of DRNGs is fully deterministic, they are ineffective against an attacker with unlimited computational resources if additional input or a reseeding procedure are not supported. An attacker with

unlimited computational power would be able to break any DRNG. Consequently, DRNGs generate pseudo-random numbers rather than truly random numbers.

29. The computational security refers to the security against an attacker with bounded computing power. Quantified by the security strength (of cryptographic mechanisms). Therefore, the security of the DRNG relies on the computational security, quantified by its security strength.

Conceptual Atomicity

30. The requests of a DRNG satisfy the condition of conceptual atomicity if the DRNG finishes every request by the application of the state transition function before any of the requested bits are used (are output or, at least, can be used by the application).
31. In the context of CPU instructions, “atomic” refers to operations that are not interruptible by other operations.

Continuous Test

32. In the context of the CCN-NPTRNG functionality class, these health tests shall be executed continuously on all digitized samples obtained from the noise source (i.e., raw random numbers) while the noise source is operating.
33. Continuous tests focus on the noise source behavior and aim to detect failures as the noise source produces output.

Digitizer

34. The component of the noise source responsible for transforming the raw random signal into raw random numbers. In other words, it is the part of the noise source that performs the digitization process on the raw random signal to produce the raw random numbers.
35. Note 11 [Digitization Function] provides a formal mathematical description of the digitalization process through a digitalization function.

DRNG Formal Model

36. The DRNG Formal Model mathematically describes the CCN-DRNG algorithmic structure by a 9-tuple. The 9-tuple is composed of the following parameters, $(S, S_{req}, R, A, I, \phi, \phi_{req}, \phi_0, \psi)$. The meaning of these parameters is described in paragraph 129 of [FCRNG].
37. The DRNG Formal Model also provides a mathematical description of the seeding procedure, which is defined as a 4-tuple, (SM, PS, S, ϕ_{seed}) and follows the structure and interpretation presented in paragraph 143 and the subsequent paragraphs of [FCRNG].
38. When applicable, the DRNG Formal Model likewise provides a mathematical description of the reseeding procedure, defined as a 4-tuple $(SM, PS, S, \phi_{reseed})$, which follows the structure and interpretation outlined in paragraph 147 and the subsequent paragraphs of [FCRNG].

39. The parameters S, S_{req}, R, A, I are sets of possible values for different variables or states and the parameters $\phi, \phi_{req}, \phi_0, \psi$ are functions or mappings defining the DRNG functioning.

Effective Internal State

40. The security-critical part of the internal state of a DRNG that an attacker does not know and that he cannot determine or guess (with probability that is significantly greater than indicated by its size, assuming optimal encoding) from the knowledge of subsequences of internal random numbers.
41. This definition does not take an attacker with unlimited computational power into account, because such an attacker could determine the entire internal state by just knowing a few output numbers. Instead, this definition aims at computational security.

Enhanced Backward Secrecy

42. In the context of the DRNG functionality classes, enhanced backward secrecy ensures that knowledge about previous output values cannot be derived with practical computational effort from the knowledge of the current internal state of the DRNG. A DRNG that offers enhanced backward secrecy is more secure than one that provides only backward secrecy and forward secrecy. These properties aim to lessen the impact of a compromise of the internal state.
43. “Deriving knowledge” means gaining significant advantage over blind guessing.

Enhanced Forward Secrecy

44. In the context of the CCN-EDRNG functionality class, enhanced forward secrecy ensures that knowledge about subsequent output values cannot be derived with practical computational effort from the knowledge of the current internal state of the EDRNG.
45. The enhanced forward secrecy requirement can only be achieved using fresh entropy. Unlike forward secrecy and backward secrecy as well as enhanced backward secrecy, enhanced forward secrecy rests entirely on the capability of inserting as much entropy as is required to make the prediction of future outputs infeasible.
46. This concept is conceptually similar to the *prediction resistance* concept from employed by NIST.

Entropy Counter

47. In the context of the CCN-NPTRNG functionality class, the entropy counter is an entropy pool component responsible for tracking the amount of entropy that has been introduced into the entropy pool, as well as the amount that has been extracted. The extracted amount corresponds to the number of internal random number bits that have been output.

48. The entropy counter is capped by both the maximum amount of entropy that the mixing function can insert into the entropy pool and the maximum amount of entropy that the entropy pool can store.
49. The NPTRNG shall never generate more internal random number bits than indicated by the entropy counter.

Entropy Pool

50. The entropy pool is an essential component of the CCN-NPTRNG architecture that accumulates and mixes the entropy extracted from raw random numbers produced by non-physical noise sources. It serves as a reservoir of entropy, ensuring that sufficient randomness is available before generating internal random numbers.
51. The entropy pool can either coincide with the internal state of the DRNG component or operate as a distinct component that periodically reseeds a separate DRNG.

Environmental Conditions

52. Environmental conditions represent the subset of external and non-controllable parameters within the overall operational conditions. (e.g., temperature, power supply variations, and related factors) under which the RNG operates as intended and provides the claimed entropy.
53. Representative environmental conditions are used for testing and entropy estimation to verify that the behavior of the raw random numbers remains valid across the admissible operating range of these parameters.
54. Environmental conditions are relevant in the context of PTRNGs, as entropy depends on the physical behavior of the hardware, which can be affected by temperature, voltage, and other environmental factors. In contrast, NPTRNGs do not depend directly on physical parameters to the same extent, but rather on software factors and the operating system.

Forward Secrecy

55. In the context of the DRNG functionality classes, forward secrecy ensures that knowledge about subsequent output values cannot be derived with practical computational effort from the knowledge of current or previous output values. Together with the backward secrecy property, this ensures that a DRNG remains unpredictable.
56. "Deriving knowledge" means gaining significant advantage over blind guessing.

Fresh Entropy

57. In the context of the CCN-DRNG functionality class, fresh entropy refers to a random bit string recently generated by a noise source. In particular, "fresh" means that the bit string has not been previously used to generate output or has otherwise been made externally available.

58. Fresh entropy is introduced to the DRNG as additional input or by reseeding procedure.
59. The noise source should be compliant with CCN-PTRNG, CCN-HPTRNG, or CCN-NPTRNG.

Global Deterministic Noise Component

60. This term refers to components of the noise source that produce predictable or highly correlated behavior and do not contribute to true randomness. These include deterministic or quasi-deterministic phenomena—such as flicker noise, crosstalk, or power supply fluctuations—that may superficially appear as noise but can be anticipated or controlled. In entropy evaluations, these components must be excluded from the stochastic model to ensure that only genuinely unpredictable effects are considered.
61. They are contrasted with truly random noise components, which stem from fundamental physical unpredictability (e.g., thermal noise) and are the only valid basis for entropy claims.

Granularity Level

62. An auxiliary term to express for which segments of the output of a DRNG security properties such as forward secrecy, backward secrecy, and enhanced backward secrecy hold.
63. The DRNG provides an algorithmic property (forward secrecy, backward secrecy or enhanced backward secrecy) on the:
 - Granularity level of internal random numbers (or within requests): if knowledge about subsequent internal random numbers cannot be derived with practical computational effort from the knowledge of current or previous internal random numbers.
 - Granularity level of requests: if knowledge about previous requests cannot be derived with a practical computational effort from the knowledge of the current internal state of a DRNG.

Heuristic Analysis

64. A heuristic analysis for entropy estimation of a non-physical noise source involves the use of empirical methods, plausible assumptions, and simplified models to approximate the amount of entropy present in the output data. This analysis provides a structured justification—based on architectural or system-level reasoning and supported by statistical evaluation of empirical data—those accounts for the unpredictability of the noise source's output in the absence of a complete mathematical model.
65. Since the underlying processes of non-physical noise sources often lack a well-defined physical basis and are not easily modeled stochastically, precise analytical entropy estimation is typically infeasible. Consequently, heuristic analysis relies on observed behavior, plausible assumptions about system

operation, and hypothetical adversarial capabilities to conservatively estimate entropy for the collected raw random numbers.

66. While these models provide practical guidance, their accuracy depends heavily on the validity of the assumptions made and the specific operational environment.

High-Entropy Additional Input

67. In the context of the CCN-EDRNG functionality class, the wording “high-entropy additional input” is used if the additional input ensures enhanced forward secrecy. This requires that the additional input has at least 240 bits of min-entropy.
68. One-time high-entropy input cannot be compensated by many low-entropy additional inputs within several requests.

Ideal RNG

69. A mathematical construct that generates independent and uniformly distributed random numbers.

Initial Internal State

70. The first internal state computed from a specific seed material. It corresponds to the starting state for each instantiation of the DRNG.

Instantiate a DRNG

71. Instantiating a DRNG refers to initializing the DRNG. The instantiation process begins by acquiring seed material which may be combined with a personalization string to generate a seed. This seed is then used to create the initial internal state of the DRNG.
72. Each DRNG instance finalizes with an un instantiation process.

Intermediate Random Number

73. In the context of the CCN-HPTRNG functionality class, intermediate random numbers are the random numbers input to the DRNG post-processing component. They coincide with the internal random numbers of the CCN-PTRNG inner PTRNG component.
74. In the context of the CCN-NPTRNG functionality class, the intermediate random numbers are the output data of the entropy pool and the input data for cryptographic post-processing.

Internal Random Number

75. Internal random numbers are the final stage of the random numbers of an RNG that are ready to be output.
76. In the context of the CCN-PTRNG functionality class, internal random numbers correspond to the output of the post-processing component, if such a

component is employed; otherwise, they correspond to the raw random numbers.

77. In the context of the CCN-HPTRNG and CCN-NPTRNG functionality classes, internal random numbers coincide with the output of the CCN-DRNG post-processing component.
78. In the context of the CCN-DRNG and CCN-EDRNG functionality classes, internal random numbers refer to the numbers generated when the DRNG produces the requested number of bits.

Internal State

79. In the context of the CCN-DRNG functionality class, internal state refers to the collection of all secret and non-secret digitized information of a DRNG as stored in memory at a given point in time.

Life Cycle of a DRNG Instance

80. The life cycle of a DRNG instance begins with the seeding procedure or the reseeding procedure. It ends with the following reseeding procedure or when the DRNG is uninstantiated. An externally triggered seeding procedure or reseeding procedure starts a new life cycle.

Min-Entropy per Internal Random Number Bit

81. Quantifies the min-entropy associated with a single bit of the internal random number.
82. It is assumed that each individual bit of the internal random number string contains the same amount of min-entropy.

Min-Entropy per Raw Random Number Bit

83. This term quantifies the min-entropy associated with a single bit of the raw random number.
84. It is assumed that each individual bit of the raw random number string contains the same amount of min-entropy.

Mixing Function

85. This function takes raw random numbers as input and updates the entropy pool when it is required. If the entropy pool coincides with the internal state, the mixing function is functionally equivalent to the seed and reseed procedures of the DRNG. This term is specific to the CCN-NPTRNG functionality class.

Noise Source

86. The component of a TRNG that implements the inherently non-deterministic, entropy-producing process. The raw random numbers are the output of the noise source.
87. This document distinguishes between two types of noise sources, depending on the corresponding functionality class: physical noise sources, in the context of

the CCN-PTRNG functionality class, and non-physical noise sources, in the context of the CCN-NPTRNG functionality class.

- a) Physical Noise Source:** A physical noise source is a source of randomness that exploits physical phenomena (thermal noise, shot noise, jitter, metastability, radioactive decay, etc.) from dedicated hardware designs (using diodes, ring oscillators, etc.) or physical experiments to produce digitized random data.

The noise source is composed of an Analog Noise Source in combination with a digitizer.

- b) Non-Physical Noise Sources:** A non-physical noise source is a source of randomness that leverages nondeterministic system behavior or external interactions, such as interrupt timings, user input events, or network activity, to generate digitized random data. Although these sources are not explicitly designed for randomness generation and lie outside the NPTRNG boundary, they are actively exploited by the NPTRNG to harvest entropy.

A non-physical noise source is a source of randomness that leverages nondeterministic system behavior or external interactions, such as interrupt timings, user input events, or network activity, to generate digitized random data. Although these sources are not explicitly designed for randomness generation and lie outside the NPTRNG boundary, they are actively exploited by the NPTRNG to harvest entropy.

Due to their complexity and dependence on external and often unpredictable factors, non-physical noise sources cannot typically be modeled with sufficient precision using formal stochastic models. As a result, designers shall rely on heuristic analysis to justify the unpredictability of their output and to derive a conservative lower bound on the available entropy.

Online Test

88. A statistical test applied to the raw random numbers (or to random data from earlier stages of the generation process) while the PTRNG is in operation. Its objective is to detect in a timely manner any unacceptable degradation of entropy, typically caused by changes in the noise source behavior or in the environmental conditions. Upon detecting such degradation, the system must raise an alarm and prevent the output of insufficiently random data.
89. Online Test is tailored to the specific stochastic model of the PTRNG and operates over fixed-size windows of consecutive random values. Due to the finite sample size, these tests are subject to statistical limitations, such as false positives (false alarms) and false negatives (silent failures). Therefore, the choice of window size and decision thresholds must balance prompt detection with statistical reliability.
90. The effectiveness of the Online Test is defined by its ability to distinguish between normal operation (model parameters within acceptable bounds) and degraded states (parameters outside of those bounds), and to do so early enough to avoid the use of low-entropy outputs in cryptographic functions.

Operational Conditions

91. Operational conditions are the specific usage modes, procedures and technical requirements needed to ensure that a PTRNG or NPTRNG operates as intended and produces the declared amount of entropy. They define how the operational environment—including hardware platforms, software, and the surrounding system architecture—must be configured and protected and specify the required user privileges. Additionally, they establish the admissible range of environmental conditions under which (usually) the PTRNG can function properly.
92. For NPTRNGs, these conditions are stricter because entropy depends heavily on the software environment, OS, and host security, requiring conservative worst-case estimates. PTRNGs rely primarily on physical hardware parameters, which are more stable.
93. Operational conditions become part of user guidance. Designers must provide evaluators with the operational conditions under which the PTRNG or NPTRNG meets its security and entropy requirements.
94. Aside from environmental conditions—which refer to external physical factors—operational conditions also encompass controllable system parameters such as enabled features, clock configurations, power modes, and interface selections.

Operational Environment

95. In the context of PTRNGs, operational environment refers to the specific hardware platform, firmware components, and surrounding system architecture within which the PTRNG is embedded and operates. The operational environment can influence the PTRNG's performance through factors such as noise coupling.
96. In the context of NPTRNGs, the operational environment refers to the specific computing platform and execution context in which the NPTRNG operates, described in terms of hardware architecture, processor model, system configuration, and operating system.
 - This includes, where applicable, the operating system and version, virtualization or hypervisor layers, and relevant hardware features enabled or disabled (e.g., specific instruction set extensions). For NPTRNGs, the operational environment is especially critical, as entropy generation relies directly on the behavior and availability of resources from this environment.
 - Example: *FreeBSD 11.4 on VMware ESXi 7 on Intel® Xeon® E5-2680 v4 (Broadwell) without PAA.*

Output Function, ψ

97. In the context of CCN-DRNGs functionality class, it refers to a function that generates internal random numbers from the current internal state in response to a request.

Personalization String

- 98. An optional input value to a DRNG during the instantiation process to make one RNG instance behave differently from other instances.
- 99. Can be a secret parameter or public parameter.

Pseudo-Random Bits

- 100. Pseudo-random bits are generated by a deterministic algorithm that, while not truly random, appear random and unpredictable to any observer without knowledge of the internal state or seed of the generator.

Raw Random Numbers

- 101. Raw random numbers are digitalized discrete values (usually bits, bit strings, or integers) that are derived at discrete points in time from a noise source of a PTRNG or NPTRNG. Raw random numbers have not been post-processed; they are the output of the digitization component of the noise source.

Raw Random Signal

- 102. The raw random signal is the output of the analog noise source. In other words, it represents the initial stage of the random numbers, prior to the digitization.

Real Data

- 103. In the context of the CCN-PTRNG functionality class, this term refers to empirical data gained from the physical noise source (analog data like voltage or timing variations, raw random numbers, etc.) that shall be used to develop, confirm, and adjust the claimed stochastic model or subclaims thereof. The entire range of admissible environmental conditions shall be considered.

Request

- 104. Operation of a DRNG to generate a requested amount of random numbers that is completed by the update of the state transition function.
- 105. Upon receiving an external request from the consuming application, the DRNG outputs the desired number of random bits. This operation usually consists of concatenating shorter internal random numbers produced by a “core function” to a larger sequence. Many DRNG designs only produce a sequence of internal random numbers of a fixed bit size. Depending on the size of the internal random numbers in bits, the DRNG internally generates one or several internal random numbers to piece together the desired number of random bits.
- 106. Other DRNG designs, such as the NIST-approved Deterministic random bit generators (DRBGs), allow for generating random bits of different lengths.

Reseeding Procedure

- 107. Refreshing the internal state of a DRNG with sufficient entropy to allow recovery from a possible compromise.

108. After reseeding, even if an attacker knew the seed material, it shall not significantly be easier to determine the new internal state than the previous one. Usually, it would be a little easier to determine the new internal state, e.g., because the seed material and the old internal state are hashed.

Security Strength

109. Security strength is usually expressed according to the attack complexity required to break a concrete cryptographic mechanism; therefore, an attack complexity of n bits means that it takes 2^n operations to perform the attack.
110. The size of the effective internal state is related to the security strength of the DRNG.

Seed

111. To initialize the internal state of a DRNG with seed material. The seed material should contain sufficient entropy (sufficient randomness if generated by a DRNG) to meet the security requirements.

Seed Entropy

112. The entropy contained in the seed material used for the seeding procedure or the reseeding procedure (if applicable).

Seedlife

113. The period between (re-)seeding the internal state of a DRNG and subsequent reseeding with new seed material, or between (re-)seeding the internal state of a DRNG and uninstating the DRNG.

Seed Material

114. A bit string containing entropy (containing randomness if generated by a DRNG) that is used for the seeding procedure or reseeding procedure a DRNG to initialize the internal state of a DRNG.
115. The seed material should contain sufficient entropy (sufficient randomness if generated by a DRNG) to meet the security requirements.

Seeding Procedure

116. Procedure for seeding (initialization) the internal state of a DRNG.

Self-Test Known Answer Tests (Self-Test KATs)

117. Self-tests in the form of Known Answer Tests (KATs) are automatically executed by the TOE under specific conditions to verify the correct functioning of its cryptographic mechanisms. These tests consist of computing outputs for predefined inputs and comparing them against expected results to ensure the integrity of the implementation before use.

Set A_{bad}

118. The set A_{bad} is defined as the complement of A_{good} ; that is, it comprises the set of values of the physical parameters and design parameters of the stochastic model for which the min-entropy per internal random bit is not sufficient.

119. Note 18 [Good Parameters] provides a formal definition.

Set A_{good}

120. The set A_{good} comprises the set of values of the physical parameters and design parameters of the stochastic model for which the min-entropy per internal random bit is sufficient.

121. Note 18 [Good Parameters] provides a formal definition.

Set A_{real}

122. The subset of parameter values from the stochastic model that are expected to occur during normal operation of the PTRNG across the entire admissible range of environmental conditions. A_{real} represents the typical behavior of the TOE in practice and shall be a subset of A_{good} , ensuring that the entropy requirements are met across all expected real-world scenarios.

123. Note 25 [Real Parameters] provides a formal definition.

Start-Up Test

124. A test executed immediately after powering up or resetting the PTRNG, before any internal random number is output. Its goal is to detect catastrophic failures or severe statistical weaknesses in the entropy source prior to normal operation.

125. The Start-Up Test ensures that the noise source and associated components are functioning correctly and that no faults have occurred during initialization. It is typically implemented by adapting the Online Test and the Total Failure Test to the Start-Up context, applying adjusted evaluation criteria appropriate for the transient conditions.

State Transition Function, ϕ

126. A function that updates the internal state during the life cycle of the DRNG.

Statistical Inconspicuousness

127. "Statistical inconspicuousness" means that it shall not be possible to distinguish sequences of internal random numbers that are generated by a DRNG from output sequences of an ideal RNG by statistical tests.

Stochastic Model

128. A stochastic model provides a partial mathematical description (of the relevant properties) of a physical noise source using random variables. It allows the verification of a lower entropy bound for the raw random numbers during the lifetime of the physical RNG. The stochastic model is based on and justified by an understanding of the noise source and validated for the concrete physical implementation of the noise source.

129. This document divides the stochastic model into two parts in order to facilitate its understanding and evaluation: the analog stochastic model, which corresponds to the analog noise source, and the digital stochastic model, which corresponds to the entire physical noise source, including the digitization component. The latter coincides with the overall stochastic model.

a) Analog Stochastic Model: The analog stochastic model provides a mathematical characterization—based on random variables—of the analog noise source, capturing the relevant properties of the raw random signal generation process. This model shall begin at the physical phenomenon that gives rise to randomness and extends up to, but not including, the digitization stage. In other words, it describes how the raw random signal emerges from the underlying physical mechanism, without relying on any derived or post-processed signals.

The analog stochastic model provides a mathematical characterization—based on random variables—of the analog noise source, capturing the relevant properties of the raw random signal generation process. This model shall begin at the physical phenomenon that gives rise to randomness and extends up to, but not including, the digitization stage. In other words, it describes how the raw random signal emerges from the underlying physical mechanism, without relying on any derived or post-processed signals.

It is acceptable to model only the components that contribute to entropy (i.e., the truly random noise components), provided it is demonstrated that any omitted effects—such as global deterministic or correlated noise—do not significantly diminish the overall entropy.

Note 9 [Analog Stochastic Model Definition] provides a formal mathematical description of the analog stochastic model.

b) Digital Stochastic Model: The digital stochastic model provides a mathematical description—based on random variables—of the noise source that spans from the underlying physical phenomenon up to the point just before post-processing, capturing the effects of digitization. It does not include any derived or processed output beyond the raw random numbers. In essence, it characterizes the generation process of raw random numbers, grounded in a clear understanding of the physical noise source and its digitization. Ideally, the digital stochastic model consists of a family of probability distributions that contains the true distribution of the digital noise source output (raw random numbers).

The digital stochastic model provides a mathematical description—based on random variables—of the noise source that spans from the underlying physical phenomenon up to the point just before post-processing, capturing the effects of digitization. It does not include any derived or processed output beyond the raw random numbers. In essence, it characterizes the generation process of raw random numbers, grounded in a clear understanding of the physical noise source and its digitization. Ideally, the digital stochastic model

consists of a family of probability distributions that contains the true distribution of the digital noise source output (raw random numbers).

The Digital Stochastic Model integrates the Analog Stochastic Model with the Digitization Function, forming a unified mathematical description of how the physical noise source produces raw random numbers through the digitization process.

Unless explicitly stated otherwise, when "stochastic model" is mentioned in this methodology, it refers to the Digital Stochastic Model.

Note 12 [Digital Stochastic Model Definition] provides a formal mathematical description of the Digital stochastic Model.

(Time-Local) Stationary

130. A sequence of random variables is considered to be "time-local" stationarily distributed if this sequence may be viewed as stationarily distributed at least over "short" timescales (in absolute time) that are, however, "large" compared to periods needed to generate samples for the Online Test and the tester tests.

131. This is an AIS-31 specific term.

Total Failure Test

132. A mechanism designed to detect complete and typically permanent failures of the noise source during the operation of the PTRNG, ensuring that internal random numbers with effectively zero entropy are not produced.

133. This test must be capable of identifying specific failure modes of the noise source, based on a prior failure mode analysis. It shall detect any such condition as early as possible and prevent any affected output from being used. Typical implementations rely on either physical measurements (e.g., absence of expected jitter or voltage variations) or statistical tests that identify patterns indicative of a non-functioning entropy source.

134. Due to the abrupt entropy loss that characterizes total failures, they can usually be detected with high reliability and low false positive rates. The test must be tailored to the concrete behavior of the digitized output under such failure conditions, such as constant or highly repetitive sequences.

Truly Random Noise Component

135. Stochastic elements of the noise source that originate from well-understood physical phenomena and exhibit inherent unpredictability. These components are responsible for the actual entropy in the output of the PTRNG and can be characterized by random variables with measurable or estimable statistical properties.

136. In contrast to global deterministic noise components, truly random noise components are uncorrelated and cannot be predicted or reproduced, even under identical operating conditions. Their behavior must be justifiable through

physical or electrical arguments and supported by a stochastic model that links their presence to the entropy observed in the generated raw random signal.

With Memory

137. Property of a post-processing algorithm. It means that the post-processing is stateful, i.e., has a state that retains information from previous invocations or steps.

2 PHYSICAL TRUE RANDOM NUMBER GENERATORS

2.1 OBJECTIVE

138. This section defines the security requirements, evaluation inputs, terminology, and evaluation tasks applicable to the CCN-PTRNG functionality class.
139. The requirements defined in this section are based on those of the PTG.2 functionality class from AIS-31, as specified in the mathematical and technical reference [FCRNG], and have been adapted following CCN's selection criteria and evaluation procedures. The evaluation tasks have been designed so that any CCN-PTRNG successfully evaluated under this methodology can be considered PTG.2 compliant.
140. Additionally, specific requirements and evaluation procedures from [SP 800-90B] have been incorporated. Therefore, any CCN-PTRNG successfully evaluated under this methodology greatly facilitates the compliance with [SP 800-90B] as a physical entropy source.

2.2 CCN-PTRNG DEFINITION

141. A Physical True Random Number Generator (PTRNG) is a type of True Random Number Generator that relies on dedicated hardware that has been designed to extract entropy from a physical random phenomenon.
142. A CCN-PTRNG is a PTRNG that has been successfully evaluated under the present methodology.

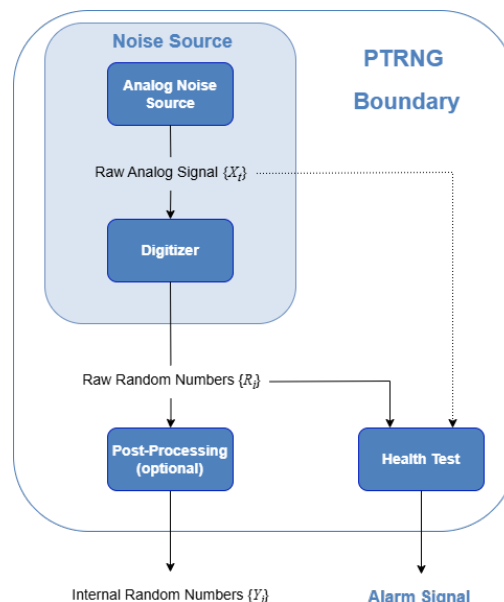


Figure 1. CCN-PTRNG Design Scheme

143. As indicated in Figure 1, four functional blocks—analogue noise source, digitizer, post-processing, and Health Tests—are contained within the CCN-PTRNG

boundary. Moreover, the analog noise source block and the digitizer together form a larger block identified as the noise source.

Noise Source

144. The Noise Source exploits an inherently unpredictable physical phenomenon to extract their stochastic behavior, generating a digitalized signal that exhibits randomness.
145. Examples of physical phenomena include thermal noise, shot noise, jitter, metastability, and radioactive decay. These may originate from dedicated hardware designs (e.g., using diodes, logic gates, or ring oscillators) or from physical experiments (e.g., measuring radioactive decay times). The dedicated hardware designs can use general-purpose components if the designer is able to understand, describe, and quantify the characteristics of the design that are relevant for the generation of random numbers.
146. The physical noise source consists of the combination of two well-defined blocks: the Analog Noise Source block and the digitizer.
 - a) **Analog Noise Source:** This component produces raw random signals—analog waveforms that arise from a physical source of randomness (e.g., thermal noise, shot noise). These signals are continuous and not directly used for mathematical modeling. Instead, the analog stochastic model defines a set of physical random variables X_t , to abstract and represent key properties of these signals. The variables X_t do not correspond to the full signal but capture specific, physically grounded aspects of it that are essential to model its stochastic behavior (e.g., timing jitter, amplitude fluctuations). They serve as the probabilistic foundation for subsequent analysis and entropy estimation.
 - b) **Digitizer:** This is the digitization mechanism component. The digitizer digitalizes the raw random signal to obtain raw random numbers, whose corresponding random variables are denoted by R_i . It is important that the chosen digitalization process does not alter the statistical properties of the raw random signal, unlike a post-processing, which intentionally modifies these properties.
147. The behavior of the physical random variables shall be modeled by the Analog Stochastic Model described in Note 9 [Analog Stochastic Model Definition].
148. The behavior of the raw random numbers must be modeled by the Digital Stochastic Model described in Note 12 [Digital Stochastic Model Definition]. The Digital Stochastic Model must cover the entire process from the origin of the randomness in the physical phenomena to the generation of the raw random numbers, with the aim of gaining complete understanding of the statistical properties of the raw random numbers.
149. Thus, the Digital Stochastic Model is defined as the composition of the Analog Stochastic Model with the digitalization function described in Note 11

[Digitization Function], together describing the entire raw random numbers generation process.

150. The Digital Stochastic Model depends on two sets of characteristic parameters that define the noise source:

- The Design Parameters, denoted by $\{q_i\}_{i=1}^m$, consist of fixed parameters related to the specific design of the noise source (e.g., sampling rates, comparator thresholds, digitizer resolution). These parameters shall be adjustable during the design phase of the noise source and shall not be manipulable by an attacker.
- The Physical Parameters, denoted by $\{p_i\}_{i=1}^n$, comprises physical or statistical quantities associated with the truly random components of the physical random variables and are directly linked to the underlying physical phenomena. These parameters must be measurable or estimable.

Health Tests

151. Variations in parameters over time—due to aging effects, changing environmental conditions, or even minor manufacturing defects, no matter how insignificant—can cause deviations from the expected behavior of the analog noise source. Thus, Health Tests are an essential component of a CCN-PTRNG, as they are designed to detect these inadvertent failures and unintended weaknesses in the analog noise source. Three types of Health Tests are considered: Start-Up Test, Online Test, and Total Failure Test.

152. Health Tests shall be tailored to the CCN-PTRNG design. Specifically, Online Test shall be tailored to the stochastic model. This implies that these tests shall be designed to detect whether the values of the stochastic model parameters ($\{p_i\}_{i=1}^n$ and $\{q_i\}_{i=1}^m$) lie within the set A_{good} , or if they have deviated from it, thus generating non-tolerable entropy defects. On the other hand, the Total Failure Test shall be supported by a detailed analysis of the noise source failure modes.

Post-Processing Component

153. Once the raw random numbers are generated and tested by the Health Tests, they may be subjected to a post-processing transformation. The post-processing function applies a deterministic transformation to the raw random numbers to produce internal random numbers, thereby enhancing their statistical properties. The corresponding random variables to the internal random numbers, denoted as Y_i , represent the final stage of the random variables describing the entire PTRNG process.

2.3 INPUTS

154. In order to simplify the revision of the documentation and reduce the evaluation effort, this section defines the set of Inputs that the vendor shall provide for the

evaluation of a Physical True Random Number Generator (PTRNG) according to the present methodology.

155. The amount and variety of evidence required to support the evaluation of a CCN-PTRNG is significant. Thus, to structure this evidence, this methodology groups the required evidence into well-defined inputs.
156. Although an Input may be composed of multiple documents or files, the **vendor shall** clearly organize and present them in a way that allows testers to easily understand the content and locate each required element. For each Input, the **vendor shall** include a main document that summarizes its content and clearly points to all the associated materials.
157. Given the challenges associated with evaluating PTRNGs, the tester may request additional evidence. In such cases, the **vendor shall** supply these supplementary materials.

I_TDS: Technical Design Specification

158. This input shall provide a structured and high-level description of the PTRNG design. It shall present the main functional blocks, their interactions, and the boundary of the TOE.
159. The specification serves as the basis for the design review and shall be coherent with the rest of the inputs. It shall allow testers to assess the completeness and soundness of the design without requiring access to low-level implementation details.
160. In particular, this input shall include the concrete evidence necessary to fulfill each requirement established in [PTRNG.01] and its associated specific notes.

I_IMP: Implementation Evidence

161. Low-level implementation artifacts of the PTRNG shall be provided, including all relevant files (e.g., HDL descriptions, gate-level netlists, layouts, or equivalent) that reflect the actual hardware realization of the TOE.
162. The materials delivered in this input shall enable a direct comparison with the functional description in **I_TDS**, allowing the tester to verify that the final implementation faithfully instantiates the declared architecture and does not introduce undocumented deviations.
163. This input shall include the concrete evidence necessary to fulfill each requirement established in [PTRNG.02] and its associated specific notes.

I_SM: Stochastic Model

164. The design of a PTRNG critically depends on the stochastic behavior of its noise source. This input shall include the complete stochastic model developed by the vendor to describe the behavior of the TOE, which serves as the foundation for its evaluation.
165. To ensure that randomness is modeled from its physical origin, the stochastic modeling shall be structured in two parts: an Analog Stochastic Model that

characterizes the analog noise source based on the physical phenomenon, and a Digital Stochastic Model that incorporates the digitization process to describe the generation of raw random numbers.

166. In addition to the models themselves, this input shall include a method to derive a conservative lower bound for the min-entropy per raw random number, consistent with the assumptions and structure of the Digital Stochastic Model. The bound may be obtained through analytical derivation or technically justified simulation.
167. The input shall also include the identification of the sets of the stochastic model parameter space for which the entropy of the internal random numbers meets the required threshold. These sets shall be formally defined based on the instantiated model, the entropy estimation method, and the effect of the post-processing function.
168. In particular, this input shall include the concrete evidence necessary to fulfill each requirement established in [PTRNG.03], [PTRNG.04], [PTRNG.05], [PTRNG.06], and their associated specific notes.

I_VAL: Stochastic Model Validation Report

169. This input shall provide the experimental and statistical evidence necessary to validate the stochastic model defined in **I_SM**. It shall demonstrate that the model's assumptions are consistent with the actual behavior of the noise source and that its parameters have been estimated from real data under representative operational conditions.
170. It shall also include all procedures, tools, and datasets used to justify that the stochastic model provides a conservative and defensible estimate of the min-entropy per raw random number bit. This includes empirical validation of model assumptions, estimation of physical parameters, and attempts to refute the entropy claim through tailored statistical methods.
171. All in all, this input shall include the concrete evidence necessary to fulfill each requirement established in [PTRNG.07], [PTRNG.08], [PTRNG.09], and their associated specific notes.

I_IRN: Output Entropy and Post-Processing Justification

172. This input shall contain all the evidence required to justify the entropy and statistical quality of the internal random numbers produced by the TOE. It shall include the rationale and validation supporting that the internal random numbers achieve a minimum min-entropy per bit of 0.98, as required by the evaluation methodology.
173. The documentation shall clearly describe the post-processing function applied to the raw random numbers, including its algorithmic structure and justification. Whether the function is algorithmic or cryptographic, the corresponding assumptions, analytical arguments, and supporting statistical results shall be provided to demonstrate that the required entropy level is maintained. The

tester shall be able to verify that the implemented post-processing matches the declared design and preserves the security guarantees.

174. In particular, this input shall include the concrete evidence necessary to fulfill the requirements established in [PTRNG.10], [PTRNG.11] and their associated specific notes.

I_HEALTH: Health Testing Evidence

175. This input shall include the documentation and evidence necessary to demonstrate that the Start-Up Test, Online Test, and Total Failure Test are correctly implemented and meet the requirements of this methodology. It shall describe each test's purpose, design, and rationale, with a focus on their ability to detect critical failures or entropy degradation.
176. The input shall justify the alignment of the Online Test with the stochastic model and include simulation methods and results supporting its detection performance. The Total Failure Test design shall be linked to identified failure modes, and the Start-Up Test shall be shown to block output until successful execution.
177. Finally, the alarm logic shall be described, ensuring that the TOE does not output any internal random number if a failure is detected.
178. This input shall support all requirements established in [PTRNG.12], [PTRNG.13], [PTRNG.14], [PTRNG.15] and their associated specific notes.

I_TEST: Black-Box Testing Evidence

179. This input shall contain the results and datasets associated with the black-box statistical test suites applied to the TOE's random outputs, in accordance with [SP 800-90B] and AIS-31. These tests serve as a general sanity check and complement the model-based analysis performed in other evaluation tasks.
180. The documentation shall specify the operational conditions under which each dataset was generated, covering the entire admissible range of environmental conditions. It shall describe how the tests were executed, the tools or scripts used, and the configuration parameters. The statistical testing evidence shall include, at minimum: a table with the numerical results for each test or estimator, the full datasets to which the tests were applied, a description of the test procedures, the software testing environment used, and proof that the dataset was captured from the correct output of the TRNG.
181. To support the credibility of the results, the vendor shall perform a witnessing session for the tester, demonstrating the data collection process used for black-box statistical testing. This session shall confirm that the datasets were collected under the admissible range of environmental conditions and correspond to the correct output of random numbers from the TOE.
182. In particular, this input shall include the concrete evidence necessary to fulfill the requirements established in [PTRNG.16], [PTRNG.17], and their associated specific notes.

2.4 CCN-PTRNG EVALUATION TASKS

183. This section defines the specific requirements for Physical True Random Number Generators, as established by CCN, which must be successfully evaluated for the TOE to be recognized as a CCN-PTRNG approved RNG.
184. The following list of evaluation tasks related to the CCN-PTRNG requirements have been defined:

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-DESIGN
The tester shall verify, using I_TDS and I_IMP , that the TOE architecture is composed of the required CCN-PTRNG functional blocks, and that the physical implementation of the design faithfully corresponds to the declared functional description.	

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-MODEL
The tester shall review the stochastic model of the noise source (I_SM), and verify that it is mathematically sound, physically grounded, and provides a justified estimation method for the min-entropy per raw random number bit.	

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-MODELVAL
The tester shall validate, using I_VAL , that the stochastic model (I_SM) accurately reflects the behavior of the TOE's noise source and supports a conservative and empirically justified min-entropy per raw random number bit bound across the entire admissible range of environmental conditions.	

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-CONDITIONING
The tester shall verify, using I_IRN , that the internal random numbers generated by the TOE achieve a min-entropy per bit of at least 0.98, considering the statistical effect of the (optional) post-processing component. In addition, the tester shall verify that the post-processing function is correctly implemented, either through cryptographic evaluation (if the function is cryptographic) or by source code review (if the function is algorithmic).	

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-HEALTH
The tester shall examine I_HEALTH and verify that the TOE applies: - An Online Test tailored to the stochastic model capable of effectively detecting entropy degradations. - A Start-Up Test that prevents the generation of internal random numbers until successfully passed. - A Total Failure Test designed to detect catastrophic failures of the noise source based on a study of its failure modes.	

CCN-PTRNG EVALUATION TASK	CCN-PTRNG-BLACKBOX
The tester shall verify, using I_TEST, that the vendor has properly applied the black-box statistical tests required by AIS-31 and [SP 800-90B] to the TOE's random outputs across the entire admissible range of environmental conditions. The tester shall repeat the tests using the provided datasets to validate the reported results.	

185. For the correct completion of the evaluation tasks, each one has been divided into a set of evaluation tests. The tester shall perform the associated tests.

CCN-PTRNG Evaluation Task	Evaluation Test
CCN-PTRNG-DESIGN Design Review	CCN-PTRNG-DESIGN/BlockDesign
	CCN-PTRNG-DESIGN/HWDesign
CCN-PTRNG-MODEL Stochastic Model Analysis	CCN-PTRNG-MODEL/AnalogModel
	CCN-PTRNG-MODEL/DigitalModel
	CCN-PTRNG-MODEL/ModelEntropy
	CCN-PTRNG-MODEL/ParameterSets
CCN-PTRNG-MODELVAL Stochastic Model Validation	CCN-PTRNG-MODELVAL/Assumptions
	CCN-PTRNG-MODELVAL/ParameterEstimation
	CCN-PTRNG-MODELVAL/EntropyValidation
CCN-PTRNG-CONDITIONING Post-Processing Evaluation	CCN-PTRNG-CONDITIONING/InternalEntropy
	CCN-PTRNG-CONDITIONING/Conformity
CCN-PTRNG-HEALTH Health Testing Review	CCN-PTRNG-HEALTH/OnlineTest
	CCN-PTRNG-HEALTH/TotalFailure
	CCN-PTRNG-HEALTH/StartUp
	CCN-PTRNG-HEALTH/Alarms
CCN-PTRNG-BLACKBOX Statistical Testing	CCN-PTRNG-BLACKBOX/AIS31
	CCN-PTRNG-BLACKBOX/90B

Table 1. Evaluation Tests defined for CCN-PTRNG Requirements

2.4.1 EVALUATION TESTS DEFINITION

186. This subsection defines the specific evaluation tests for Physical True Random Number Generators, as established by CCN, which must be performed by the tester to complete the evaluation tasks in accordance with the specific CCN-PTRNG requirements.

2.4.1.1 PTRNG DESIGN

187. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-DESIGN** evaluation task.

2.4.1.1.1 Functional Design Review

- The **TOE** design **shall** [PTRNG.01] include a functional design description which follows the specific notes listed in the table below:

Functional Design Review	
CCN Class	Notes
CCN-PTRNG	Note 1
	Note 2
	Note 3
	Note 4
	Note 5
	Note 6

Table 2. Functional Design Review – Notes

Note 1 [Functional Design Description]

The functional design description shall:

- Clearly identify and describe the blocks that constitute the required structure of a CCN-PTRNG (see the specific notes below) and explicitly define the boundary of the TOE.
- Describe the internal operation of each block and its high-level interactions with the rest (data and control flows), providing sufficient detail to understand their joint contribution to the generation of internal random numbers that fulfill the requirements of this evaluation.
- Specify the operational conditions under which the PTRNG is expected to operate, covering the entire admissible range of environmental conditions and the concrete operational environment.
- Ensure consistency with the rest of the evaluation inputs.

Note 1 [Functional Design Description]

- Enable traceability with the implementation evidence provided in I_IMP.

Note 2 [Analog Noise Source]

The functional design description shall clearly identify the Analog Noise Source as the component that harnesses the physical phenomenon which generates the inherent unpredictability of the PTRNG.

This component shall not perform any digitization; its output is the raw random signal.

Note 3 [Digitizer]

The functional design description shall clearly identify the digitizer as the component that transforms the raw random signal into the raw random numbers.

No post-processing component beyond the required quantization itself shall be implemented as part of the digitizer.

The Analog Noise Source, together with the digitizer, shall be identified as the Noise Source.

Note 4 [Post-Processing Block]

The functional design description shall clearly identify the Post-Processing Block as the deterministic component aimed at enhancing the statistical properties of the internal random numbers.

It shall take as input the raw random numbers generated by the Noise Source and produce Internal Random Numbers.

If no post-processing is used, this component shall be explicitly identified as an identity transformation.

Note 5 [Health Tests Block]

The functional design description shall clearly identify the Health Tests Block as the component implementing Start-Up Test, Online Test, and the Total Failure Test.

Note 5 [Health Tests Block]

These tests shall operate on the raw random numbers or on realizations of analog random variables from the analog signal.

In no case shall these tests be applied to the Internal Random Numbers.

Note 6 [Interfaces Identification]

The functional design description shall clearly identify and describe all interfaces of the TOE.

Internal interfaces shall be defined for each connection between the main blocks (Analog Noise Source, Digitizer, Post-Processing Block, Health Tests Block).

External interfaces shall include, at minimum:

- Power Interface, employed to power the TOE, which may influence the noise components.
- Output Interface, which provides the internal random numbers to the consuming application.
- Any Testing Interface which allows to gather data from the TOE for evaluation purposes.

The method of interaction with each external interface shall be described.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-DESIGN/BlockDesign	Inputs
Review I_TDS to confirm that it provides a functional description identifying all CCN-PTRNG blocks and interfaces which comply to the specific notes listed in Table 2.	I_TDS

2.4.1.1.2 Hardware Design Review

188.A formal description of the design alone is not sufficient, since by nature, physical true random number generators (PTRNGs) critically depend on their specific physical implementation. Therefore, implementation-level evidence must be provided to substantiate that the described architecture is correctly and consistently instantiated.

- The **TOE design shall** [PTRNG.02] be supported by implementation evidence (**I_IMP**) that:
 - Includes the full implementation files of the TOE architecture (e.g., HDL, netlist, layout, or equivalent).

- Demonstrates that the actual instantiation of the PTRNG architecture faithfully corresponds to the functional design description provided in **I_TDS**.
- The **tester shall** perform the following evaluation test:

CCN-PTRNG-DESIGN/HWDesign	Inputs
Review I_IMP to verify that the actual instantiation of the PTRNG architecture faithfully corresponds to the functional design description provided in I_TDS. The tester shall identify any inconsistencies or undocumented elements. Note: If the tester considers that the evidence provided in I_IMP is insufficient to verify that the hardware implementation is consistent with the design description, the tester may request concrete enhanced evidence.	I_IMP I_TDS

2.4.1.2 STOCHASTIC MODEL ANALYSIS

189. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-MODEL** evaluation task.

2.4.1.2.1 Analog Stochastic Model

- The **TOE shall** [PTRNG.03] rely on an analog noise source whose behavior can be modeled by an Analog Stochastic Model $M(t, p_1, \dots, p_n)$ which follows the specific notes listed in the table below:

Analog Stochastic Model	
CCN Class	Notes
CCN-PTRNG	Note 7 Note 8 Note 9 Note 10

Table 3. Analog Stochastic Model – Notes

Note 7 [Physical Phenomenon]
The physical phenomenon exploited by the analog noise source shall be clearly identified as the origin of randomness. The Analog Stochastic Model shall be substantiated using physical or electrical engineering arguments related to this physical phenomenon.

Note 8 [Deterministic Noise]

The final aim of the Stochastic Model is to provide an entropy lower bound based only on the truly random, well-characterized noise components.

Thus, global deterministic or highly correlated contributions that do not add unpredictability—or for which no solid stochastic modelling exists in literature—shall be excluded from the model.

As an example, in ring oscillator-based noise sources, only the thermal noise component of the jitter is known to be physically justified and modeled in the literature.

Note 9 [Analog Stochastic Model Definition]

The Analog Stochastic Model shall be defined as a stochastic process that adheres to the following general formal definition:

$$M(t, p_1, \dots, p_n) := \{X_t\}_{t \in T}$$

Where:

- $T \subseteq \mathbb{R}$ is the time domain (discrete or continuous).
- X_t , the “physical random variables”, are random variables over a state space V , selected to directly model the physical random phenomenon exploited by the noise source.

They shall represent non-digitalized and physically grounded quantities that evolve stochastically as a result of the identified source of randomness.

- The “physical parameters”, $\{p_i\}_{i=1}^n$, are physical or statistical quantities associated with the truly random component of $\{X_t\}$ and shall:
 - Be directly linked to the physical phenomenon as reflected in the distributions of X_t .
 - Be measurable or estimable.

All the listed elements shall be clearly identified.

All in all, the objective of the Analog Stochastic Model is to provide a physically justified probabilistic description of the analog noise source, ensuring that the model originates directly from the real physical phenomenon stage and not from any derived or processed signal.

Note 10 [Assumptions]

All assumptions on which the model is built should be clearly listed.

In particular:

- Any assumption related to the random variables X_t (e.g., X_t are iid, are distributed according to $\mathcal{D}$...).
- Any assumption related to the knowledge that, under the Kerkhoff hypothesis, a potential attacker might have.

Even in the case that the Analog Noise Source is based on a well-studied model from the literature, all assumptions shall be identified and a justification that the design of the analog noise source is consistent with them shall be provided.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODEL/AnalogModel	Inputs
Verify that I_SM includes a stochastic model for the analog noise source and assess its mathematical and logical consistency. To do so, the tester shall: 1. Verify that I_SM includes an Analog Stochastic Model that explicitly covers the analog noise source employed by the TOE. 2. Verify that I_SM complies with the specific notes provided in Table 3; in particular, the tester shall verify that the model meets the definition given in Note 9 and that each required item is explicitly identified. 3. Analyze whether the Analog Stochastic Model is based on previously studied models and whether it cites relevant state-of-the-art sources that support the appropriateness of the modeling approach. 4. Compare the model presented in I_SM with well-studied models for the same type of analog noise sources as the one employed by the TOE. 5. Verify that all mathematical steps and derivations supporting the Analog Stochastic Model are sound and correctly followed and explicitly identify any incorrect, missing, or unjustified elements.	I_SM

CCN-PTRNG-MODEL/AnalogModel	Inputs
Based on the previous steps, the tester shall accept or reject the Analog Stochastic Model and assess the impact of this verdict on the rest of evaluation tests defined in this methodology. Note: This test has been described in a detailed and stepwise manner. However, the tester is expected to conduct the test with a critical mindset. In particular, the use of a widely accepted stochastic model in the literature for the specific type of noise source employed by the TOE and the tester's experience should facilitate the review without compromising the depth of the analysis.	

2.4.1.2.2 Digital Stochastic Model

- The **TOE shall** [PTRNG.04] generate raw random numbers that can be interpreted as realizations of a (time-local) stationary stochastic process, modeled by a Digital Stochastic Model $D(p_1, \dots, p_n, q_1, \dots, q_m)$, which follows the specific notes listed in the table below:

Digital Stochastic Model	
CCN Class	Notes
CCN-PTRNG	Note 11 Note 12 Note 13 Note 14 Note 15

Table 4. Digital Stochastic Model – Notes

Note 11 [Digitization Function]
The digitization function defining the operation of the digitizer component of the noise source shall be mathematically described in a way which conforms to the following general formal definition: Let, $\{X_t\}_{t \in T}$ be the stochastic process defined in the Analog Stochastic Model, with $\mathbb{Q} \in \mathbb{Q}$. $X_i^{(k)} := (X_{t_i}, X_{t_{i-1}}, \dots, X_{t_{i-k}}) \in V^{k+1}$ be a tuple of consecutive values from the Analog Stochastic Process. Then, the digitization function is defined as: $f_{\text{dig}}: V^{k+1} \times \mathcal{Q} \rightarrow \mathcal{R}$

Note 11 [Digitization Function]

so that,

$$R_i = f_{\text{dig}}(X_i^{(k)}, \{q_j\})$$

where:

- The “design parameters”, $\{q_i\}_{i=1}^m \subseteq \mathcal{Q}$, are related to the concrete design of the noise source (e.g., sampling rates, comparator thresholds, digitizer resolution) and shall:
 - Be adjustable in the design phase of the noise source.
 - Not be manipulable by the attacker.
- $R_i \in \mathcal{R}$ are the raw random numbers, and $\mathcal{R}$ is the raw random number alphabet (raw random numbers can consist of single bits, ℓ -bit vectors, or even integers).

All the listed elements shall be clearly identified and defined.

Note 12 [Digital Stochastic Model Definition]

The Digital Stochastic Model shall be defined as the composition of the Analog Stochastic Model (see Note 9) and the digitization function (see Note 11) and models the raw random numbers generation process.

Formally:

$$D(p_1, \dots, p_n, q_1, \dots, q_m) := \{R_i\}_{i \in \mathbb{N}} \text{ with } R_i = f_{\text{dig}}(X_i^{(k)}, \{q_j\})$$

The Digital Stochastic Model shall model the sequence of raw random numbers as a discrete-time stochastic process, obtained by respecting the formulation of the Analog Stochastic Model—potentially defined over a continuous time domain—and by applying the digitization function as defined.

Note 13 [Auxiliary Processes]

The Digital Stochastic Model – and, in particular, the digitization function – may internally rely on intermediate stochastic processes, beyond the primary physical process $\{X_t\}_{t \in T}$, to formally structure the transformation leading to the raw random numbers $\{R_i\}_{i \in \mathbb{N}}$ stochastic process.

These auxiliary stochastic processes, (e.g., $\{Z_t\}_{t \in T}, \{W_t\}_{t \in T}$) may not correspond to directly measurable quantities, but may serve to define a logical chain of transformations with clearer formalism and enabling the proper application of probability theory.

Note 13 [Auxiliary Processes]

In that case, the digitization process could thus be represented as a composition of functions such that:

$$f_{\text{dig}} = f_n \circ f_{n-1} \circ \dots \circ f_1$$

leading to a sequence of stochastic processes:

$$\{X_t\} \rightarrow^{f_1} \{W_t\} \rightarrow^{f_2} \{Z_t\} \rightarrow \dots \rightarrow \{R_j\}$$

If employed within the model, any auxiliary stochastic process shall be clearly identified, and its relevant properties shall be clearly stated.

Note 14 [Avoiding Post-Processing]

The digitization process shall not introduce deterministic components beyond the required quantization itself; it shall not include adaptive logic, memory elements, or any form of post-processing (e.g., hash functions, XORs, LFSRs).

Note 15 [PTG.2.1]

The TOE shall generate raw random numbers that can be viewed as realizations of a (time-local) stationary stochastic process $R_1, R_2, \dots$

If the raw random numbers are not binary-valued but ℓ -bit vectors, it may be the case that the random variables that describe the random behavior of the ℓ -bit vectors are time-locally stationarily distributed, but not the individual bits.

Thus, the Digital Stochastic Model shall ensure this requirement.

Note: Time-local stationarity ([FCRNG] specific term) means that the raw (or auxiliary) random variables can be treated as stationary over short time intervals that are large compared to the sample sizes used in evaluation and Online Test, allowing approximate stationarity despite possible long-term non-stationarity.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODEL/DigitalModel	Inputs
Examine the Digital Stochastic Model I_SM and assess whether it provides a mathematically and logically consistent description of the raw random number generation process, in accordance with the formal definition and specific notes above. To do so, the tester shall:	I_SM

CCN-PTRNG-MODEL/DigitalModel	Inputs
1. Ensure that the CCN-PTRNG-MODEL/AnalogModel evaluation test has been successfully performed, and the Analog Stochastic Model has been accepted. 2. Verify that I_SM includes a mathematical definition of the digitization function that explicitly covers the digitizer employed by the TOE. 3. Verify that the model complies with the specific notes provided in Table 4. In particular, the tester shall verify that the Digital stochastic Model is defined as the composition of the Analog Stochastic Model and the digitization function. 4. Verify that the digitization function is appropriate for the Analog Stochastic Model. 5. Analyze whether the Digital Stochastic Model is based on previously studied models and whether it cites relevant state-of-the-art sources that support the appropriateness of the modeling approach. 6. Compare the I_SM model with well-studied models for the same type of noise sources as the one employed by the TOE. 7. Verify that all mathematical steps and derivations supporting the Digital Stochastic Model are sound and correctly followed and explicitly identify any incorrect, missing, or unjustified elements. Based on the previous steps, the tester shall accept or reject the Digital Stochastic Model and assess the impact of this verdict on the rest of evaluation tests defined in this methodology. Note: This test has been described in a detailed and stepwise manner. However, the tester is expected to conduct the test with a critical mindset. In particular, the use of a widely accepted stochastic model in the literature for the specific type of noise source employed by the TOE and the tester's experience should facilitate the review without compromising the depth of the analysis.	

2.4.1.2.3 Raw Entropy Estimation Method

- The **TOE shall** [PTRNG.05] generate raw random numbers that can be modeled by a Digital Stochastic Model $D(p_1, \dots, p_n, q_1, \dots, q_m)$, from which a method to derive a conservative lower bound, h_{rrn}^{model} , for the min-entropy per raw random

number shall be defined, and which shall comply with the specific notes listed in the table below:

Raw Entropy Estimation Method	
CCN Class	Notes
CCN-PTRNG	Note 16 Note 17

Table 5. Raw Entropy Estimation Method – Notes

Note 16 [Entropy Bound Formulation]

The min-entropy lower bound h_{rrn}^{model} , shall be derived using either an explicit mathematical formulation or a clearly defined algorithmic procedure, and shall be consistent with the structure and assumptions of the Digital Stochastic Model.

For example, if the model assumes statistical dependencies (e.g., Markovian or higher-order structures on the raw random numbers), it is expected to employ conditional formulations rather than expressions that assume independence.

This lower bound formulation shall explicitly depend on the parameters of the model, including both the physical parameters $\{p_i\}_{i=1}^n$ and the design parameters $\{q_i\}_{i=1}^m$.

Note 17 [Entropy Bound Computation]

The computation method for the lower bound h_{rrn}^{model} shall be carried out either through analytical derivation or via stochastic simulation. In the latter case, the simulation method shall be clearly described and technically justified.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODEL/ModelEntropy	Inputs
Verify, using I_SM, that: - From the Digital Stochastic Model, a method to derive a conservative lower bound, h_{rrn}^{model}, for the min-entropy per raw random number has been defined. - The described method complies with the specific notes listed in Table 5.	I_SM

2.4.1.2.4 Parameter Sets Definition

- The **TOE shall** [PTRNG.06]:
 - Be accompanied by the definition of the sets A_{good} and A_{bad} , where A_{good} contains all the values of the digital Stochastic Model parameters $\{p_i\}_{i=1}^n$ and $\{q_i\}_{i=1}^m$ for which the-min entropy per internal random number bit is greater or equal to 0.98 and A_{bad} is its complement.
 - Comply with the specific note listed in the table below:

Parameter Sets Definition	
CCN Class	Notes
CCN-PTRNG	Note 18

Table 6. Parameter Sets Definition – Notes

Note 18 [Good Parameters]
The sets A_{good} and A_{bad} shall be defined by the following (or an equivalent) procedure: 1. Instantiate the Digital Stochastic Model with concrete values of the physical parameters $\{p_i\}_{i=1}^n$ and with the fixed values for the design parameters $\{q_i\}_{i=1}^m$. 2. Apply the method discussed in section 2.3.1.2.4 Raw Entropy Estimation Method, to derive h_{rrn}^{model}. 3. Use the post-processing characterization compliant to requirements in section 2.4.1.4.1 Internal Random Numbers Entropy to obtain h_{irn}. 4. Check if $h_{irn} \geq 0.98$. Then, A_{good} is defined as the set of physical and design parameters for which $h_{irn} \geq 0.98$. The set A_{bad} is its complement. Since the design parameters are fixed for a given implementation, A_{good} shall be defined as a subset of R^n, where n is the number of different physical parameters employed by the stochastic modeling.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODEL/ParameterSets	Inputs
Verify, using I_SM , that: - The sets A_{good} and A_{bad} are properly established. - The TOE complies with the specific note listed in Table 6.	I_SM I_IRN I_HEALTH

CCN-PTRNG-MODEL/ParameterSets	Inputs
Verify the consistency with the references to these sets and the employed methods in I_IRN and I_HEALTH.	

2.4.1.3 STOCHASTIC MODEL VALIDATION

190. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-MODELVAL** evaluation task.
191. The validation of the stochastic model can be considered the most critical part of the evaluation of a PTRNG. A proper validation of the stochastic model confirms that it fits the actual design and physical implementation of the noise source.
192. Because of the wide range of possible noise sources, stochastic models and implementation technologies, it is not feasible to define a fully prescriptive, ready-to-use checklist for validating the stochastic model. However, the current task has been designed to serve as a minimum applicable to any particular case, while also facilitating the work of vendors and testers.
193. All in all, the objective of the vendor is to demonstrate the tester that the proposed stochastic model does indeed describe the actual implementation of the noise source. The tester's objective is to examine the vendor's validation strategy and accept or reject it.
194. Considering the impact of environmental conditions on the behavior of the noise source, the following note applies to the entire task:

Note 19 [Validation Conditions]

The validation of the stochastic model shall consider the influence of the entire admissible range of environmental conditions on the behavior of the noise source and the stability of the model physical parameters.

For this, empirical data collected across the entire admissible range of environmental conditions shall be used for the validation of the stochastic model.

Thus, for the concrete operational environment:

- The entire range of admissible environmental conditions shall be taken into account (e.g., temperature, supply voltage). Of course, sampling may be applied, but extreme conditions shall be considered.
- Manufacturing process variations shall be addressed by analyzing multiple copies of the TOE.
- System-level influences (e.g., surrounding circuitry) and aging effects may also be considered where relevant.

Note 19 [Validation Conditions]

The validation test plan shall cover all the above-mentioned requirements.

2.4.1.3.1 Stochastic Model Assumptions Validation

- The **TOE shall** [PTRNG.07] base the design of its noise source on a stochastic model whose underlying assumptions have been experimentally validated, and the validation process shall comply with the following notes:

Validation of Assumptions	
CCN Class	Notes
CCN-PTRNG	Note 19
	Note 20
	Note 21
	Note 22

Table 7. Validation of Assumptions – Notes

Note 20 [Pre-silicon Design Validation]

The stochastic model selected or developed for the noise source shall have been considered during its design phase.

Ideally, the model should exist prior to the concrete implementation of the entropy source, serving as a design guide.

Evidence shall be provided showing how the stochastic model has guided design decisions. A stochastic model developed retrospectively for an already implemented entropy source is strongly discouraged.

Note 21 [Analog Model Assumptions]

All assumptions underlying the analog Stochastic Model shall be experimentally validated.

Examples, without loss of generality, include:

- Properties of the process $\{X_t\}_{t \in T}$ (e.g., whether the variables are iid, the process is Markovian, is stationary, is q-dependent...),
- Distributional assumptions (e.g., $X_t \sim \mathcal{D}$ for a known $\mathcal{D}$).

For this purpose, appropriate statistical hypothesis testing techniques shall be employed.

Note 22 [PTG.2.1 Validation]

The (time-local) stationarity property of the $\{R_i\}_{i \in N}$ process shall be supported by empirical results. This shall be done by analyzing autocorrelation and autocovariance computed either from the raw random numbers themselves, or from realizations of the physical variables X_t – provided that the raw random numbers inherit this property from the process $\{X_t\}_{t \in T}$.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODELVAL/Assumptions	Inputs
Verify, using I_VAL that the assumptions considered in I_SM are consistent with the design and the implementation of the noise source, and have been experimentally validated. To do so, the tester shall: 1. Verify, using I_VAL, that I_SM complies with the specific notes provided in Table 7. 2. Assess, using I_VAL, the reproducibility of the validation process performed by the vendor. To this end, the tester shall: a. Review the statistical procedures, tools, and data sets provided in I_VAL. - Repeat at least one set of validation procedures, using a representative sample of the procedures in I_VAL. 3. If deemed necessary, perform additional validation checks using independent tools or methods employing the datasets provided in I_VAL.	I_SM I_VAL

2.4.1.3.2 Physical Parameters Estimation

195. To ensure consistency between the Analog Stochastic Model and the physical implementation, the model shall be instantiable using values obtained from the TOE itself. To this end:

- The **TOE shall** [PTRNG.08] be based on a noise source for which the physical parameters $\{p_i\}_{i=1}^n$ of the Analog Stochastic Model $M(t, p_1, \dots, p_n)$ have been experimentally estimated, and follow the specific notes listed in the table below:

Parameter Estimation	
CCN Class	Notes
CCN-PTRNG	Note 19 Note 23 Note 24 Note 25

Table 8. Parameter Estimation – Notes

Note 23 [Noise Component Separation]

The estimation of the physical parameters $\{p_i\}_{i=1}^n$ shall consider only the truly random noise components explicitly modeled in the Analog Stochastic Model. (see Note 9). Deterministic or highly correlated components shall be excluded from the estimation process.

The separation of components during the estimation process shall be clearly described, and, where applicable, established methods from the literature (e.g., thermal jitter separation in ring oscillator-based noise sources) shall be referenced or technically justified.

Note 24 [Estimation Methods]

The physical parameters $\{p_i\}_{i=1}^n$ shall be experimentally estimated using technically justified methods. These may include:

- External measurement techniques (e.g., oscilloscope-based signal capture)
- Internal measurement techniques (e.g., signal monitoring within the device).
- Inference from statistical test results from appropriate variables.

The estimation process shall describe the employed measurement techniques, including devices, sensitivity, number of measurements per quantity as well as statistical techniques.

The measurement errors associated with the estimated parameters shall be evaluated.

The stability of the estimated parameters shall be assessed across the entire range of admissible environmental conditions (see Note 19).

All methods and assumptions shall be clearly documented and justified.

The estimation procedures shall be reproducible using the employed data, tools, and descriptions.

Note 25 [Real Parameters]

Based on the employed parameter estimation procedures, the set A_{real} shall be established as the subset of A_{good} containing the physical parameter values that are expected to occur when the TOE is in operation across the entire admissible range of environmental conditions.

It shall be verified that $A_{real} \subseteq A_{good}$. In particular, the boundary between A_{real} and A_{good} shall be identified and it shall be demonstrated that no values within A_{real} fall outside A_{good} .

Note: Design parameters, which are fixed, shall also be considered.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODELVAL/ParameterEstimation	Inputs
Verify, using I_VAL that the physical parameters in I_SM have been experimentally estimated. To do so, the tester shall: 1. Verify, using I_VAL, that I_SM complies with the specific notes provided in Table 8. 2. Assess, using I_VAL, the reproducibility of the estimation process performed by the vendor. To this end, the tester shall: a. Review the experimental and statistical procedures, tools, and data sets provided in I_VAL. b. Repeat at least one set of estimation procedures, using a representative sample of the procedures in I_VAL. If deemed necessary, perform additional validation checks using independent tools or methods employing the datasets provided in I_VAL.	I_SM I_VAL

2.4.1.3.3 Modeled Entropy Claim

- The **TOE shall** [PTRNG.09]:
 - Be accompanied by instantiated values of the min-entropy per raw random number bit estimate h_{rrn}^{model} across the entire admissible range of environmental conditions, computed using the defined method from the digital Stochastic Model.
 - Comply with the specific notes listed in the table below:

Modeled Entropy Claim	
CCN Class	Notes
CCN-PTRNG	Note 19 Note 26

Table 9. Modeled Entropy Claim – Notes

Note 26 [Prediction Estimate]
The min-entropy per raw random number bit lower bound estimate, h_{rrn}^{model}, as derived from the Digital Stochastic Model, shall be validated by demonstrating that it cannot be refuted through an appropriate statistical procedure. Specifically, it shall be shown that no tailored method, applied to raw random number datasets gathered from the TOE, produce an experimental lower entropy estimate, h_{rrn}^{exp} for which $h_{rrn}^{exp} < h_{rrn}^{model}$. All in all, if a predictor tailored to the structure and assumptions of the stochastic model is able to predict the raw random numbers significantly better than expected (if $h_{rrn}^{exp} < h_{rrn}^{model}$) then, the min-entropy per raw random number bit lower bound h_{rrn}^{model} is refuted. This indicates that the Digital Stochastic Model underestimates dependencies or overstates entropy.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-MODELVAL/EntropyValidation	Inputs
Verify, using I_VAL, that: - The min-entropy per raw random number bit lower bound estimate, h_{rrn}^{model}, has been derived for the entire admissible range of environmental conditions. - The TOE complies with the specific notes listed in Table 9. In order to do so, the tester shall review and repeat the method employed by the vendor to validate h_{rrn}^{model}. The tester may apply extra min-entropy predictor-based estimators.	I_VAL

2.4.1.4 POST-PROCESSING EVALUATION

196. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-CONDITIONING** evaluation task.

2.4.1.4.1 Internal Random Numbers Entropy

- The **TOE shall** [PTRNG.10]:
 - Generate internal numbers that can be interpreted as realizations of random variables $\{Y_i\}_{i \in N}$ for which the min-entropy per internal random number bit, h_{irn} , shall be greater than or equal to 0.98.
 - Comply with the following specific notes listed in the table below:

Internal Random Numbers Entropy	
CCN Class	Notes
CCN-PTRNG	Note 27 Note 28 Note 29 Note 30

Table 10. Internal Random Numbers Entropy – Notes

Note 27 [PTG.2.2]

If the random variables $\{Y_i\}_{i \in N}$ are binary-valued, it shall be $Pr(Y_j = 1) \in (0.493, 0.507)$. If the random variables $\{Y_i\}_{i \in N}$ assume binary vectors, this condition shall be met by the projections onto the particular bits.

Note 28 [Post-Processing Types]

To verify the above-mentioned requirements ([PTRNG.10] and Note 27) the effect of the post-processing function (applied to the raw random numbers) on the entropy of the internal random numbers shall be justified.

The post-processing function shall be algorithmically defined and belong to one of the following types:

- No post-processing: this requires that the raw random numbers have enough min-entropy ($h_{rrn}^{model} \geq 0.98$)
In this case, the post-processing shall be defined as the identity function.
- Algorithmic post-processing: in this case, the post-processing function can be defined as a simple mapping such as XORing consecutive raw random numbers.
- Cryptographic post-processing.

A demonstration that $h_{irn} \geq 0.98$ shall be provided. If the raw random numbers already provide sufficient entropy, it suffices to show that the post-processing function does not decrease it.

Note 28 [Post-Processing Types]

Note: The post-processing function can be defined as a composition of several post-processing steps. In that case, each step shall be analyzed on its own.

Note 29 [Algorithmic Post-Processing]

If the TOE implements an algorithmic post-processing function, it shall be tailored to the Digital Stochastic Model.

This implies that a mathematical justification - consistent with the Digital Stochastic Mode – shall be provided to demonstrate compliance to the requirements defined in [PTRNG.10] and Note 27.

Furthermore, statistical evidence shall be provided to support the above-mentioned.

Note₁: A stochastic model for the internal random numbers can be provided to support this requirement.

Note₂: Section 5.1 from [FCRNG] provides examples of algorithmic post-processing functions, as well as related justifications.

Note 30 [Cryptographic Post-Processing]

If the TOE implements a cryptographic post-processing function, the justification and evaluation effort is significantly reduced:

- The requirement stated in Note 27 shall be justified based on distributional assumptions derived from the widely recognized cryptographic assumptions in relation with the employed construction.

For example, this applies when the employed cryptographic construction can be modeled as a pseudorandom function (PRF) under standard assumptions.

- To demonstrate the effect of the cryptographic post-processing function on the entropy of the internal random numbers (requirement [PTRNG.10]), the method described in [SP 800-90B], Section 3.1.5 shall be applied considering the following:
 - The narrowest internal width, n_w , of the post-processing function shall be defined and justified based on the employed cryptographic construction.
 - The min-entropy per internal random number bit, shall be estimated as:

Note 30 [Cryptographic Post-Processing]

$$h_{irn} = \frac{h_{out}}{n_{out}}$$

where h_{out} is the min-entropy of the output as computed by the *Output_Entropy()* function defined in [SP 800-90B], and n_{out} is the bit length of the internal random numbers, as defined in the same specification.

In any case, the above-mentioned methods can be applied provided that the cryptographic construction can be successfully evaluated under the [CCN-MEMeC] evaluation framework.

Note: The tester shall accept that the post-processing function qualifies as cryptographic. For instance, constructions such as LFSRs shall not be considered cryptographic.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-CONDITIONING/InternalEntropy	Inputs
Verify, using I_IRN, that: - The TOE generates internal random numbers for which the min-entropy per bit, is greater than or equal to 0.98. - The TOE complies with the specific notes listed in Table 10. In order to do so, the tester shall review the evidence provided in I_IRN, verify the validity of the methods used to support the justification, and reproduce the results by applying those methods.	I_IRN

2.4.1.4.2 Post-Processing Conformity

- The **TOE shall** [PTRNG.11] implement the post-processing function in a faithful and verifiable manner, corresponding to the declared design.
- The **tester shall** perform the following evaluation test:

CCN-PTRNG-CONDITIONING/Conformity	Inputs
In the case of cryptographic post-processing algorithms, the tester shall ensure that a [CCN-MEMeC] evaluation at level CL2 or CL3 has been successfully performed. In the case of non-cryptographic post-processing algorithms. The tester shall perform a focused source code review of the post-	I_IRN I_TDS I_IMP

CCN-PTRNG-CONDITIONING/Conformity	Inputs
processing implementation as part of the CCN-PTRNG-DESIGN/HWDesign test.	

2.4.1.5 HEALTH TESTING REVIEW

197. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-HEALTH** evaluation task.

2.4.1.5.1 Online Test

- The **TOE shall** [PTRNG.12] implement an Online Test that checks the quality of the raw random numbers while the RNG is in operation, detects non-tolerable entropy defects of the raw random numbers sufficiently soon and complies with the following specific notes:

Online Test	
CCN Class	Notes
CCN-PTRNG	Note 31 Note 32 Note 33 Note 34

Table 11. Online Test – Notes

Note 31 [Tailored to the Stochastic Model]

The Online Test shall be tailored to the stochastic model. This means that it shall be designed to detect whether the values of the stochastic model parameters (the physical parameters $\{p_i\}_{i=1}^n$ and the design parameters $\{q_i\}_{i=1}^m$, that should be fixed by design) are inside the set A_{good} while the TOE is in operation, or have deviated from it, thus generating non-tolerable entropy defects.

In particular:

- The null hypothesis shall state that stochastic model parameters values lie within A_{good} .
- The alternative hypothesis shall state that the stochastic model parameters values lie within A_{bad} .
- To decide between these hypotheses, the Online Test shall compute, for each window of m consecutive raw random numbers, a test value $c = \xi(r^1, \dots, r_m)$ using a test function ξ specifically adjusted to the stochastic model.

Note 31 [Tailored to the Stochastic Model]

- The Online Test shall raise an alarm based on decision thresholds (cutoff values) defining the acceptance region for the test value.

Note 32 [Performance Justification]

The Online Test shall be effective in the sense that it reliably detects any non-tolerable degradation of entropy and does so early enough to prevent the generation of excessive weak random numbers.

To ensure this:

- An acceptance region $[L, U]$ shall be defined for the chosen statistic to reliably separate A_{good} and A_{bad} .
- The false alarm probability (α) and the silent failure probability (β) shall be explicitly determined and justified. These probabilities shall be determined with respect to the sample size (window) of the Online Test and shall be as low as possible.

A correct estimation of the false alarm probability (α) and the silent failure probability (β) shall involve performing representative stochastic simulations covering both: situations where the PTRNG operates with its parameters within A_{good} and scenarios where the parameters have deviated into A_{bad} . The resulting test data shall be used to estimate these probabilities and to justify the chosen acceptance region $[L, U]$ and the window size.

The estimation of both probabilities shall achieve an acceptable statistical confidence level by providing a confidence interval of at least 95%, or an equivalent uncertainty bound. The sample sizes employed for the simulation shall be justified accordingly.

Note 33 [Testing Real Parameters]

The Online Test may be designed to detect whether the parameter values while the TOE is in operation lie within A_{real} instead of the full set A_{good} , provided that it has been demonstrated that $A_{real} \subseteq A_{good}$. This can allow for more efficient Online Test.

Note 34 [Sufficiently Soon]

To ensure that any non-tolerable parameter deviation is detected sufficiently soon, the Online Test shall be executed continuously during operation of the PTRNG.

Note 34 [Sufficiently Soon]

Moreover, the size of the Online Test window shall be selected and justified to provide a balance between statistical reliability and timely detection, so that parameter drifts do not go undetected for an unacceptable duration.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-HEALTH/OnlineTest	Inputs
Verify that the evidence presented in I_HEALTH demonstrates the compliance of the Online Test with the specific notes listed in Table 11. In addition, the tester shall: - Review the provided material and results to confirm that the simulation coverage and the statistical confidence levels for the estimated false alarm probability (α) and the silent failure probability (β) are sufficient and justified. - Repeat at least one representative simulation case using the input data contained in I_HEALTH to verify the reproducibility of the vendor's results. - If deemed necessary, perform additional independent checks or simulations to confirm the claimed detection performance.	I_HEALTH

2.4.1.5.2 Total Failure Test

- The **TOE shall** [PTRNG.13] implement a Total Failure Test that detects if a total failure of the physical noise source occurs while the PTRNG is in operation and complies with the following specific notes:

Total Failure Test	
CCN Class	Notes
CCN-PTRNG	Note 35 Note 36 Note 37

Table 12. Total Failure Test – Notes

Note 35 [Immediate Detection]

The Total Failure Test shall detect a complete failure of the noise source as soon as possible, preventing the output of internal random numbers with

Note 35 [Immediate Detection]

(essentially) zero entropy. The test shall act on the raw random numbers or even before.

Note 36 [Failure Modes]

The design of the Total Failure Test shall be justified by a detailed study of the noise source failure modes. This study shall identify technically plausible failure scenarios, describe their impact on the raw random signal and the resulting internal random numbers entropy, and explain how each scenario is reliably detected by the Total Failure Test.

As an example (for a ring oscillator-based noise source): a ring oscillator may become locked due to aging or supply voltage failure, producing a static output signal with zero oscillation frequency.

Note 37 [Implementation]

The Total Failure Test shall be implemented by means of physical measurements (e.g., voltage, current, jitter) or statistical tests, provided they are sufficient to detect a complete entropy loss condition.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-HEALTH/TotalFailure	Inputs
Verify that the evidence presented in I_HEALTH demonstrates the compliance of the Total Failure Test with the specific notes listed in Table 12. If deemed necessary, perform additional independent checks or simulations to confirm the claimed detection performance.	I_HEALTH

2.4.1.5.3 Start-Up Test

- The **TOE shall** [PTRNG.14] apply a Start-Up Test after the RNG has been started. It shall be designed to detect a total failure of the physical noise source and severe statistical weaknesses. The TOE shall not output any internal random numbers before the Start-Up Test has been passed. It shall comply with the following specific notes:

Start-Up Test	
CCN Class	Notes
CCN-PTRNG	Note 38
	Note 39
	Note 40

Table 13. Start-Up Test – Notes

Note 38 [Warm-Up]

If the noise source requires a warm-up period after powering up, the Start-Up Test shall be applied after the warm-up period.

Note 39 [Detection Objective]

The Start-Up Test shall cover detection of a total failure and severe statistical weaknesses. A reasonable approach is to apply the Online Test and the Total Failure Test with (possible) adjusted evaluation rules suitable for start-up conditions. In this case, those adjustments shall be justified in line with the nature of the tests.

If alternative test methods are used, they shall meet equivalent requirements for failure detection and statistical weakness coverage as the Online Test and the Total Failure Test.

Note 40 [Known Answer Test]

The Start-Up Test shall include a Known Answer Test (KAT) for each deterministic component of the PTRNG.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-HEALTH/StartUp	Inputs
Verify, using I_HEALTH, that the PTRNG does not output any internal random number until the Start-Up Test has passed with success. Verify that the evidence presented in I_HEALTH demonstrates the compliance of the Start-Up Test with the specific notes listed in Table 13. If deemed necessary, perform additional independent checks or simulations to confirm the claimed detection performance.	I_HEALTH

2.4.1.5.4 Health Testing Alarms

- The **TOE shall** [PTRNG.15] implement alarm logic associated with the failure conditions detected by the Start-Up Test, Online Test, and Total Failure Test, and shall comply with the specific notes in the table below:

Health Testing Alarms	
CCN Class	Notes
CCN-PTRNG	Note 41

Table 14. Health Testing Alarms – Notes

Note 41 [Health Testing Alarms]
An alarm signal shall be generated when the Total Failure Test or the Online Test detect a failure (as defined for each of them). - In the case of Online Test alarms, it shall be ensured that no internal random numbers with insufficient entropy are output. - In the case of a Total Failure Test alarm, the TOE shall prevent the output of internal random numbers that depend on any raw random number that has been generated after the total failure of the physical noise source. For the Start-Up Test, the alarm logic works inversely: the TOE shall only permit the output of internal random numbers once a signal confirms that the test has been successfully passed. In any case, it shall be proven that the alarm logic for the Start-Up Test, Online Test, and Total Failure Test is correctly implemented and ensures the required blocking of internal random numbers as specified. The appropriate response to an alarm depends on the deployment environment and the system's criticality. Note: It is not an acceptable consequence for the noise alarm to allow the system to simply continue operation until an Online Test “passes” by chance due to a false positive.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-HEALTH/Alarms	Inputs
Verify, using I_HEALTH that alarm logic is implemented for the failure conditions detected by the Start-Up Test, Online Test, and Total Failure Test as specified. Verify, using I_HEALTH, the requirements expressed in Note 41.	I_HEALTH

2.4.1.6 BLACK-BOX STATISTICAL TESTING

198. This topic defines the evaluation tests required to fulfill the **CCN-PTRNG-BLACKBOX** evaluation task.
199. These tests apply the standard black-box statistical test suites defined in [SP 800-90B] and AIS-31. It shall be noted that the results obtained in this task serve merely as a sanity check, since black-box statistical tests alone can only be considered a necessary, but not sufficient, condition for the satisfactory evaluation of an entropy source.
200. Considering the impact of environmental conditions on the behavior of the noise source, the following note applies to the entire task:

Note 42 [Validation Conditions]

The black box statistical test suites shall be applied under all admissible range of environmental conditions.

All datasets shall be extracted from the correct interface (raw random numbers or internal random numbers) during normal operation and shall consist of consecutive samples.

2.4.1.6.1 AIS-31 Test Suite

- The **TOE shall** [PTRNG.16] generate internal random numbers that pass the test suite T_{irn} according to the specific notes below:

AIS-31 Testing	
CCN Class	Notes
CCN-PTRNG	Note 42 Note 43 Note 44

Table 15. AIS-31 Testing – Notes

Note 43 [AIS-31 Test Suite]

The T_{irn} from AIS-31, as defined in section 4.6.4 of the technical reference [FCRNG], shall be employed.

Note 44 [Evaluation Rules]

For each dataset (corresponding to each value of the admissible range of environmental conditions), the test suite requires 2040000 bits.

Note 44 [Evaluation Rules]

The internal random bits for each dataset shall be acquired during normal operation and should be consecutive.

The following rules apply:

- The T_{irn} black box test suite is passed if all individual statistical tests have been passed.
- If at least one individual statistical test has failed, the T_{irn} test suite is failed.
- If the T_{irn} black box test suite has failed, it shall be applied a second time. If the second trial is successful, the T_{irn} black box test suite is viewed as passed. Another repetition of the T_{irn} black box test suite is not allowed.
- Rules (a) to (c) apply separately for all admissible range of environmental conditions, for which T_{irn} is applied.

All the results, datasets and tools shall be presented in **I_TEST**.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-BLACKBOX/AIS31	Inputs
Verify, using I_TEST, that the T_{irn} black box test suite was applied properly by the vendor, following the specific notes from Table 15 and that the internal random numbers successfully pass the test suite. Additionally, validate the results by repeating all tests using the data provided in I_TEST.	I_TEST

2.4.1.6.2 SP 800-90B Test Suite

- The **TOE shall** [PTRNG.17] generate raw random numbers and internal random numbers that pass the entropy estimation and statistical testing procedures described in [SP 800-90B] and follow the specific notes below:

SP 800-90B Test Suite	
CCN Class	Notes
CCN-PTRNG	Note 42 Note 45 Note 46

Table 16. SP 800-90B Test Suite– Notes

Note 45 [Block Assessment]

In order to obtain a black box min-entropy estimate per raw random number bit, h_{rrn}^{block} :

- All the min-entropy estimators described in section 6.2 of the [SP 800-90B] document shall be applied. The lowest estimator value shall be credited as h_{rrn}^{test} .
- The above-mentioned shall be repeated for different datasets generated under all admissible range of environmental conditions.
- Each dataset shall contain at least 1,000,000 consecutive samples of raw random numbers collected during normal operation.

The same process shall be performed over the internal random numbers, in order to produce a black box min-entropy estimate per internal random number bit, denoted as h_{irn}^{block} , according to the requirements presented in [SP 800-90B].

All the results, datasets and tools shall be presented in **I_TEST**.

Note 46 [Restart Assessment]

The restart tests defined in Section 3.1.4 of the [SP 800-90B] document shall be applied to the raw random numbers generated by the noise source.

The restart process employed during testing shall reflect the expected behavior of the device in real-world use, including any initialization delays, startup procedures, or waiting periods declared in the operational documentation.

All restarts shall be performed under all the admissible range of environmental conditions (e.g., room temperature, nominal voltage).

All the results, datasets and tools shall be presented in **I_TEST**.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-BLACKBOX/90B	Inputs
Verify, using I_TEST , that the [SP 800-90B] entropy estimation procedures were applied properly by the vendor, following the specific notes from Table 16.	I_TEST
Additionally, validate the results by repeating all tests using the data provided in I_TEST and the official [SP 800-90B] testing tool.	I_VAL
	I_IRN

CCN-PTRNG-BLACKBOX/90B	Inputs
Compare (using I_VAL , I_IRN and I_TEST) the min-entropy per bit estimation values produced by NIST estimators with the values of h_{rrn}^{model} and h_{irn} as a sanity check.	

3 HYBRID PHYSICAL TRUE RANDOM NUMBER GENERATORS

3.1 OBJECTIVE

201. This section defines the security requirements, evaluation inputs, terminology, and evaluation tasks applicable to the CCN-HPTRNG functionality class.
202. The requirements defined in this section are based on those of the PTG.3 functionality class from AIS-31, as specified in the mathematical and technical reference [FCRNG]. They have been adapted according to CCN's selection criteria and evaluation methodology, taking into account the "typical design" referenced in [FCRNG] for PTG.3, where the inner PTRNG is PTG.2 compliant.
203. However, the CCN-HPTRNG is more than the sum of its individual components. Therefore, this section defines the specific requirements and evaluation tasks that apply to the composite design of the TOE, addressing the correct interaction and operation of the CCN-PTRNG and CCN-DRNG components as a unified construction; the CCN-HPTRNG.

3.2 CCN-HPTRNG DEFINITION

204. A Hybrid Physical True Random Number Generator (HPTRNG) is a type of Physical True Random Number Generator (PTRNG) that includes cryptographic post-processing with memory, based on a DRNG.
205. A CCN-HPTRNG is a HPTRNG that has been successfully evaluated under the present methodology.
206. The general structure of the CCN-HPTRNG is shown in Figure 2. It indicates that a CCN-HPTRNG consists of the combination of an inner PTRNG and a DRNG post-processing component, both of which must comply with the CCN-PTRNG and CCN-DRNG functionality classes, respectively. Consequently, the evaluation of a CCN-HPTRNG necessarily involves the evaluation of both a CCN-PTRNG and a CCN-DRNG.

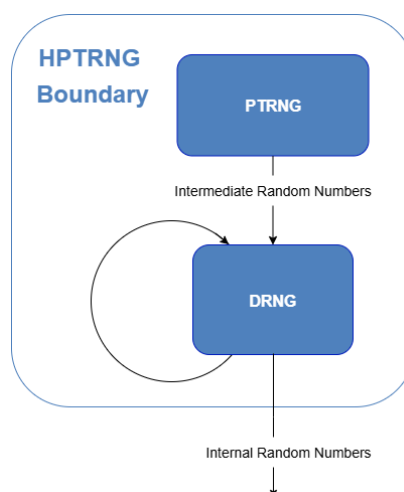


Figure 2. CCN-HPTRNG Design Scheme

207. The goal of this functionality class is to ensure computational security in addition to the information-theoretical security provided by its physical noise source (as in CCN-PTRNG). This is achieved through the following requirements:

- Using a stateful cryptographic post-processing component compliant to CCN-DRNG class, as opposed to the (optional) simple algorithmic post-processing used in CCN-PTRNG functionality class. The DRNG post-processing component may be viewed as an additional security anchor for a situation where the entropy from the noise source per output bit is smaller than assumed.
- Data compression to increase entropy per bit. The CCN-HPTRNG class requires that $c_{rate} > 1$. Unlike in CCN-EDRNG, the cryptographic post-processing in CCN-HPTRNG cannot extend its input data, its role is strictly to enhance the quality of the already present entropy.

3.3 INPUTS

208. In order to simplify the revision of the documentation and reduce the evaluation effort, this section defines the set of Inputs that the vendor shall provide for the evaluation of a Hybrid Physical True Random Number Generator (HPTRNG) according to the present methodology.

209. The amount of evidence required to support the evaluation of a CCN-HPTRNG is significantly reduced compared to other classes, as most of the evaluation relies on the existing evaluations of the underlying CCN-PTRNG and CCN-DRNG components. Nonetheless, to structure the evidence relevant to the composite design, this methodology defines a single specific Input.

210. Given the challenges associated with evaluating HPTRNGs, the tester may request additional evidence. In such cases, the **vendor shall** supply these supplementary materials.

211. Some requirements of this functionality class must be demonstrated in the context of the evaluations of the CCN-PTRNG and CCN-DRNG components that form part of the TOE. Therefore, the evidence related to such requirements shall be provided within the corresponding Inputs of the evaluations of the CCN-PTRNG and CCN-DRNG integrated into the TOE. In particular:

- The evidence required to demonstrate compliance with [HPTRNG.02] and its associated specific note shall be included in the **I_HEALTH** input of the internal CCN-PTRNG evaluation, as defined in Section 2.3 “Inputs”.
- The evidence required to demonstrate compliance with [HPTRNG.03] and its associated specific notes shall be included in the **I_VQ_DRNG** (or **I_90A**) input of the evaluation of the CCN-DRNG implemented by the TOE, as defined in Section 5.3 “Inputs”.

I_HPTRNG: Composite Design Evidence

212. This input shall include the design and implementation evidence necessary to demonstrate how the CCN-PTRNG and CCN-DRNG components are integrated within the TOE as a composite construction. It shall provide an identification of the employed PTRNG and DRNG, a clear and structured description of the roles of each component, their internal interface, and how they operate together to generate internal random numbers.
213. In particular, the input shall include the concrete evidence required to fulfill the requirement established in [HPTRNG.01] and its associated specific note. This includes a clear identification of both components, a description of the interface and interaction logic between them, and implementation-level evidence (e.g., annotated code, interface specifications) that reflects the actual integration.
214. Moreover, this input shall be consistent with, and reference, the provided evidence for requirements [HPTRNG.02] and [HPTRNG.04] which, as explained, shall be provided in the context of the internal CCN-PTRNG and CCN-DRNG evaluations.

3.4 CCN-HPTRNG EVALUATION TASKS

215. This section defines the specific requirements for Hybrid Physical Random Numbers Generators, as established by CCN, which must be successfully evaluated for the TOE to be recognized as a CCN-HPTRNG approved RNG.
216. The following list of evaluation tasks related to the CCN-HPTRNG requirements have been defined:

CCN-HPTRNG EVALUATION TASK	CCN-HPTRNG-DESIGN
The tester shall review I_HPTRNG to verify that the TOE is designed as a composite construction integrating an inner PTRNG acting as entropy source and a DRNG post -processing component, and that the interface between both components is clearly described and faithfully implemented.	

CCN-HPTRNG EVALUATION TASK	CCN-HPTRNG-SOURCE
The tester shall perform an evaluation of the inner PTRNG component in accordance with the present methodology to confirm that it belongs to the CCN-PTRNG functionality class or verify that the TOE uses a previously certified CCN-PTRNG to generate intermediate random numbers.	

CCN-HPTRNG EVALUATION TASK	CCN-HPTRNG-POSTPROCESSING
The tester shall perform an evaluation of the DRNG post-processing component in accordance with the present methodology to confirm that it belongs to the CCN-DRNG functionality class. Additionally, the tester shall verify that: - Intermediate random numbers are properly injected into the DRNG via seeding (at instantiation) and via reseeding and/or additional input for entropy refresh. - Compression is explicitly defined, guarantees that the DRNG does not expand its input, and is effectively implemented.	

217. For the correct completion of the evaluation tasks, each one has been divided into a set of evaluation tests. The tester shall perform the associated tests.

CCN-HPTRNG Evaluation Task	Evaluation Test
CCN-HPTRNG-DESIGN Design Review	CCN-HPTRNG- DESIGN/CompositeDesign
CCN-HPTRNG-SOURCE Entropy Source Evaluation	CCN-HPTRNG-SOURCE/PTRNG
CCN-HPTRNG-POSTPROCESSING Post-Processing Evaluation	CCN-HPTRNG- POSTPROCESSING/DRNG
	CCN-HPTRNG- POSTPROCESSING/EntropyInjection

Table 17. Evaluation Tests defined for CCN-HPTRNG Requirements

3.4.1 EVALUATION TESTS DEFINITION

218. This subsection defines the specific evaluation tests for Hybrid Physical True Random Number Generators, as established by CCN, which must be performed by the tester to complete the evaluation tasks in accordance with the specific CCN-HPTRNG requirements.

3.4.1.1 HPTRNG DESIGN

219. This topic defines the evaluation tests required to fulfill the **CCN-HPTRNG-DESIGN** evaluation task.

3.4.1.1.1 Composite Design

- The **TOE shall** [HPTRNG.01]:
 - Be designed as a composite construction that integrates an inner PTRNG and a DRNG post-processing component.
 - Comply with the specific note listed in the table below:

Composite Design	
CCN Class	Notes
CCN-HPTRNG	Note 47

Table 18. Composite Design – Notes

Note 47 [Integration Evidence]
The design and implementation evidence shall describe how the PTRNG and the DRNG components are integrated within the TOE. This shall include: ▪ A clear identification of both components and their roles. ▪ An indication of whether they are implemented on the same platform and technology or not (e.g., the PTRNG is HW-based, but the DRNG component can be implemented in software). ▪ A description of the internal interface between the components and how they interact at the logical level. ▪ Implementation (source code) evidence for the internal interface.

- The **tester shall** perform the following evaluation test:

CCN-HPTRNG-DESIGN/CompositeDesign	Inputs
Review I_HPTRNG in order to: - Verify that it provides a consistent and complete description and implementation evidence of the integration between the CCN-PTRNG and CCN-DRNG components that form the TOE. - Confirm that the composite structure is clearly explained in the design documentation, including the interface between both components, and that the actual implementation corresponds faithfully to this description. - Verify that the evidence complies with the specific note listed in Table 18.	I_HPTRNG

3.4.1.2 ENTROPY SOURCE EVALUATION

220. This topic defines the evaluation tests required to fulfill the **CCN-HPTRNG-SOURCE** evaluation task.

3.4.1.2.1 PTRNG Evaluation

- The **TOE shall** [HPTRNG.02] employ a CCN-PTRNG-compliant PTRNG to generate the intermediate random numbers that are input into the DRNG post-processing component and comply with the specific note listed in the table below:

PTRNG Evaluation	
CCN Class	Notes
CCN-HPTRNG	Note 48

Table 19. PTRNG Evaluation – Notes

Note 48 [Inner PTRNG Health]
The health tests implemented by the inner CCN-PTRNG shall take into account that the CCN-DRNG post-processing component acts as the consuming application of the intermediate random numbers. In particular, the DRNG shall be treated as an entropy buffer that may hold some residual entropy even after the failure of the physical noise source. Therefore, alarms raised by the inner PTRNG Health Test must trigger output control logic at the CCN-HPTRNG level. In particular, in the case of the Total Failure Test, after a total failure: - If no intermediate random numbers are available, the TOE shall not output any further internal random numbers.

Note 48 [Inner PTRNG Health]

- Alternatively, if intermediate random numbers are available and - taking into account the role of the DRNG's effective internal state - the TOE shall output at most t' bits, where t' is the amount of entropy remaining in the DRNG's internal state prior to failure.

This note shall be evaluated (and the vendor shall provide evidence for it) in the context of the **CCN-PTRNG-HEALTH** evaluation task of the inner PTRNG employed by the TOE.

- The **tester shall** perform the following evaluation test:

CCN-HPTRNG-SOURCE/PTRNG	Inputs
Verify that the implemented internal PTRNG is compliant with the CCN-PTRNG functionality class. Both of the following options are valid for this verification: - Perform an evaluation in accordance with the present methodology to verify compliance with the corresponding requirements of the CCN-PTRNG functionality class. - Verify that the TOE incorporates an internal PTRNG that has been previously certified as CCN-PTRNG. In this case, verify that the PTRNG has been configured in accordance with the methods, parameters and requirements established in its secure usage procedures and user guidance documentation. In addition, verify, in the context of the evaluation of the inner PTRNG, that the TOE complies with the specific note listed in Table 19.	I_TDS I_IMP I_SM I_VAL I_IRN I_HEALTH I_TEST

3.4.1.3 POST-PROCESSING EVALUATION

221. This topic defines the evaluation tests required to fulfill the **CCN-HPTRNG-POSTPROCESSING** evaluation task.

3.4.1.3.1 DRNG Evaluation

- The **TOE shall** [HPTRNG.03] employ a DRNG that is algorithmically compliant to the CCN-DRNG functionality class acting as its post-processing component which takes the intermediate random numbers as input and outputs the internal random numbers.
- The **tester shall** perform the following evaluation test:

CCN-HPTRNG-POSTPROCESSING/DRNG	Inputs
Perform tasks CCN-DRNG-IMPLEMENTATION and CCN-DRNG-ALGORITHMIC to confirm that the DRNG post-processing component implemented by the TOE complies with the corresponding requirements of the CCN-DRNG functionality class.	I_VQ_DRNG I_90A I_IMP_DRNG

3.4.1.3.2 Entropy Injection

- The **TOE shall** [HPTRNG.04]:
 - Input its intermediate random numbers into the DRNG via seeding (at instantiation) and via reseeding and/or additional input for entropy refresh, and it shall be guaranteed that the DRNG does not expand its input sequence.
 - Comply with the specific notes listed in the table below:

Entropy Injection	
CCN Class	Notes
CCN-HPTRNG	Note 49 Note 50 Note 51

Table 20. Entropy Injection – Notes

Note 49 [Compression Rate]

The compression rate of the DRNG, defined as the ratio between the number of input bits and the number of output bits, shall be explicitly determined.

The compression rate shall fulfill the following requirements:

- It shall be greater than or equal than one, ensuring that the DRNG does not expand its input.
- It shall be effective, in the sense that the number of internal random bits output after each seeding or reseeding procedure shall not exceed the size of the internal state. The internal state shall be treated as the narrowest internal width of the DRNG.

Example: The intermediate random numbers provided by the inner CCN-PTRNG comprise 4096 bits per reseeding. If the DRNG used as cryptographic post-processing has an internal state of only 256 bits, outputting 4096 internal random number bits before the next reseeding would yield a compression rate of 1, but the compression would not be considered effective. Since the

Note 49 [Compression Rate]

DRNG's internal state can only retain 256 bits of entropy, no more than 256 bits may be output after each reseeding.

Note: If the above conditions are met, the internal random numbers of the TOE can be assumed to have a min-entropy of at least 0.98 bits per bit, as guaranteed by the inner CCN-PTRNG component.

However, if a higher min-entropy claim is desired for the internal random numbers, it shall be shown that the DRNG performs effective compression up to the claimed level. For this, the DRNG shall be modeled as a composition of random mappings, and the results described in Section 4.4.2 of [FCRNG] for random mappings shall be employed to support the entropy claim.

Note 50 [Reseeding Option]

If the reseeding procedure is used to incorporate the intermediate random numbers, the state transition function of the DRNG shall be applied after the last internal random number has been generated. When the PTRNG starts, the internal state of the DRNG (used as the cryptographic post-processing component) has not yet received any entropy from the intermediate random numbers. Thus, only during start-up may a seeding procedure be used instead of a reseeding procedure.

Note 51 [Entropy Injection Justification]

The requirements defined in [HPTRNG.04] and the associated previous notes shall be justified in the context of the seeding and reseeding procedures and/or the additional input mechanism of the DRNG component used by the TOE.

Accordingly, during the evaluation of the DRNG as a CCN-DRNG in accordance with the present methodology, the vendor shall provide the necessary evidence that the DRNG complies with these requirements, and the tester shall verify such compliance within the scope of the DRNG evaluation.

- The **tester shall** perform the following evaluation test:

CCN-HPTRNG-POSTPROCESSING/EntropyInjection	Inputs
Verify that the TOE inputs its intermediate random numbers into the DRNG post-processing component via seeding (at instantiation) and via reseeding and/or additional input for	I_VQ_DRNG I_90A I_IMP_DRNG

CCN-HPTRNG-POSTPROCESSING/EntropyInjection	Inputs
entropy refresh, and that the DRNG does not expand its input sequence. In order to do so: - Perform an evaluation in accordance with the present methodology to confirm that the DRNG post-processing component implemented by the TOE belongs to the CCN-DRNG functionality class. - Verify, in the context of CCN-DRNG evaluation, that the TOE inputs its intermediate random numbers into the DRNG post-processing component by its seeding procedure at DRNG instantiation, and by reseeding procedure and/or as additional input for entropy refresh. Also, verify that the TOE complies with the specific notes listed in Table 20. - Verify, using I_HPTRNG, that the composite design and implementation evidence are consistent with the above-mentioned requirements.	I_HPTRNG

4 NON-PHYSICAL TRUE RANDOM NUMBER GENERATORS

4.1 OBJECTIVE

222. This section defines the security requirements, evaluation inputs, terminology, and evaluation tasks applicable to the CCN-NPTRNG functionality class.
223. The requirements defined in this section are based on those of the NTG.1 functionality class from AIS-31, as specified in the mathematical and technical reference [FCRNG], and have been adapted following CCN's selection criteria and evaluation procedures. The evaluation tasks have been designed so that any CCN-NPTRNG successfully evaluated under this methodology can be considered NTG.1 compliant.
224. Additionally, specific requirements and evaluation procedures from [SP 800-90B] have been incorporated. Therefore, a CCN-NPTRNG successfully evaluated under this methodology facilitates its evaluation as a non-physical entropy source in accordance with [SP 800-90B].

4.2 CCN-NPTRNG DEFINITION

225. A Non-Physical True Random Number Generator (NPTRNG) is a type of True Random Number Generator that collects entropy from non-physical noise sources, typically derived from asynchronous system events or external interactions such as user input timings, network traffic, or system-level interrupts. These sources are not intentionally designed to generate randomness but are nevertheless exploited to extract unpredictable behavior.
226. A CCN-NPTRNG is an NPTRNG that has been successfully evaluated under the present methodology.
227. As shown in Figure 3. The generic design of an NPTRNG comprises four conceptual components: two or more external non-physical noise sources, an entropy mixing function, an entropy pool with an associated entropy counter, and a post-processing component compliant with the CCN-DRNG functionality class. The non-physical noise sources lie outside the NPTRNG boundary, although they are exploited by the device on which the NPTRNG is implemented. The entropy pool may coincide with the internal state of the DRNG component, or exist as a separate component that collects and stores entropy to be periodically injected into the DRNG through reseeding.

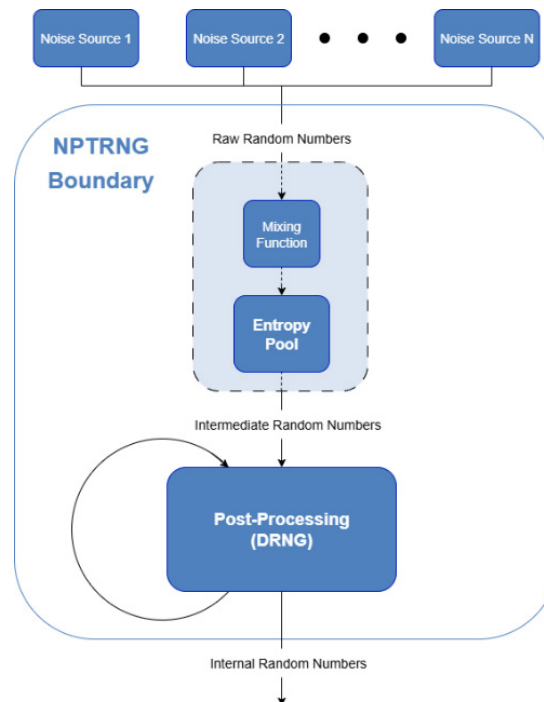


Figure 3. CCN-NPTRNG Design Scheme

228. The non-physical noise sources used by NPTRNGs are not internal components of the TOE, but rather external data sources available to the platform. While they may belong to the system hosting the NPTRNG, their behavior is not directly controlled by the designer. These sources must be individually identified and analyzed through heuristic justification, which shall demonstrate that their output is sufficiently unpredictable under the declared operational conditions. Due to their strong dependence on platform-specific configurations and external influences, the evaluation of a CCN-NPTRNG is valid only for the specified operational environment and shall not be extrapolated to others.
229. The raw random numbers collected from non-physical noise sources are combined and accumulated within an entropy pool. This process is performed through a mixing function that incorporates fresh entropy while updating the pool's state and associated entropy counter. The entropy pool serves as the reservoir from which internal random numbers are derived, either directly—if it coincides with the internal state of the DRNG component—or indirectly—by reseeding the DRNG with the accumulated entropy at defined intervals.
230. The DRNG component shall be to the CCN-DRNG functionality class. This ensures that, once properly seeded, the DRNG component acts as a reliable security anchor, providing computationally secure internal random numbers. As in the CCN-HPTRNG functionality class, it is not sufficient to merely employ a CCN-DRNG; this component must be periodically reseeded with fresh entropy to maintain the required entropy threshold. The number of internal random numbers output by the TOE shall never exceed the amount of entropy currently tracked by the entropy counter, thus ensuring that each output bit is backed by sufficient entropy input.

4.3 INPUTS

- 231. In order to simplify the revision of the documentation and reduce the evaluation effort, this section defines the set of Inputs that the vendor shall provide for the evaluation of a Non-Physical True Random Number Generator (NPTRNG) according to the present methodology.
- 232. The amount and variety of evidence required to support the evaluation of a CCN-NPTRNG is significant. Thus, to structure this evidence, this methodology groups the required evidence into well-defined inputs.
- 233. Although an Input may be composed of multiple documents or files, the **vendor shall** clearly organize and present them in a way that allows testers to easily understand the content and locate each required element. For each Input, the **vendor shall** include a main document that summarizes its content and clearly points to all the associated materials.
- 234. Given the challenges associated with evaluating NPTRNGs, the tester may request additional evidence. In such cases, the **vendor shall** supply these supplementary materials.

I_TDS_NP: Technical Design Specification for NPTRNGs

- 235. This input shall provide a structured and high-level description of the NPTRNG design. It shall describe the architecture of the TOE, clearly identifying the different non-physical noise sources, the entropy pool, and the DRNG post-processing component, along with their interactions and the internal data flows.
- 236. The specification shall explicitly define the boundary of the TOE and the operational conditions required for the noise sources to function correctly. It shall also describe the assumptions under which the claimed entropy levels hold.
- 237. This input shall include sufficient information to understand how the TOE collects and mixes entropy, how the entropy counter is maintained, and how the intermediate random numbers are delivered to the DRNG post-processing component.
- 238. In particular, this input shall include the concrete evidence necessary to fulfill each requirement established in [NPTRNG.01], [NPTRNG.02], [NPTRNG.03] and their associated specific notes.

I_IMP_NP: Implementation Evidence for NPTRNGs

- 239. This input shall provide the complete implementation evidence for the NPTRNG. It shall include the full source code of the TOE and all its components, as well as all necessary build files, including build scripts, configuration files, and any supporting materials required to compile and instantiate the TOE under the evaluated conditions.
- 240. The implementation shall faithfully correspond to the design description provided in I_TDS_NP. The tester shall be able to verify that the declared

architecture is correctly instantiated in the actual implementation and that no undocumented deviations exist.

241. In particular, this input shall include the concrete evidence necessary to fulfill each requirement established in [NPTRNG.01], [NPTRNG.02], [NPTRNG.03] and their associated specific notes.

I_ENT_NP: Entropy Justification and Heuristic Analysis for NPTRNGs

242. This input shall include the complete evidence used to justify the entropy content of the NPTRNG, structured into two parts. First, it shall provide a structured heuristic analysis for each non-physical noise source used by the TOE. Second, it shall include the derivation and justification of the min-entropy per bit of the internal random numbers, taking into account the compression effect of the DRNG post-processing component.
243. The heuristic analysis shall clearly identify the source of unpredictability in each noise source, define the operational conditions required for its correct behavior, and argue—using system-level or architectural reasoning—why an external adversary cannot significantly degrade the entropy provided. A conservative estimated value for the min-entropy per raw random number bit shall be derived from this analysis and be supported by empirical validation.
244. Additionally, this input shall provide the justification that the internal random numbers achieve an explicit estimated min-entropy per bit of at least 0.98. To support this, the **vendor shall** use the heuristic entropy estimates of the noise sources and demonstrate, based on the theoretical results of random mappings described in [FCRNG], that the DRNG post-processing achieves effective compression up to the claimed entropy level.
245. In particular, this input shall include the concrete evidence necessary to fulfill the requirements established in [NPTRNG.04], [NPTRNG.05], and their associated specific notes.

I_HEALTH_NP: Health Testing Evidence for NPTRNGs

246. This input shall include the documentation and evidence required to demonstrate that the NPTRNG implements effective Health Testing mechanisms. It shall describe the continuous tests used to monitor the quality of the raw random numbers during operation and detect entropy degradation or catastrophic failures. These tests shall follow the principles and statistical foundations established in [SP 800-90B], including either the Repetition Count Test and the Adaptive Proportion Test, or properly justified substitutes. The documentation shall detail how the failure thresholds are selected and how the statistical performance of the tests (e.g., false alarm and silent failure probabilities) is aligned with the operational characteristics of the TOE.
247. The input shall also describe the Start-Up Test, which must ensure that entropy-producing components function correctly before generating internal random numbers. It shall show that no internal output is allowed until the test is successfully passed and that the test covers the required number of samples.

The input shall also demonstrate the inclusion of a Known Answer Test (KAT) for each deterministic component of the TOE. If the continuous tests are reused with adjusted thresholds or alternative procedures during Start-Up, the rationale for such modifications shall be clearly justified.

248. Finally, this input shall describe the alarm logic associated with the Health Tests. It shall show that the TOE blocks the output of internal random numbers in the event of test failures, and that these alarms prevent the output of weak or invalid data in case of entropy loss or source malfunction. The conditions triggering the alarms, the affected outputs, and the logic enforcing blocking must be documented and supported with implementation-level evidence.
249. This input shall include the concrete evidence necessary to fulfill the requirements established in [NPTRNG.06], [NPTRNG.07], [NPTRNG.08] and their associated specific notes.

I_TEST_NP: Black-Box Testing Evidence for NPTRNGs

250. This input shall contain the results and datasets associated with the black-box statistical test suite applied to the TOE's random outputs, in accordance with [SP 800-90B]. These tests serve as a general sanity check and complement the model-based analysis performed in other evaluation tasks.
251. The documentation shall specify the conditions under which each dataset was generated, covering the declared operational environment. It shall describe how the tests were executed, the tools or scripts used, and the configuration parameters. The statistical testing evidence shall include, at minimum: a table with the numerical results for each test or estimator, the full datasets to which the tests were applied, a description of the test procedures, the software testing environment used, and proof that the dataset was captured from the correct output of the NPTRNG.
252. To support the credibility of the results, the **vendor shall** perform a witnessing session for the tester, demonstrating the data collection process used for black-box statistical testing. This session shall confirm that the datasets were collected under the declared operational conditions and correspond to the correct output of random numbers from the TOE.
253. In particular, this input shall include the concrete evidence necessary to fulfill the requirements established in [NPTRNG.09] and its associated specific notes.

4.4 CCN-NPTRNG EVALUATION TASKS

254. This section defines the specific requirements for Non-Physical Random Number Generators, as established by CCN, which must be successfully evaluated for the TOE to be recognized as a CCN-NPTRNG approved RNG.
255. The following list of evaluation tasks related to the CCN-NPTRNG requirements have been defined:

CCN-NPTRNG EVALUATION TASK	CCN-NPTRNG-DESIGN
The tester shall verify, using I_TDS_NP and I_IMP_NP , that the TOE architecture implements the required CCN-NPTRNG components, and that the software implementation faithfully corresponds to the declared design	

CCN-NPTRNG EVALUATION TASK	CCN-NPTRNG-ENTROPY
The tester shall verify, using I_ENT_NP, that each noise source credited with entropy in the TOE is supported by a structured heuristic analysis that provides a conservative estimate of the min-entropy per raw random number bit. Additionally, the tester shall verify that the internal random numbers generated by the TOE achieve a min-entropy per bit of at least 0.98, and that this estimate is justified by combining the heuristic estimates for the raw random numbers with an analysis of the DRNG component effect on the entropy.	

CCN-NPTRNG EVALUATION TASK	CCN-NPTRNG-HEALTH
The tester shall examine I_HEALTH_NP and verify that the TOE applies: - Continuous Tests capable of effectively detecting entropy degradations and total failures. - A Start-Up Test that prevents the generation of internal random numbers until successfully passed.	

CCN-NPTRNG EVALUATION TASK	CCN-NPTRNG-BLACKBOX
The tester shall verify, using I_TEST_NP , that the vendor has properly applied the black-box statistical tests required by [SP 800-90B] to the TOE's random outputs. The tester shall repeat the tests using the provided datasets to validate the reported results.	

256. For the correct completion of the evaluation tasks, each one has been divided into a set of evaluation tests. The tester shall perform the associated tests.

CCN-NPTRNG Evaluation Task	Evaluation Test
CCN-NPTRNG-DESIGN Design Review	CCN-NPTRNG-DESIGN/Sources
	CCN-NPTRNG-DESIGN/EntropyPool
	CCN-NPTRNG-DESIGN/DRNG
CCN-NPTRNG-ENTROPY Entropy Estimation	CCN-NPTRNG-ENTROPY/HeuristicAnalysis
	CCN-NPTRNG-ENTROPY/InternalEntropy
CCN-NPTRNG-HEALTH Health Testing Review	CCN-NPTRNG-HEALTH/ContinuousTest
	CCN-NPTRNG-HEALTH/StartUp
	CCN-NPTRNG-HEALTH/Alarms
CCN-NPTRNG-BLACKBOX Black Box Statistical Testing	CCN-NPTRNG-BLACKBOX/90B

Table 21. Evaluation Tests defined for CCN-NPTRNG Requirements

4.4.1 EVALUATION TESTS DEFINITION

257. This subsection defines the specific evaluation tests for Non-Physical True Random Number Generators, as established by CCN, which must be performed by the tester to complete the evaluation tasks in accordance with the specific CCN-NPTRNG requirements.

4.4.1.1 DESIGN REVIEW

258. This topic defines the evaluation tests required to fulfill the **CCN-NPTRNG-DESIGN** evaluation task.

4.4.1.1.1 Noise Sources

- The **TOE shall** [NPTRNG.01] collect raw random numbers from different non-physical noise sources and comply with the specific notes listed in the table below:

Noise Sources	
CCN Class	Notes
CCN-NPTRNG	Note 52

Table 22. Noise Sources – Notes

Note 52 [Noise Source Conditions]
A CCN-NPTRNG-compliant NPTRNG may use any data source as a noise source, provided it is subject to heuristic analysis and there is a compelling technical explanation of its unpredictability under the defined operational environment (e.g., physical platform, CPU type, virtualization constraints, entropy relevant parameters, or required security features). Note that non-physical noise sources exhibit a strong dependence on the operational environment of the TOE. Thus, the evaluation of a CCN-NPTRNG shall be performed for a concrete operational environment, and results from one shall not be extrapolated to others. The TOE may collect raw random numbers from (extra) non-evaluated noise sources, but in that case, no entropy content shall be accounted for.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-DESIGN/Sources	Inputs
Verify (using I_TDS_NP and I_IMP_NP) that: The TOE collects raw random numbers from different non-physical noise sources.	I_TDS_NP I_IMP_NP

CCN-NPTRNG-DESIGN/Sources	Inputs
- Verify that the TOE complies with the specific notes listed in Table 22.	

4.4.1.1.2 Entropy Pool

- The **TOE shall** [NPTRNG.02]:
 - Implement an entropy pool and an entropy counter that tracks the estimated amount of entropy currently stored in the entropy pool.
 - Comply with the specific notes listed in the table below:

Entropy Pool	
CCN Class	Notes
CCN-NPTRNG	Note 53 Note 54 Note 55

Table 23. Entropy Pool – Notes

Note 53 [Entropy Counter]
The entropy counter, which can be implicit or explicit, shall comply with the following conditions: - It shall be initialized to zero. - It shall be incremented by the estimated amount of entropy contributed by each block of raw random numbers mixed into the entropy pool. - It shall be decremented by the number of bits extracted from the pool as internal random numbers. - Its updates shall be consistent with the behavior of the mixing function. - Its value shall never exceed the maximum amount of entropy that the pool can store, determined by the data size of the entropy pool. - Most importantly, the TOE shall never output more internal random number bits than indicated by the current value of the entropy counter. Note: Any entropy estimation for the raw random numbers shall be derived from structured heuristic analysis of the noise sources, that have been successfully verified by the CCN-NPTRNG-ENTROPY/HeuristicAnalysis evaluation test.

Note 54 [Mixing Entropy]

The entropy pool of the NPTRNG may either coincide with the internal state of the DRNG post-processing component or consist of a separate component.

- If the entropy pool coincides with the internal state of the DRNG, the NPTRNG behaves as a DRNG that is periodically reseeded (in a way that ensures that the required entropy threshold is maintained) with raw random numbers. In this case, the intermediate random numbers coincide with the raw random numbers, and the mixing function shall be functionally equivalent to the seed and reseed procedures.
- If the entropy pool is a separate component, it shall input the intermediate random numbers into the DRNG component through reseeding. In this case, the mixing function used to incorporate the raw random numbers into the entropy pool shall be cryptographic.

Note 55 [NTG.1.4]

At startup, the NPTRNG shall not generate any internal random numbers until the following condition has been met: the entropy pool has collected at least 240 bits of min-entropy from at least two different noise sources each. Both noise sources shall rely on different principles to provide randomness.

Viewed as a DRNG, the NPTRNG shall be seeded using 240 bits of min-entropy from each of the two noise sources.

If the entropy pool is separate from the DRNG's internal state, the DRNG shall be (re)seeded after this condition has been fulfilled before any internal random number is output.

Note₁: NPTRNGs often derive randomness from the timing of events. Different types of events (such as user interactions, incoming network packets, or system-level events) can be considered distinct non-physical noise sources, even if they share the same sampling mechanism (e.g., interrupt timestamps).

Note₂: Any entropy estimation for the raw random numbers shall be derived from structured heuristic analysis of the noise sources, that have been successfully verified by the **CCN-NPTRNG-ENTROPY/HeuristicAnalysis** evaluation test.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-DESIGN/EntropyPool	Inputs
Verify (using I_TDS_NP and I_IMP_NP) that: - The TOE implements an entropy pool and an entropy counter that tracks the estimated amount of entropy currently stored in the entropy pool. - Verify that the evidence complies with the specific notes listed in Table 23.	I_TDS_NP I_IMP_NP

4.4.1.1.3 DRNG Component

- The **TOE shall** [NPTRNG.03]:
 - Employ a DRNG that is compliant to the CCN-DRNG functionality class acting as its post-processing component, which takes the intermediate random numbers as input and outputs the internal random numbers.
 - Comply with the specific note listed in the table below:

DRNG Component	
CCN Class	Notes
CCN-NPTRNG	Note 56

Table 24. DRNG Component – Notes

Note 56 [Fresh Entropy]
The TOE shall input its intermediate random numbers into the DRNG post-processing component by its reseeding procedure or as additional input, and it shall be guaranteed that it never outputs more internal random number bits than indicated by the current value of the entropy counter. If the reseeding procedure is used to incorporate the intermediate random numbers, the state transition function of the DRNG shall be applied after the last internal random number has been generated. When the PTRNG starts, the internal state of the DRNG (used as the cryptographic post-processing component) has not yet received any entropy from the intermediate random numbers. Thus, only during start-up may a seeding procedure be used instead of a reseeding procedure. Note: The requirement [NPTRNG.03] shall be justified in the context of the seeding and reseeding procedures and/or the additional input mechanism of the DRNG component used by the TOE.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-DESIGN/DRNG	Inputs
Perform an evaluation in accordance with the present methodology to confirm that the DRNG post-processing component implemented by the TOE belongs to the CCN-DRNG functionality class. Additionally, verify, in the context of DRNG evaluation, that the TOE inputs its intermediate random numbers into the DRNG post-processing component by its reseeding procedure or as additional input and that the TOE complies with the specific note listed in Table 24.	I_VQ_DRNG I_90A I_IMP_DRNG

4.4.1.2 ENTROPY ESTIMATION

259. This topic defines the evaluation tests required to fulfill the **CCN-NPTRNG-ENTROPY** evaluation task.

4.4.1.2.1 Heuristic Analysis

- The **TOE shall** [NPTRNG.04] rely on non-physical noise sources whose behavior shall be described by heuristic analysis that supports a conservative min-entropy estimates per raw random number bit, $h_{rrn}^{heuristic}$, and which follows the specific notes listed in the table below:

Heuristic Analysis	
CCN Class	Notes
CCN-NPTRNG	Note 57 Note 58 Note 59

Table 25. Heuristic Analysis – Notes

Note 57 [Unpredictability Justification]
Heuristic analysis of the noise sources shall be substantiated with system-level or architectural reasoning that explains why the output is unpredictable. This justification shall: - Define the operational conditions and configuration parameters under which the noise source is expected to function as intended. - Clearly identify the origin of randomness exploited by the non-physical noise source.

Note 57 [Unpredictability Justification]

- Demonstrate that, if the operational conditions are met, an external attacker shall not be able to significantly degrade the entropy of the NPTRNG.
- Consider realistic attack vectors, including the attacker's ability to observe or influence the noise source.

Note 58 [Heuristic Estimate]

The heuristic analysis of the noise sources shall provide a conservative estimate of the min-entropy per raw random number bit, $h_{rrn}^{heuristic}$. It shall be supported by technical arguments and empirical evidence demonstrating that the estimated value does not overestimate the true entropy of the noise source.

Additionally, the heuristic analysis and its associated entropy estimate shall serve as a basis for the design of the noise source, guiding the selection and tuning of entropy-relevant parameters.

Note 59 [Rigor Requirement]

In contrast to the stochastic modelling physical noise sources, the literature lacks formal foundations and broadly accepted examples for the heuristic analysis of non-physical noise sources.

However, the heuristic analysis described in [JEPROC] for the CPU Jitter Entropy Library can be considered a benchmark for maturity, structure, and conservativeness in heuristic analysis.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-ENTROPY/HeuristicAnalysis	Inputs
Verify, using I_ENT_NP, that: - A heuristic analysis for each noise source credited with entropy in the TOE is provided, from which a conservative value for $h_{rrn}^{heuristic}$ is derived. - The heuristic analysis complies with the specific notes listed in Table 25. In order to do so, the tester shall review the evidence provided in I_ENT_NP, verify the validity of the methods used to support the	I_ENT_NP

CCN-NPTRNG-ENTROPY/HeuristicAnalysis	Inputs
heuristic analysis, and reproduce the results by applying those methods in order to validate the estimated entropy.	

4.4.1.2.2 Internal Random Numbers Entropy

- The **TOE shall** [NPTRNG.05]:
 - Generate internal numbers for which their min-entropy per bit, h_{irn} , shall belong to the interval $[0.98, 1-2^{-32}]$.
 - Comply with the following specific note listed in the table below:

Entropy Estimation	
CCN Class	Notes
CCN-NPTRNG	Note 60

Table 26. Entropy Estimation – Notes

Note 60 [Entropy Content]
An explicit estimation for h_{irn} shall be provided. In order to do so: - The entropy provided by each noise source shall be considered. For this, the estimated values of $h_{rrn}^{heuristic}$ for each noise source shall be employed. - It shall be shown that the DRNG performs effective compression up to the claimed level. For this, the DRNG shall be modeled as a composition of random mappings, and the results described in Section 4.4.2 of [FCRNG] for random mappings shall be employed to support the entropy claim h_{irn}.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-ENTROPY/InternalEntropy	Inputs
Verify, using I_ENT_NP, that: - The TOE generates internal random numbers for which the min-entropy per bit is greater than or equal to 0.98. - The TOE complies with the specific note listed in Table 26. In order to do so, the tester shall review the evidence provided in I_ENT, verify the validity of the methods used to support the justification, and reproduce the results by applying those methods.	I_ENT_NP

4.4.1.3 HEALTH TESTING REVIEW

260. This topic defines the evaluation tests required to fulfill the **CCN-NPTRNG-HEALTH** evaluation task.

4.4.1.3.1 Continuous Tests

- The **TOE shall** [NPTRNG.06] implement continuous tests that check the quality of the raw random numbers while the TOE is in operation, aiming to detect entropy degradation and catastrophic failures in its underlying noise sources, and comply with the following specific notes:

Continuous Test	
CCN Class	Notes
CCN-NPTRNG	Note 61
	Note 62
	Note 63

Table 27. Continuous Test – Notes

Note 61 [Continuous Tests Specifications]

The continuous tests shall include:

- A test aiming to detect whether some value is consecutively repeated many more times than expected. This shall consist of one of the following:
 - The Repetition Count Test defined in Section 4.4.1 from [SP 800-90B]. It is designed to quickly detect catastrophic failures that cause the noise source to become "stuck" on a single output value for a long period of time.
 - A developer-defined test that meets the requirements for a substitution of the Repetition Count Test, as described in Section 4.5 from [SP 800-90B].
- A test aiming to detect whether some value becomes much more common in the sequence of noise source outputs than expected, given the assessed entropy per sample of the source. This shall consist of one of the following:
 - The Adaptive Proportion Test defined in Section 4.4.2 from [SP 800-90B]. It is designed to detect a large loss of entropy.
 - A developer-defined test that meets the requirements for a substitution of the Adaptive Proportion Test, as described in Section 4.5 from [SP 800-90B].

Note 61 [Continuous Tests Specifications]

In addition, the selected continuous tests shall comply with the requirements specified in Section 4.3 from [SP 800-90B].

Note 62 [Failure Modes]

The design of the implemented continuous tests shall be justified by a detailed study of all possible noise source failure modes. This study shall identify technically plausible failure scenarios, describe their impact on the raw random numbers and explain how each scenario is reliably detected by the tests.

This study shall be based on (or be part of) the heuristic analysis of the noise sources evaluated by the **CCN-NPTRNG-ENTROPY/HeuristicAnalysis** evaluation test.

Note 63 [Performance Justification - NP]

Continuous tests shall be effective in the sense that they reliably detect any non-tolerable degradation of entropy or a catastrophic failure and do so early enough to prevent the generation of excessive weak internal random numbers.

For each implemented continuous test:

- Reasonable false alarm and silent failures probabilities shall be selected, denoted by α and β , respectively, based on the details of the TOE and its consuming application. In any case, the false alarm probability shall be between 2^{-20} and 2^{-40}
- Cutoff values shall be derived from α and the min-entropy per raw random number bit estimate, $h_{rrn}^{heuristic}$.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-HEALTH/ContinuousTest	Inputs
Review I_HEALTH_NP and verify that: - The TOE implements continuous tests that check the quality of the raw random numbers while the TOE is in operation aiming to detect entropy degradation and catastrophic failures in its underlying noise sources.	I_HEALTH_NP

CCN-NPTRNG-HEALTH/ContinuousTest	Inputs
- The TOE complies with the specific notes listed in Table 27. If deemed necessary, the tester may perform additional independent checks to confirm the claimed detection performance.	

4.4.1.3.2 Start-Up Test

- The **TOE shall** [NPTRNG.07] perform a Start-Up Test on all noise sources that contribute with entropy to the entropy pool upon initialization of the TOE, and shall comply with the following specific notes:

Start-Up Test	
CCN Class	Notes
CCN-NPTRNG	Note 61 Note 62 Note 63 Note 64 Note 65 Note 66

Table 28. Start-Up Test – Notes

Note 64 [Detection Objective]

The Start-Up Test shall provide some assurance that the entropy source components are working as expected before they are used during normal operating conditions, and that no component has failed since the last time that the start-up tests were run. In particular, the TOE shall not output any internal random numbers before the Start-Up Test has been passed. The samples subjected to startup testing may be released.

The entropy source's startup tests shall run the continuous health tests over at least 1024 consecutive samples.

Note 65 [Start-Up Test Specifications]

A reasonable approach is to apply the continuous tests with (possible) adjusted evaluation rules suitable for start-up conditions. In this case, those adjustments shall be justified in line with the nature of the tests.

Note 65 [Start-Up Test Specifications]

If alternative test methods are used, they shall meet equivalent requirements for failure detection and statistical weakness coverage as the continuous tests.

Additionally, Note 61, Note 62 and Note 63 shall be fulfilled.

Note 66 [Known Answer Test]

The Start-Up Test shall include a Known Answer Test (KAT) for each deterministic component of the NPTRNG.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-HEALTH/StartUp	Inputs
Verify, using I_HEALTH_NP, that the TOE does not output any internal random number until the Start-Up Test has passed with success. Verify that the evidence presented in I_HEALTH_NP demonstrates the compliance of the Start-Up Test with the specific notes listed in Table 28. If deemed necessary, the tester may perform additional independent checks to confirm the claimed detection performance.	I_HEALTH_NP

4.4.1.3.3 Health Testing Alarms

- The **TOE shall** [NPTRNG.08] implement alarm logic associated with the failure conditions detected by the Health Tests, and shall comply with the specific note in the table below:

Health Testing Alarms	
CCN Class	Notes
CCN-NPTRNG	Note 67

Table 29. Health Testing Alarms – Notes

Note 67 [Health Testing Alarms]

When the health tests fail, the entropy source shall notify the consuming application of the error condition.

Note 67 [Health Testing Alarms]

The TOE shall only permit the output of internal random numbers once a signal result of the Start-Up Test confirms that the test has been successfully passed.

An alarm signal resulting from the continuous tests shall be raised to prevent the output of internal random numbers when:

- There is a significant decrease in the entropy of the generated random numbers. It shall be ensured that no internal random numbers with insufficient entropy are output.
- Noise source failures occur. Any internal random numbers that depend on any raw random number that has been generated after the total failure of the physical noise source shall be output.

In any case, it shall be proven that the alarm logic for the Health Tests is correctly implemented and ensures the required blocking of internal random numbers as specified.

- The **tester shall** perform the following evaluation test:

CCN-NPTRNG-HEALTH/Alarms	Inputs
Verify, using I_HEALTH_NP that alarm logic is implemented for the failure conditions detected by the Start-Up Test and continuous tests as specified. Verify, using I_HEALTH_NP, the requirements expressed in the note listed in Table 29.	I_HEALTH_NP

4.4.1.4 BLACK-BOX STATISTICAL TESTING

261. This topic defines the evaluation tests required to fulfill the **CCN-NPTRNG-BLACKBOX** evaluation task.

262. These tests apply the standard black-box statistical test suites defined in [SP 800-90B]. It shall be noted that the results obtained in this task serve merely as a sanity check, since black-box statistical tests alone can only be considered a necessary, but not sufficient, condition for the satisfactory evaluation of an entropy source.

4.4.1.4.1 SP 800-90B Test Suite

- The **TOE shall** [NPTRNG.09] generate raw random numbers and internal random numbers that pass the entropy estimation and statistical testing procedures described in [SP 800-90B] and follow the specific notes below:

SP 800-90B Test Suite	
CCN Class	Notes
CCN-NPTRNG	Note 68 Note 69

Table 30. SP 800-90B Test Suite– Notes

Note 68 [Block Assessment]

In order to obtain a black box min-entropy estimate per raw random number bit, denoted as h_{rrn}^{block} , the vendor shall perform the following steps on the output of all such noise sources:

- All the min-entropy estimators described in section 6.2 of the [SP 800-90B] document shall be applied to the collected raw random numbers. The lowest estimator value shall be credited as h_{rrn}^{test} .
- The above-mentioned shall be repeated for the different datasets generated under all admissible range of operational conditions.
- Each dataset shall contain at least 1,000,000 consecutive samples of raw random numbers collected during normal operation.

The same process shall be performed over the internal random numbers, in order to produce a black box min-entropy estimate per internal random number bit, denoted as h_{irn}^{block} , according to the requirements presented in [SP 800-90B].

All the results, datasets and tools shall be presented in **I_TEST_NP**.

Note 69 [Restart Assessment]

The restart tests defined in Section 3.1.4 of the [SP 800-90B] document shall be applied to the raw random numbers generated by all noise sources that contribute entropy to the entropy pool.

The restart process employed during testing shall reflect the expected behavior of the device in real-world use, including any initialization delays, startup procedures, or waiting periods declared in the operational conditions' documentation.

All the results, datasets and tools shall be presented in **I_TEST_NP**.

- The **tester shall** perform the following evaluation test:

CCN-PTRNG-BLACKBOX/90B	Inputs
Verify, using I_TEST_NP, that the [SP 800-90B] entropy estimation procedures were applied properly by the vendor, following the specific notes from Table 30. Additionally, validate the results by repeating all tests using the data provided in I_TEST_NP and the official [SP 800-90B] testing tool. Compare (using I_ENT_NP and I_TEST_NP) the min-entropy per bit estimation values produced by NIST estimators with the values of $h_{rrn}^{heuristic}$ and h_{irn} as a sanity check.	I_TEST_NP I_ENT_NP

5 DETERMINISTIC RANDOM NUMBER GENERATORS

5.1 OBJECTIVE

263. This section defines the security requirements, evaluation inputs, terminology, and evaluation tasks applicable to the CCN-DRNG functionality class.
264. The requirements defined in this section are based on those of the DRG.3 functionality class from AIS-20, as specified in the mathematical and technical reference [FCRNG], and have been adapted following CCN's selection criteria and evaluation procedures. The evaluation tasks have been designed so that any CCN-DRNG successfully evaluated under this methodology can be considered DRG.3 compliant.
265. This methodology explicitly accounts for the security guarantees already established for constructions such as Hash_DRBG and HMAC_DRBG. As a result, CCN-DRNG implementations based on these mechanisms may be subject to a more streamlined evaluation process.

5.2 CCN-DRNG DEFINITION

266. Deterministic Random Number Generators (DRNGs) is a type of RNG that produces pseudo-random numbers using deterministic algorithms.
267. DRNGs are cryptographic components that generate pseudo-random bit sequences through fully deterministic algorithms. Their operation is entirely defined by an initial internal state—typically derived from seed material obtained during instantiation—and by a combination of cryptographically secure state transition and output functions.
268. A DRNG is claimed to belong to the CCN-DRNG functionality class if has been successfully evaluated under the present methodology.

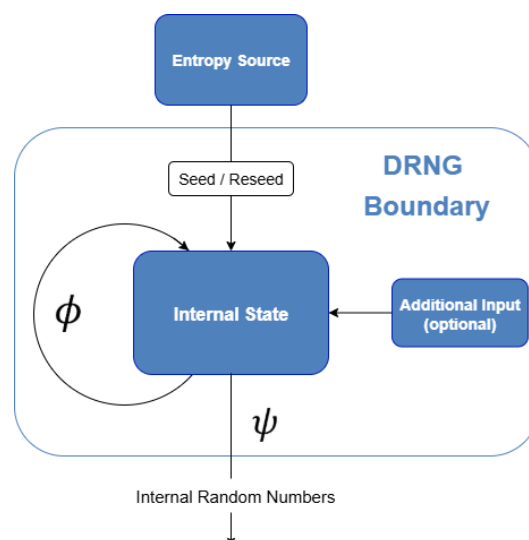


Figure 4. CCN-DRNG Design Scheme

269. The general structure of the CCN-DRNG is shown in Figure 4 and its operation is based on the following procedures:

DRNG Instantiation

270. First, the DRNG is instantiated. The instantiation process of a DRNG begins by acquiring seed material by means of the seeding procedure. The seed material may be combined with a personalization string, to generate a seed. This seed is then used to create the initial internal state of the DRNG. In the case of a CCN-DRNG, the seed material must originate from a valid entropy source, which shall be evaluated by the present methodology.

Internal State Evolution

271. Once instantiated, the state transition function, denoted as ϕ , updates the internal state during the life cycle of the DRNG instance.

272. For the CCN-DRNG functionality class, both the reseeding procedure and the insertion of the additional input are optional. In DRNGs that do not support either feature, identical seed material will lead to identical internal random numbers, due to the deterministic nature of the generation process.

Output Generation

273. Upon receiving an external request from the consuming application, the output function, denoted as ψ , generates internal random numbers from the current internal state in response to the request. These internal random numbers represent the cryptographic output of the DRNG, which supports a concrete security strength.

5.3 INPUTS

274. In order to simplify the revision of the documentation and reduce the evaluation effort, this section defines the set of Inputs that the **vendor shall** provide for the evaluation of a Deterministic Random Number Generator (DRNG) according to the present methodology.

275. For the evaluation of the CCN-DRNG functionality class, apart from the actual implementation evidence (**I_IMP_DRNG**), two different main inputs shall be provided depending on the DRNG scheme implemented by the TOE:

- If the DRNG is based on the standardized Hash_DRBG or HMAC_DRBG constructions from [SP 800-90A], the **vendor shall** provide **I_90A**.
- In any other case, the **vendor shall** provide **I_VQ_DRNG**.

276. Given the challenges associated with evaluating DRNGs, the tester may request additional evidence. In such cases, the **vendor shall** supply these supplementary materials.

I_VQ_DRNG: DRNG Vendor Questionnaire

277. The **vendor shall** provide this input in case the DRNG is based on a generic scheme not compliant with the standardized Hash_DRBG or HMAC_DRBG constructions from [SP 800-90A]
278. This input consists of a Vendor Questionnaire (VQ), which contains the vendor's answers to a structured set of questions provided by the laboratory. These questions are designed to extract all relevant information required to evaluate generic DRNG designs, with a focus on the formal justification of the AIS-20 security properties from [FCRNG].
279. All in all, this input contains all the evidence related to the DRNG's algorithmic design, formal model, entropy acquisition, seeding and reseeding procedures, and security properties.

I_90A: SP 800-90A Vendor Questionnaire

280. The **vendor shall** provide this input in case the DRNG is based on the standardized Hash_DRBG or HMAC_DRBG constructions from [SP 800-90A]
281. This input also consists of a Vendor Questionnaire (VQ), based on a structured template provided by the laboratory. It is specifically intended for DRNGs based on the standardized Hash_DRBG or HMAC_DRBG schemes defined in [SP 800-90A]. The **vendor shall** provide detailed answers and supporting evidence demonstrating full compliance with the applicable [SP 800-90A] requirements, including the seeding and reseeding procedures, entropy input handling, internal state management, and parameter selection.

Note: As a formal proof exists in [FCRNG] for DRG.3 algorithmic compliance of Hash_DRBG and HMAC_DRBG and the vast majority of the [SP 800-90A] requirements can be evaluated via cryptographic conformance testing (e.g., MEMeC evaluation), the required effort to provide all the evidence in this input is significantly lower than in the case of the **I_VQ_DRNG** input.

I_IMP_DRNG: DRNG Implementation Evidence

282. Within this input, **the vendor** shall include the complete implementation artifacts of the TOE, such as source code, configuration files, and any supporting libraries or build scripts. The materials provided must represent the exact version of the DRNG deployed in the TOE, allowing for a direct inspection of its algorithmic structure, state management, entropy injection mechanisms, and cryptographic primitives.
283. In particular, this input shall include the concrete evidence necessary to fulfill each requirement established in [DRNG.06], [DRNG.11], [DRNG.12], [DRNG.13] and their associated specific notes.

5.4 CCN-DRNG EVALUATION TASKS

284. This section defines the specific requirements for Deterministic Random Number Generators, as established by CCN, which must be successfully evaluated for the TOE generator to be recognized as a CCN-DRNG approved RNG.

285. The following list of evaluation tasks related to the CCN-DRNG requirements have been defined:

CCN-DRNG EVALUATION TASK	CCN-DRNG-IMPLEMENTATION
The tester shall verify, using I_VQ_DRNG (or I_90A), that the TOE has been successfully evaluated against the [CCN-MEMeC] methodology at level CL2 or CL3. This task ensures that the TOE is correctly implemented, corresponds to its declared design, and complies with the applicable CCN cryptographic requirements.	

CCN-DRNG EVALUATION TASK	CCN-DRNG-ENTROPY
The tester shall review I_VQ_DRNG (or I_90A), to verify that the TOE obtains its seed material from a valid entropy source. The tester shall also verify that the seeding and, if implemented, the reseeding procedures are sufficient to ensure that the resulting internal state contains at least 240 bits of min-entropy.	

CCN-DRNG EVALUATION TASK	CCN-DRNG-ALGORITHMIC
The tester shall: - If the TOE implements Hash_DRBG or HMAC_DRBG, review I_90A to verify that the implementation complies with the applicable requirements in [SP 800-90A]; if compliance is confirmed, the previous requirement is considered fulfilled. - Otherwise, review the formal model of the TOE provided in I_VQ_DRNG and (using I_IMP_DRNG) that it is aligned to the TOE implementation. In addition, verify that it is properly employed to justify the algorithmic security properties required for CCN-DRNG. In all cases, the tester shall verify that appropriate evidence supports the claimed security strength.	

286. For the correct completion of the evaluation tasks, each one has been divided into a set of evaluation tests. The **tester shall** perform the associated mandatory

tests, and also the implementation-dependent tests, if applicable, according to the TOE design.

CCN-DRNG Evaluation Task	Evaluation Test	Test Categorization
CCN-DRNG-IMPLEMENTATION DRNG Implementation	CCN-DRNG-IMPLEMENTATION/MEMeC	Mandatory
CCN-DRNG-ENTROPY Entropy Acquisition Evaluation	CCN-DRNG-ENTROPY/Source	Mandatory
	CCN-DRNG-ENTROPY/Seeding	Mandatory
	CCN-DRNG-ENTROPY/Reseeding	Mandatory
CCN-DRNG-ALGORITHMIC Algorithmic Security Properties	CCN-DRNG-ALGORITHMIC/90A	Impl-Dep
	CCN-DRNG-ALGORITHMIC/FormalModel	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.2	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.3	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.5	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.6	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.7	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.8	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.9	Impl-Dep
	CCN-DRNG-ALGORITHMIC/DRG.3.10	Impl-Dep

CCN-DRNG Evaluation Task	Evaluation Test	Test Categorization
	CCN-DRNG-ALGORITHMIC/SecurityStrength	Mandatory

Table 31. Evaluation Tests defined for CCN-DRNG Requirements

5.4.1 EVALUATION TESTS DEFINITION

287. This subsection defines the specific evaluation tests for Deterministic Random Number Generators, as established by CCN, which must be performed by the tester to complete the evaluation tasks in accordance with the specific CCN-DRNG requirements.

5.4.1.1 DRNG IMPLEMENTATION

288. This topic defines the evaluation tests required to fulfill the **CCN-DRNG-IMPLEMENTATION** evaluation task.

289. A Deterministic Random Number Generator (DRNG) is, by design, an algorithmic construction that derives its security properties from the those of the cryptographic mechanisms it employs. As such, both its design and implementation must be understood and evaluated as cryptographic mechanisms.

290. For this reason, the design and implementation of TOEs belonging to the CCN-DRNG functionality class shall be evaluated following the [CCN-MEMeC] methodology. Its use provides assurance that the TOE implementation corresponds to the declared design, that the cryptographic mechanisms employed and their parametrization are recommended according to [CCN-STIC-221] and that the associated CCN cryptographic requirements are enforced. This implies the secure management of the DRNG's internal state and the correct implementation of self-tests.

5.4.1.1.1 Cryptographic Evaluation

- The **TOE shall** [DRNG.01] be evaluated in accordance with the Cryptographic Mechanisms Evaluation Methodology ([CCN-MEMeC]) at evaluation level **CL2** or **CL3** and, in the context of this evaluation, comply with the specific note listed in the table below:

Cryptographic Evaluation	
CCN Class	Notes
CCN-DRNG	Note 70

Table 32. Cryptographic Evaluation – Notes

Note 70 [Self-Tests]

As part of the **CCN-SELFTEST** evaluation task defined in the [CCN-MEMeC] methodology, the TOE shall execute the implemented self-tests prior to the output of any internal random number.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-IMPLEMENTATION/MEMeC	Inputs
Verify in, I_VQ_DRNG (or I_90A) that the TOE has been successfully evaluated against the [CCN-MEMeC] methodology at level CL2 or CL3 , considering the specific note listed in Table 32.	I_VQ_DRNG I_90A

5.4.1.2 ENTROPY ACQUISITION EVALUATION

291. This topic defines the evaluation tests required to fulfill the **CCN-DRNG-ENTROPY** evaluation task.

5.4.1.2.1 Entropy Source Verification

- The **TOE shall** [DRNG.02] obtain the seed material (for the seeding procedure and reseeding procedure) from:
 - a PTRNG of class CCN-PTRNG,
 - a PTRNG of class CCN-HPTRNG,
 - an NPTRNG of class CCN-NPTRNG,
 - a DRNG of class CCN-EDRNG.
- The **tester shall** perform the following evaluation test:

CCN-DRNG-ENTROPY/Source	Inputs
Verify, using I_VQ_DRNG (or I_90A), that the seed material used by the TOE for seeding and reseeding is obtained from a random number generator belonging to one of the allowed random number generator classes listed above, acting as source of randomness. Note: Of course, by definition, verify that the random number generator, employed for the seeding and (optional) reseeding of the TOE, has been successfully evaluated according to the present methodology.	I_VQ_DRNG I_90A

5.4.1.2.2 Seeding Procedure

- The **TOE shall** [DRNG.03]:
 - Perform its seeding procedure employing seed material generated by a RNG which complies to [DRNG.02] and ensure that the resulting effective internal state contains at least 240 bits of min-entropy after seeding.
 - Comply with the following specific notes listed in the table below:

Seeding Procedure	
CCN Class	Notes
CCN-DRNG	Note 71 Note 72

Table 33. Seeding Procedure – Notes

Note 71 [DRG.3.4]

If the seed material is generated by a CCN-EDRNG compliant DRNG, then the following conditions shall be met:

- The DRNG shall have been provided with enhanced forward secrecy prior to generating the seed material.
- The seed material shall comprise at least 256 bits.
- It shall be justified that if the seed material contains at least 250 bits of min-entropy, then the effective internal state resulting from the seeding procedure contains at least 240 bits of min-entropy.

Note 72 [Seeding Evidence]

The seeding procedure shall be supported by evidence covering:

- The algorithmic description of the procedure.
- The interfaces (logical or physical) used to receive and inject the seed material into the TOE.
- The conditions under which the procedure is triggered (e.g., at startup, on demand).
- The implemented protections or error-handling mechanisms for invalid or insufficient seed material.
- The impact of the seeding procedure on the entropy of the effective internal state.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ENTROPY/Seeding	Inputs
Review I_VQ_DRNG (or I_90A) to verify that: - The resulting effective internal state contains at least 240 bits of min-entropy after seeding. - The TOE complies with the specific notes listed in Table 33.	I_VQ_DRNG I_90A

5.4.1.2.3 Reseeding Procedure

292. In the case the TOE employs a reseeding procedure:

- The **TOE shall** [DRNG.04]:
 - Perform its reseeding procedure employing seed material generated by a RNG which complies to [DRNG.02] and shall ensure that the resulting effective internal state contains at least 240 bits of min-entropy after reseeding.
 - Comply with the following specific notes listed in the table below:

Reseeding Procedure	
CCN Class	Notes
CCN-DRNG	Note 71 Note 73

Table 34. Reseeding Procedure – Notes

Note 73 [Reseeding Evidence]
The reseeding procedure shall be supported by evidence covering: - The algorithmic description of the procedure. - The interfaces (logical or physical) used to receive and inject the seed material into the TOE. - The reseeding policy, including the conditions or thresholds that trigger the procedure, and the maximum seedlife permitted. - The implemented protections or error-handling mechanisms for invalid or insufficient seed material. - The impact of the reseeding procedure on the entropy of the effective internal state.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ENTROPY/Reseeding	Inputs
Review I_VQ_DRNG (or I_90A) to verify that: - The resulting effective internal state contains at least 240 bits of min-entropy after seeding. - The TOE complies with the specific notes listed in Table 34.	I_VQ_DRNG I_90A

5.4.1.3 ALGORITHMIC SECURITY PROPERTIES

293. This topic defines the cryptographic evaluation tests required to fulfill the **CCN-DRNG-ALGORITHMIC** evaluation task.
294. Any DRNG belonging to the CCN-DRNG functionality class must fulfill a set of algorithmic security requirements related to forward secrecy, backward secrecy and enhanced backward secrecy in the presence of different adversarial models. These properties, which aim to lessen the impact of a compromise of the internal state, shall be properly demonstrated relying on a formal mathematical description of the DRNG's algorithmic structure and its seeding and reseeding procedures. The technical reference [FCRNG] provides a framework for this endeavor.
295. In practice, evaluating these algorithmic properties from a formal mathematical model can be challenging. However, for some DRNGs based on certain standardized schemes, some of this work has already been done and published. Specifically, [FCRNG], in section 5.3, already provides formal conformity proofs for the Hash_DRBG and HMAC_DRBG constructions defined in [SP 800-90A], which demonstrate that they meet the algorithmic requirements presented in this evaluation task.
296. As a result, when evaluating a DRNG based on Hash_DRBG or HMAC_DRBG, the mentioned conformity proof streamlines its evaluation. However, this only applies if the implementation is shown to fully comply with all requirements of [SP 800-90A]. If the DRNG scheme does not meet these conditions, the full set of algorithmic evaluation tests within this task shall be performed.
297. A paradigmatic example is CTR_DRBG, for which no pre-approved formal mathematical conformity proof exists. Moreover, its variants based on AES-128 or AES-192 are known not to meet the algorithmic requirements defined in this evaluation task.

5.4.1.3.1 Hash-Based DRBGs

298. In order to verify that the implementation of a DRNG based on the Hash_DRBG or HMAC_DRBG schemes complies with the specification defined in [SP 800-90A], the [CCN-MEMeC] evaluation required by the **CCN-DRNG-IMPLEMENTATION** task ensures compliance with many of the relevant

requirements, but not all. Therefore, the present evaluation test has been defined to assess those specific requirements of [SP 800-90A] that are not covered by the cryptographic evaluation tasks of the [CCN-MEMeC] methodology.

299. Thus, in case the TOE is based on the Hash_DRBG or HMAC_DRBG from [SP 800-90A]:

- The **TOE shall** [DRNG.05] comply with all applicable requirements defined in [SP 800-90A], based on the DRBG scheme it implements (e.g., Hash_DRBG or HMAC_DRBG).
- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/90A	Inputs
Using I_90A, verify that all listed requirements are satisfied. This shall be done by reviewing that the evidence provided by the vendor in I_90A adequately supports compliance with each of the referenced [SP 800-90A] requirements. Note: The successful verification of this test and of CCN-DRNG-IMPLEMENTATION/MEMeC implies the successful verification of the CCN-DRNG-ALGORITHMIC/FormalModel test and the nine CCN-DRNG-ALGORITHMIC/DRG.3.X tests.	I_90A

5.4.1.3.2 DRNG Formal Model

300. The **TOE shall** [DRNG.06] be defined by a Formal Model for its algorithmic structure, and a Formal Model for the seeding and (optional) reseeding procedures which comply with the specific notes listed in the table below:

DRNG Formal Model	
CCN Class	Notes
CCN-DRNG	Note 74 Note 75 Note 76 Note 77

Table 35. DRNG Formal Model – Notes

Note 74 [Algorithmic Structure Formal Model]
The general DRNG algorithmic structure shall be defined by a formal model comprising a 9-tuple $(S, S_{req}, R, A, I, \phi, \phi_{req}, \phi_0, \psi)$ which follows the structure and interpretation as presented in paragraph 129 and the subsequent paragraphs of [FCRNG].

Note 75 [Seeding and Reseeding Formal Model]

The seeding procedure shall be defined by a formal model which comprises a 4-tuple, (SM, PS, S, ϕ_{seed}) which follows the structure and interpretation as presented in paragraph 143 and the subsequent paragraphs of [FCRNG].

If employed, the reseeding procedure shall be defined by a formal model which comprises a 4-tuple $(SM, PS, S, \phi_{reseed})$ which follows the structure and interpretation as presented in paragraph 147 and the subsequent paragraphs of [FCRNG].

Note 76 [Parameters]

If public parameters and/or secret parameters affect any of the mappings ϕ, ϕ_{req}, ϕ_0 , then these data items shall be part of S .

Note 77 [Link to Implementation]

Evidence shall be provided to demonstrate that the formal modelling of the DRNG is aligned with its concrete implementation. A mapping shall be provided between the elements of the different tuples in the model and the corresponding functions, algorithms, processes and variables in the actual implementation.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/FormalModel	Inputs
Review I_VQ_DRNG to verify that: - It provides a formal model of the TOE that complies with the specific notes listed in Table 35. - The formal model has been employed as the theoretical ground for the justification of the security requirements addressed across the rest of evaluation tests within the CCN-DRNG-ALGORITHMIC evaluation task. Additionally, perform an implementation code review (using I_IMP_DRNG) to verify that the TOE complies with Note 77.	I_VQ_DRNG I_IMP_DRNG

5.4.1.3.3 Request Length

- The **TOE shall** [DRNG.07] accept requests that comprise a maximum length of 2^{19} bits, and comply with the specific notes listed in the table below:

Request Length	
CCN Class	Notes
CCN-DRNG	Note 78 Note 79

Table 36. Request Length – Notes

Note 78 [Request Counter]

The TOE shall implement a counter for the number of internal random numbers necessary per request.

Note 79 [Zeroize Non-Output Bits]

Any internal random number bits not included in the response to the request shall not be stored and must be securely destroyed.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.2	Inputs
Review I_VQ_DRNG to verify that: - The TOE accepts requests that comprise a maximum length of 2^{19} bits. - The TOE complies with the specific notes listed in Table 36.	I_VQ_DRNG

5.4.1.3.4 Effective Internal State

301. The effective internal state refers to the portion of the DRNG's internal state that determines future outputs and contains usable entropy. Its size is directly related to the DRNG's security strength.

- The **TOE shall** [DRNG.08] have an effective internal state that comprises at least 246 bits, and comply with the specific note listed in the table below:

Effective Internal State	
CCN Class	Notes
CCN-DRNG	Note 80

Table 37. Effective Internal State – Notes

Note 80 [Attacker Assumption.1]

The effective internal state and its size shall be determined under the assumption that the attacker knows a large number of internal random numbers. For the reseeding procedure, the least favorable scenario shall be assumed where an attacker knows the previous internal state.

In particular, the following (conceivable) reasoning will not be accepted for verifying this requirement: since immediately after the seeding procedure/reseeding procedure an attacker did not have a chance to collect any information about the DRNG, any part of the internal state (apart from publicly known input) is unknown and thus effective.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.3	Inputs
Review I_VQ_DRNG to verify that: - The TOE has an effective internal state that comprises at least 246 bits. - The effective internal state of the TOE complies with the specific note listed in Table 37.	I_VQ_DRNG

5.4.1.3.5 Forward Secrecy

- The **TOE shall** [DRNG.09] provide forward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:

Forward Secrecy	
CCN Class	Notes
CCN-DRNG	Note 81 Note 82 Note 83

Table 38. Forward Secrecy – Notes

Note 81 [Algorithmic Properties.1]

Forward secrecy and backward secrecy requirements shall be guaranteed by the algorithmic properties of the DRNG, i.e., by the interaction of the state transition function and the output function. A lack of algorithmic properties cannot be compensated for by other measures, e.g., by high-entropy additional input.

Note 82 [Attacker Assumption.2]

When providing the forward secrecy and the backward secrecy properties, it shall be assumed that an attacker knows a sequence of internal random numbers generated within one or several requests.

Note 83 [Security Strength]

In the context of forward secrecy, backward secrecy and enhanced backward secrecy for k-bit output strings the security strength (of cryptographic mechanisms) shall not be significantly lower than 240 bits.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.5	Inputs
Review I_VQ_DRNG to verify that: - The TOE provides forward secrecy on the granularity level of internal random numbers. - The TOE complies with the specific notes listed in Table 38.	I_VQ_DRNG

5.4.1.3.6 Backward Secrecy

- The **TOE shall** [DRNG.10] provide backward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:

Backward Secrecy	
CCN Class	Notes
CCN-DRNG	Note 81 Note 82 Note 83

Table 39. Backward Secrecy – Notes

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.6	Inputs
Review I_VQ_DRNG to verify that: - The TOE provides backward secrecy on the granularity level of internal random numbers. - The TOE complies with the specific notes listed in Table 39.	I_VQ_DRNG

5.4.1.3.7 Enhanced Backward Secrecy

- The **TOE shall** [DRNG.11] provide enhanced backward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:
 - Alternatively, provide enhanced backward secrecy on the granularity level of requests, provided that the requests meet the condition of conceptual atomicity.

Enhanced Backward Secrecy	
CCN Class	Notes
CCN-DRNG	Note 83 Note 84 Note 85 Note 86

Table 40. Enhanced Backward Secrecy – Notes

Note 84 [Algorithmic Properties.2]

Enhanced backward secrecy is an algorithmic property, as forward and backward secrecy properties, it shall be guaranteed by the algorithmic properties of the DRNG. It shall not be compensated for or supported by technical security measures that (are claimed to) prevent the internal state from being compromised or modified.

Note 85 [Attacker Assumption.3]

When providing enhanced backward secrecy property, it shall be assumed that an attacker knows the current internal state of the DRNG (after the request is performed) and (if relevant) the last additional input.

In contrast to forward secrecy and backward secrecy, the size of the effective internal state is not relevant for enhanced backward secrecy—assuming that

Note 85 [Attacker Assumption.3]

sufficient entropy has been injected—since the attacker is assumed to have full knowledge of the internal state in any case.

Note 86 [One-way function]

The enhanced backward secrecy property essentially requires a state transition function that is a one-way function.

This specific note is addressed in the **CCN-DRNG-ALGORITHMIC/DRG.3.9** evaluation test.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.7	Inputs
Review I_VQ_DRNG and perform a source code review (using I_IMP_DRNG) to verify that: - The TOE provides enhanced backward secrecy on the granularity level of internal random numbers. - The TOE complies with the specific notes listed in Table 40. Note: For the analysis of enhanced backward secrecy, the tester shall not only consider the formal model, but also the TOE implementation.	I_VQ_DRNG I_IMP_DRNG

5.4.1.3.8 Additional Input

302. A reasonable design strategy to prevent additional input from weakening the DRNG is to employ a state transition function whose core is based on a cryptographic hash function, using it to incorporate the additional input into the internal state. However, depending on the specific DRNG design, simpler state transition functions may also be suitable.

- If additional input is permitted, the **TOE shall** [DRNG.12] accept additional input that does not weaken the strength of the DRNG even if an attacker is able to control the additional input, and comply with the specific note listed in the table below:

Additional Input	
CCN Class	Notes
CCN-DRNG	Note 81

Table 41. Additional Input – Notes

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.8	Inputs
Review I_VQ_DRNG and perform a source code review (using I_IMP_DRNG) to verify that: - The TOE accepts additional input that does not weaken the strength of the DRNG even if an attacker is able to control the additional input. - The TOE complies with the specific note listed in Table 41.	I_VQ_DRNG I_IMP_DRNG

5.4.1.3.9 DRNG Functions

- The **TOE shall** [DRNG.13] implement a cryptographic and one-way state transition function, ϕ , and a cryptographic output function, ψ .
- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.9	Inputs
Review I_VQ_DRNG and perform an implementation code review (using I_IMP_DRNG) to verify that: - The transition function, ϕ, is one-way and cryptographic function, and - the output function, ψ, is cryptographic.	I_VQ_DRNG I_IMP_DRNG

5.4.1.3.10 Statistical Inconspicuousness

- The **TOE shall** [DRNG.14] output internal random numbers that have statistical inconspicuousness and comply with the specific note listed in the table below:

Statistical Inconspicuousness	
CCN Class	Notes
CCN-DRNG	Note 87

Table 42. Statistical Inconspicuousness – Notes

Note 87 [Justification]
The statistical inconspicuousness justification shall be supported by theoretical considerations supported by statistical tests, with justification of the choice.

Note 87 [Justification]

Usually, evidence that the DRNG fulfills this requirement can be given by theoretical arguments about the implemented cryptographic mechanisms, by leveraging widely recognized distributional assumptions.

- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.10	Inputs
Review I_VQ_DRNG to verify that: - The TOE outputs internal random numbers that have statistical inconspicuousness. - The TOE complies with the specific note listed in Table 42. If the tester suspects that the TOE design does not have the property of statistical inconspicuousness, targeted statistical tests should be applied to test for these possibilities.	I_VQ_DRNG

5.4.1.3.11 Security Strength

- The **TOE shall** [DRNG.15] achieve the declared security strength, supported by justification based on its algorithmic security properties.
- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/SecurityStrength	Inputs
Review I_VQ_DRNG (or I_90A) to verify that it declares a target security strength for the TOE and provides appropriate justification that it is effectively achieved, based on either the formal model of the DRNG or the correct selection of cryptographic parameters in line with [SP 800-90A].	I_VQ_DRNG I_90A

6 ENHANCED DETERMINISTIC RANDOM NUMBER GENERATORS

6.1 OBJECTIVE

303. This section defines the security requirements, evaluation inputs, terminology, and evaluation tasks applicable to the CCN-EDRNG functionality class.
304. The requirements defined in this section are based on those of the DRG.4 functionality class from AIS-20, as specified in the mathematical and technical reference [FCRNG], and have been adapted following CCN's selection criteria and evaluation procedures. Many of the requirements and evaluation tasks coincide with those defined for the CCN-DRNG functionality class, as the CCN-EDRNG builds upon it by incorporating enhanced forward secrecy capabilities. The evaluation tasks have been designed so that any CCN-EDRNG successfully evaluated under this methodology can be considered DRG.4 compliant.

6.2 CCN-EDRNG DEFINITION

305. Enhanced Deterministic Random Number Generators, denoted by EDRNG, are a type of RNG that generates random numbers through deterministic algorithms supported by the injection of fresh entropy.
306. A EDRNG is claimed to belong to the CCN-EDRNG functionality class if has been successfully evaluated under the present methodology. CCN-EDRNG is a CCN class that extends the security requirements of CCN-DRNG class by adding the enhanced forward secrecy capability.
307. The general structure of the CCN-EDRNG is shown in Figure 5.

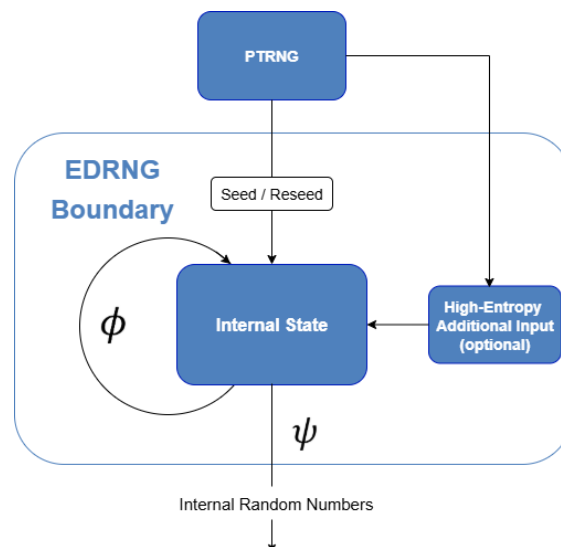


Figure 5. CCN-EDRNG Design Scheme

308. For a DRNG to comply with CCN-EDRNG functionality class:
- The DRNG must be capable of performing at least one of the following: triggering a reseeding procedure or accepting high-entropy additional input—

provided by at least 240 bits of min-entropy—through its state transition function.

- Any fresh entropy—i.e., the seed material used in the reseeding process and the high-entropy additional input—inserted into the DRNG must be provided by a validated CCN-PTRNG.

6.3 INPUTS

309. For EDRNGs, the evidence to be provided shall consist of the same inputs defined for DRNGs (**I_IMP_DRNG** and either **I_VQ_DRNG** or **I_90A**) supplemented with the specific evidence required to demonstrate compliance with the enhanced forward secrecy requirement [EDRNG.14]. This includes a clear description of the mechanism used to inject fresh entropy and justification that it guarantees at least 240 bits of min-entropy when required, in accordance with the applicable reseeding or additional input procedures.

6.4 CCN-EDRNG EVALUATION TASKS

310. This section defines the specific requirements for Enhanced Deterministic Random Number Generators, as established by CCN, which must be successfully evaluated for the TOE generator to be recognized as a CCN-EDRNG approved RNG.
311. The following list of evaluation tasks related to the CCN-EDRNG requirements have been defined:

CCN-EDRNG EVALUATION TASK	CCN-EDRNG-IMPLEMENTATION
The tester shall verify, using I_VQ_DRNG (or I_90A), that the TOE has been successfully evaluated against the [CCN-MEMeC] methodology at level CL2 or CL3. This task ensures that the TOE is correctly implemented, corresponds to its declared design, and complies with the applicable CCN cryptographic requirements.	

CCN-EDRNG EVALUATION TASK	CCN-EDRNG-ENTROPY
The tester shall review I_VQ_DRNG (or I_90A), to verify that the TOE obtains its seed material from a PTRNG that belongs to either CCN-PTRNG class or CCN-HPTRNG class.	

CCN-EDRNG EVALUATION TASK	CCN-EDRNG-ENTROPY
The tester shall also verify that the seeding and, if implemented, the reseeding procedures are sufficient to ensure that the resulting internal state contains at least 240 bits of min-entropy.	

CCN-EDRNG EVALUATION TASK	CCN-EDRNG-ALGORITHMIC
The tester shall: - Review the formal model of the TOE provided in I_VQ_DRNG and (using I_IMP_DRNG) that it is aligned to the TOE implementation. In addition, verify that it is properly employed to justify the algorithmic security properties required for CCN-EDRNG. - If the TOE implements Hash_DRBG or HMAC_DRBG, review I_90A to verify that the implementation complies with the applicable requirements in [SP 800-90A]; if compliance is confirmed, the previous requirement is considered fulfilled. - In all cases, the tester shall verify that appropriate evidence supports the claimed security strength.	

312. For the correct completion of the evaluation tasks, each one has been divided into a set of evaluation tests. The **tester shall** perform the associated mandatory tests, and also the implementation-dependent tests, if applicable, according to the TOE design.

CCN-EDRNG Evaluation Task	Evaluation Test	Test Categorization
CCN-EDRNG- IMPLEMENTATION EDRNG Implementation	CCN-EDRNG- IMPLEMENTATION/MEMeC	Mandatory
CCN-EDRNG- ENTROPY Entropy Acquisition Evaluation	CCN-EDRNG-ENTROPY/Source	Mandatory
	CCN-EDRNG-ENTROPY/Seeding	Mandatory
	CCN-EDRNG-ENTROPY/Reseeding	Mandatory

CCN-EDRNG Evaluation Task	Evaluation Test	Test Categorization
CCN-EDRNG- ALGORITHMIC Algorithmic Security Properties	CCN-EDRNG-ALGORITHMIC/90A	Impl-Dep
	CCN-EDRNG- ALGORITHMIC/FormalModel	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.2	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.3	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.5	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.6	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.7	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.8	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.3.9	Impl-Dep
	CCN-EDRNG-ALGORITHMIC/DRG.4.10	Mandatory
	CCN-EDRNG-ALGORITHMIC/DRG.3.10	Impl-Dep
	CCN-EDRNG- ALGORITHMIC/SecurityStrength	Mandatory

Table 43. Evaluation Tests defined for CCN-EDRNG Requirements

6.4.1 EVALUATION TESTS DEFINITION

313. This subsection defines the specific evaluation tests for CCN-EDRNG generators, as established by CCN, which must be performed by the tester to complete the evaluation tasks in accordance with the specific CCN-EDRNG requirements.

6.4.1.1 EDRNG IMPLEMENTATION

314. This topic defines the evaluation tests required to fulfill the **CCN-EDRNG-IMPLEMENTATION** evaluation task.

315. An Enhanced Deterministic Random Number Generator (EDRNG) is, by design, an algorithmic construction that derives its security properties from the those of the cryptographic mechanisms it employs. As such, both its design and implementation must be understood and evaluated as cryptographic mechanisms.

316. For this reason, the design and implementation of TOEs belonging to the CCN-EDRNG functionality class (as in the case of the CCN-DRNG functionality class) shall be evaluated following the [CCN-MEMeC] methodology. Its use provides assurance that the TOE implementation corresponds to the declared design, that the cryptographic mechanisms employed and their parametrization are recommended according to [CCN-STIC-221] and that the associated CCN cryptographic requirements are enforced. This implies the secure management of the EDRNG's internal state and the correct implementation of self-tests.

6.4.1.1.1 Cryptographic Evaluation

- The **TOE shall** [EDRNG.01] be evaluated in accordance with the Cryptographic Mechanisms Evaluation Methodology ([CCN-MEMeC]) at evaluation level **CL2** or **CL3** and, in the context of this evaluation, comply with the specific note listed in the table below:

Cryptographic Evaluation	
CCN Class	Notes
CCN-EDRNG	Note 70

Table 44. Cryptographic Evaluation – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-IMPLEMENTATION/MEMeC	Inputs
Verify in, I_VQ_DRNG (or I_90A) that the TOE has been successfully evaluated against the [CCN-MEMeC] methodology at level CL2 or CL3 , considering the specific note listed in Table 44.	I_VQ_DRNG I_90A

6.4.1.2 ENTROPY ACQUISITION EVALUATION

317. This topic defines the evaluation tests required to fulfill the **CCN-EDRNG-ENTROPY** evaluation task.

6.4.1.2.1 Entropy Source Verification

- The **TOE shall** [EDRNG.02] obtain the seed material (for the seeding procedure and reseeding procedure) from a PTRNG that belongs either to the CCN-PTRNG class or to the CCN-HPTRNG class.
- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ENTROPY/Source	Inputs
Verify, using I_VQ_DRNG (or I_90A), that the seed material used by the TOE for seeding and reseeding is obtained from a random number generator belonging to one of the allowed random number generator classes listed above, acting as source of randomness. Note: Of course, by definition, the random number generator employed for the seeding and reseeding of the TOE, shall have been successfully evaluated according to this methodology.	I_VQ_DRNG I_90A

6.4.1.2.2 Seeding Procedure

- The **TOE shall** [EDRNG.03]:
 - Perform its seeding procedure employing seed material generated by a RNG which complies to [EDRNG.02] and ensure that the resulting effective internal state contains at least 240 bits of min-entropy after seeding.
 - Comply with the following specific note listed in the table below:

Seeding Procedure	
CCN Class	Notes
CCN-EDRNG	Note 72

Table 45. Seeding Procedure – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ENTROPY/Seeding	Inputs
Review I_VQ_DRNG (or I_90A) to verify that:	I_VQ_DRNG I_90A

CCN-EDRNG-ENTROPY/Seeding	Inputs
- The resulting effective internal state contains at least 240 bits of min-entropy after seeding. - The TOE complies with the specific note listed in Table 45.	

6.4.1.2.3 Reseeding Procedure

318. In the case the TOE employs a reseeding procedure:

- The **TOE shall** [EDRNG.04]:
 - Perform its reseeding procedure employing seed material generated by a RNG which complies to [EDRNG.02] and shall ensure that the resulting effective internal state contains at least 240 bits of min-entropy after reseeding.
 - Comply with the following specific note listed in the table below:

Reseeding Procedure	
CCN Class	Notes
CCN-EDRNG	Note 73

Table 46. Reseeding Procedure – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ENTROPY/Reseeding	Inputs
Review I_VQ_DRNG (or I_90A) to verify that: - The resulting effective internal state contains at least 240 bits of min-entropy after seeding. - The TOE complies with the specific note listed in Table 46.	I_VQ_DRNG I_90A

6.4.1.3 ALGORITHMIC SECURITY PROPERTIES

319. The topic defines the cryptographic evaluation tests required to fulfill the **CCN-EDRNG-ALGORITHMIC** evaluation task.

320. Any EDRNG belonging to the CCN-EDRNG functionality class must fulfill a set of algorithmic security requirements related to forward secrecy, backward secrecy and enhanced backward secrecy in the presence of different adversarial models. These properties, which aim to prevent an attacker from recovering the internal state of the EDRNG, shall be properly demonstrated relying on a formal mathematical description of the EDRNG's algorithmic structure and its seeding

and reseeding procedures. The technical reference [FCRNG] provides a framework for this endeavor.

321. In practice, evaluating these algorithmic properties from a formal mathematical model can be challenging. However, for some EDRNGs based on certain standardized schemes, some of this work has already been published. Specifically, [FCRNG], in section 5.3, already provides formal conformity proofs for the Hash_DRBG and HMAC_DRBG constructions defined in [SP 800-90A], which demonstrate that they meet the algorithmic requirements presented in this evaluation task.
322. As a result, when evaluating an EDRNG based on Hash_DRBG or HMAC_DRBG, the mentioned conformity proof streamlines its evaluation. However, this only applies if the implementation is shown to fully comply with all requirements of [SP 800-90A]. If the EDRNG scheme does not meet these conditions, the full set of algorithmic evaluation tests within this task shall be performed.
323. A paradigmatic example is CTR_DRBG, for which no pre-approved formal mathematical conformity proof exists. Moreover, its variants based on AES-128 or AES-192 are known not to meet the algorithmic requirements defined in this evaluation task.

6.4.1.3.1 Hash-Based DRBGs

324. In order to verify that the implementation of a EDRNG based on the Hash_DRBG or HMAC_DRBG schemes complies with the specification defined in [SP 800-90A], the [CCN-MEMeC] evaluation required by the **CCN-EDRNG-IMPLEMENTATION** task ensures compliance with many of the relevant requirements—but not all. Therefore, the present evaluation test has been defined to assess those specific requirements of [SP 800-90A] that are not covered by the cryptographic evaluation tasks of the [CCN-MEMeC] methodology.
325. Thus, in case the TOE is based on the Hash_DRBG or HMAC_DRBG from [SP 800-90A]:
- The **TOE shall** [EDRNG.05] comply with all applicable requirements defined in [SP 800-90A], based on the DRBG scheme it implements (e.g., Hash_DRBG or HMAC_DRBG).
 - The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/90A	Inputs
Using I_90A , verify that all listed requirements are satisfied. This shall be done by reviewing that the evidence provided by the vendor in I_90A adequately supports compliance with each of the referenced [SP 800-90A] requirements.	I_90A

CCN-EDRNG-ALGORITHMIC/90A	Inputs
Note: The successful verification of this test and of CCN-DRNG-IMPLEMENTATION/MEMeC implies the successful verification of the CCN-DRNG-ALGORITHMIC/FormalModel test and the nine CCN-DRNG-ALGORITHMIC/DRG.3.X tests.	

6.4.1.3.2 DRNG Formal Model

326. The **TOE shall** [EDRNG.06] be defined by a Formal Model for its algorithmic structure, and a Formal Model for the seeding and reseeding procedures which comply with the specific notes listed in the table below:

DRNG Formal Model	
CCN Class	Notes
CCN-EDRNG	Note 74 Note 75 Note 76 Note 77

Table 47. DRNG Formal Model – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/FormalModel	Inputs
Review I_VQ_DRNG to verify that: - It provides a formal model of the TOE that complies with the specific notes listed in Table 47. - The formal model has been employed as the theoretical ground for the justification of the security requirements addressed across the rest of evaluation tests within the CCN-EDRNG-ALGORITHMIC evaluation task. Additionally, perform an implementation code review (using I_IMP_DRNG) to verify that the TOE complies with Note 77.	I_VQ_DRNG I_IMP_DRNG

6.4.1.3.3 Request Length

- The **TOE shall** [EDRNG.07] accept requests that comprise a maximum length of 2^{19} bits, and comply with the specific notes listed in the table below:

Request Length	
CCN Class	Notes
CCN-EDRNG	Note 78 Note 79

Table 48. Request Length – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.2	Inputs
Review I_VQ_DRNG to verify that: - The TOE accepts requests that comprise a maximum length of 2^{19} bits. - The TOE complies with the specific notes listed in Table 48.	I_VQ_DRNG

6.4.1.3.4 Effective Internal State

327. The effective internal state refers to the portion of the EDRNG's internal state that determines future outputs and contains usable entropy. Its size is directly related to the EDRNG's security strength.

- The **TOE shall** [EDRNG.08] have an effective internal state that comprises at least 246 bits, and comply with the specific note listed in the table below:

Effective Internal State	
CCN Class	Notes
CCN-EDRNG	Note 80

Table 49. Effective Internal State – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.3	Inputs
Review I_VQ_DRNG to verify that: - The TOE has an effective internal state that comprises at least 246 bits. - The effective internal state of the TOE complies with the specific note listed in Table 49.	I_VQ_DRNG

6.4.1.3.5 Forward Secrecy

- The **TOE shall** [EDRNG.09] provide forward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:

Forward Secrecy	
CCN Class	Notes
CCN-EDRNG	Note 81 Note 82 Note 83

Table 50. Forward Secrecy – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.5	Inputs
Review I_VQ_DRNG to verify that: - The TOE provides forward secrecy on the granularity level of internal random numbers. - The TOE complies with the specific notes listed in Table 50.	I_VQ_DRNG

6.4.1.3.6 Backward Secrecy

- The **TOE shall** [EDRNG.10] provide backward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:

Backward Secrecy	
CCN Class	Notes
CCN-EDRNG	Note 81 Note 82 Note 83

Table 51. Backward Secrecy – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.6	Inputs
Review I_VQ_DRNG to verify that: - The TOE provides backward secrecy on the granularity level of internal random numbers.	I_VQ_DRNG

CCN-EDRNG-ALGORITHMIC/DRG.3.6	Inputs
- The TOE complies with the specific notes listed in Table 51.	

6.4.1.3.7 Enhanced Backward Secrecy

- The **TOE shall** [EDRNG.11] provide enhanced backward secrecy on the granularity level of internal random numbers, and comply with the specific notes listed in the table below:
- Alternatively, provide enhanced backward secrecy on the granularity level of requests, provided that the requests meet the condition of conceptual atomicity.

Enhanced Backward Secrecy	
CCN Class	Notes
CCN-EDRNG	Note 81 Note 83 Note 84 Note 85 Note 86

Table 52. Enhanced Backward Secrecy – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.7	Inputs
Review I_VQ_DRNG and perform a source code review (using I_IMP_DRNG) to verify that: - The TOE provides enhanced backward secrecy on the granularity level of internal random numbers. - The TOE complies with the specific notes listed in Table 52. Note: For the analysis of enhanced backward secrecy, the tester shall not only consider the formal model, but also the TOE implementation.	I_VQ_DRNG I_IMP_DRNG

6.4.1.3.8 Additional Input

- If additional input is permitted, the **TOE shall** [EDRNG.12] accept additional input that does not weaken the strength of the EDRNG even if an attacker is able to control the additional input, and comply with the specific note listed in the table below:

Additional Input	
CCN Class	Notes
CCN-EDRNG	Note 81

Table 53. Additional Input – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.8	Inputs
Review I_VQ_DRNG and perform a source code review (using I_IMP_DRNG) to verify that: - The TOE accepts additional input that does not weaken the strength of the EDRNG even if an attacker is able to control the additional input. - The TOE complies with the specific note listed in Table 53.	I_VQ_DRNG I_IMP_DRNG

6.4.1.3.9 DRNG Functions

- The **TOE shall** [EDRNG.13] implement a cryptographic and one-way state transition function, ϕ , and a cryptographic output function, ψ .
- The **tester shall** perform the following evaluation test:

CCN-DRNG-ALGORITHMIC/DRG.3.9	Inputs
Review I_VQ_DRNG and perform an implementation code review (using I_IMP_DRNG) to verify that: - The transition function, ϕ, is one-way and cryptographic function, and - the output function, ψ, is cryptographic.	I_VQ_DRNG I_IMP_DRNG

6.4.1.3.10 Enhanced Forward Secrecy

328. This subsection defines the requirements for achieving enhanced forward secrecy in DRNGs belonging to functionality class CCN-EDRNG, which constitutes the fundamental difference with respect to CCN-DRNG functionality class.

329. Enhanced forward secrecy can only be achieved by periodically introducing fresh entropy, through reseeding procedures or by properly incorporating high-entropy additional input into the internal state.

- The **TOE shall** [EDRNG.14] provide enhanced forward secrecy and comply with the specific note listed in the table below:

Enhanced Forward Secrecy	
CCN Class	Notes
CCN-EDRNG	Note 88

Table 54. Enhanced Forward Secrecy – Notes

Note 88 [DRG.4.10]
Enhanced forward secrecy shall be achieved by the reseeding procedure or by high-entropy additional input such as that the effective internal state has a min-entropy of, at least, 240 bits. Both methods can be applied by the EDRNG. By employing the above-mentioned methods, enhanced forward secrecy can be provided on demand, based on a designated condition or by following a time-based policy. As a minimum requirement, until the next reseeding procedure, or the next high-entropy additional input is injected, at most 2^{17} internal random number bits shall be generated. The concrete procedure followed by the TOE to achieve enhanced forward secrecy shall be documented and justified.

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.4.10	Inputs
Review I_VQ_DRNG (or I_90A) to verify that: - The TOE provides enhanced forward secrecy. - The TOE complies with the specific note listed in Table 54.	I_VQ_DRNG I_90A

6.4.1.3.11 Statistical Inconspicuousness

- The **TOE shall** [EDRNG.15] output internal random numbers that have statistical inconspicuousness and comply with the specific note listed in the table below:

Statistical Inconspicuousness	
CCN Class	Notes
CCN-EDRNG	Note 87

Table 55. Statistical Inconspicuousness – Notes

- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/DRG.3.10	Inputs
Review I_VQ_DRNG to verify that: - The TOE outputs internal random numbers that have statistical inconspicuousness. - The TOE complies with the specific note listed in Table 55. If the tester suspects that the TOE design does not have the property of statistical inconspicuousness, targeted statistical tests should be applied to test for these possibilities.	I_VQ_DRNG

6.4.1.3.12 Security Strength

- The **TOE shall** [EDRNG.16] achieve the declared security strength, supported by justification based on its algorithmic security properties.
- The **tester shall** perform the following evaluation test:

CCN-EDRNG-ALGORITHMIC/Security Strength	Inputs
Review I_VQ_DRNG (or I_90A) to verify that it declares a target security strength for the TOE and provides appropriate justification that it is effectively achieved, based on either the formal model of the DRNG or the correct selection of cryptographic parameters in line with [SP 800-90A].	I_VQ_DRNG I_90A

7 ANNEX A: TRACEABILITY OF EVALUATION TASKS

330. This annex summarizes the evaluation tasks related to each CCN functionality class. The objective is to provide the tester with the complete list of evaluation tasks to be performed in order to evaluate each type RNG.

7.1 CCN-PTRNG EVALUATION TASKS

331. The following table summarizes the evaluation tasks to be performed for TOEs belonging to the CCN-PTRNG functionality class:

Summary of CCN-PTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
CCN-PTRNG-DESIGN	CCN-PTRNG-DESIGN/BlockDesign Notes Note 1 [Functional Design Description] Note 2 [Analog Noise Source] Note 3 [Digitizer] Note 4 [Post-Processing Block] Note 5 [Health Tests Block] Note 6 [Interfaces Identification]
	CCN-PTRNG-DESIGN/HWDesign
CCN-PTRNG-MODEL	CCN-PTRNG-MODEL/AnalogModel Notes Note 7 [Physical Phenomenon] Note 8 [Deterministic Noise] Note 9 [Analog Stochastic Model Definition] Note 10 [Assumptions]
	CCN-PTRNG-MODEL/DigitalModel Notes Note 11 [Digitization Function] Note 12 [Digital Stochastic Model Definition] Note 13 [Auxiliary Processes] Note 14 [Avoiding Post-Processing] Note 15 [PTG.2.1]
	CCN-PTRNG-MODEL/ModelEntropy Notes Note 16 [Entropy Bound Formulation] Note 17 [Entropy Bound Computation]

Summary of CCN-PTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
	CCN-PTRNG-MODEL/ParameterSets Notes Note 18 [Good Parameters]
CCN-PTRNG-MODELVAL	CCN-PTRNG-MODELVAL/Assumptions Notes Note 19 [Validation Conditions] Note 20 [Pre-silicon Design Validation] Note 21 [Analog Model Assumptions] Note 22 [PTG.2.1 Validation]
	CCN-PTRNG-MODELVAL/ParameterEstimation Notes Note 19 [Validation Conditions] Note 23 [Noise Component Separation] Note 24 [Estimation Methods] Note 25 [Real Parameters]
	CCN-PTRNG-MODELVAL/EntropyValidation Notes Note 19 [Validation Conditions] Note 26 [Prediction Estimate]
CCN-PTRNG-CONDITIONING	CCN-PTRNG-CONDITIONING/InternalEntropy Notes Note 27 [PTG.2.2] Note 28 [Post-Processing Types] Note 29 [Algorithmic Post-Processing] Note 30 [Cryptographic Post-Processing]
	CCN-PTRNG-CONDITIONING/Conformity
CCN-PTRNG-HEALTH	CCN-PTRNG-HEALTH/OnlineTest Notes Note 31 [Tailored to the Stochastic Model] Note 32 [Performance Justification] Note 33 [Testing Real Parameters] Note 34 [Sufficiently Soon]
	CCN-PTRNG-HEALTH/TotalFailure Notes

Summary of CCN-PTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
	Note 35 [Immediate Detection] Note 36 [Failure Modes] Note 37 [Implementation]
	CCN-PTRNG-HEALTH/StartUp Notes Note 38 [Warm-Up] Note 39 [Detection Objective] Note 40 [Known Answer Test]
	CCN-PTRNG-HEALTH/Alarms Notes Note 41 [Health Testing Alarms]
	CCN-PTRNG-BLACKBOX/AIS31 Notes Note 42 [Validation Conditions] Note 43 [AIS-31 Test Suite] Note 44 [Evaluation Rules]
CCN-PTRNG-BLACKBOX	CCN-PTRNG-BLACKBOX/90B Notes Note 42 [Validation Conditions] Note 45 [Block Assessment] Note 46 [Restart Assessment]

Table 56. Summary of CCN-PTRNG Evaluation Tasks

7.2 CCN-HPTRNG EVALUATION TASKS

332. The following table summarizes the evaluation tasks to be performed for TOEs belonging to the CCN-HPTRNG functionality class:

Summary of CCN-HPTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
CCN-HPTRNG-DESIGN	CCN-HPTRNG-DESIGN/CompositeDesign Notes Note 47 [Integration Evidence]
CCN-HPTRNG-SOURCE	CCN-PTRNG-SOURCE/PTRNG Notes Note 48 [Inner PTRNG Health]
CCN-HPTRNG-POSTPROCESSING	CCN-PTRNG-POSTPROCESSING/DRNG
	CCN-PTRNG-POSTPROCESSING /EntropyInjection Notes Note 49 [Compression Rate] Note 50 [Reseeding Option] Note 51 [Entropy Injection Justification]

Table 57. Summary of CCN-HPTRNG Evaluation Tasks

7.3 CCN-NPTRNG EVALUATION TASKS

333. The following table summarizes the evaluation tasks to be performed for TOEs belonging to the CCN-DRNG functionality class:

Summary of CCN-NPTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
CCN-NPTRNG-DESIGN	CCN-NPTRNG-DESIGN/Sources Notes Note 52 [Noise Source Conditions]
	CCN-NPTRNG-DESIGN/EntropyPool Notes Note 53 [Entropy Counter] Note 54 [Mixing Entropy] Note 55 [NTG.1.4]
	CCN-NPTRNG-DESIGN/DRNG Notes Note 56 [Fresh Entropy]
CCN-NPTRNG-ENTROPY	CCN-DRNG-ENTROPY/HeuristicAnalysis Notes Note 57 [Unpredictability Justification] Note 58 [Heuristic Estimate] Note 59 [Rigor Requirement]
	CCN-DRNG-ENTROPY/InternalEntropy Notes Note 60 [Entropy Content]
CCN-NPTRNG-HEALTH	CCN-DRNG-HEALTH/ContinuousTest Notes Note 61 [Continuous Tests Specifications] Note 62 [Failure Modes] Note 63 [Performance Justification - NP]

Summary of CCN-NPTRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
	CCN-DRNG-HEALTH/StartUp Notes Note 61 [Continuous Tests Specifications] Note 62 [Failure Modes] Note 63 [Performance Justification - NP] Note 64 [Detection Objective] Note 65 [Start-Up Test Specifications] Note 66 [Known Answer Test]
	CCN-DRNG-HEALTH/Alarms Notes Note 67 [Health Testing Alarms]
CCN-NPTRNG-BLACKBOX	CCN-DRNG-BLACKBOX/90B Notes Note 68 [Block Assessment] Note 69 [Restart Assessment]

Table 58. Summary of CCN-NPTRNG Evaluation Tasks

7.4 CCN-DRNG EVALUATION TASKS

334. The following table summarizes the evaluation tasks to be performed for TOEs belonging to the CCN-DRNG functionality class:

Summary of CCN-DRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
CCN-DRNG-IMPLEMENTATION	CCN-DRNG-IMPLEMENTATION/MEMeC
	Notes Note 70 [Self-Tests]
CCN-DRNG-ENTROPY	CCN-DRNG-ENTROPY/Source
	CCN-DRNG-ENTROPY/Seeding
	Notes Note 71 [DRG.3.4] Note 72 [Seeding Evidence]
	CCN-DRNG-ENTROPY/Reseeding
CCN-DRNG-ALGORITHMIC	Notes Note 71 [DRG.3.4] Note 73 [Reseeding Evidence]
	CCN-DRNG-ALGORITHMIC/90A
	CCN-DRNG-ALGORITHMIC/FormalModel
	Notes Note 74 [Algorithmic Structure Formal Model] Note 75 [Seeding and Reseeding Formal Model] Note 76 [Parameters] Note 77 [Link to Implementation]
	CCN-DRNG-ALGORITHMIC/DRG.3.2
	Notes Note 78 [Request Counter] Note 79 [Zeroize Non-Output Bits]
	CCN-DRNG-ALGORITHMIC/DRG.3.3
	Notes Note 80 [Attacker Assumption.1]
	CCN-DRNG-ALGORITHMIC/DRG.3.5
	Notes Note 81 [Algorithmic Properties.1] Note 82 [Attacker Assumption.2]

Summary of CCN-DRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
	Note 83 [Security Strength]
	CCN-DRNG-ALGORITHMIC/DRG.3.6
	Notes Note 81 [Algorithmic Properties.1] Note 82 [Attacker Assumption.2] Note 83 [Security Strength]
	CCN-DRNG-ALGORITHMIC/DRG.3.7
	Notes Note 81 [Algorithmic Properties.1] Note 83 [Security Strength] Note 84 [Algorithmic Properties.2] Note 85 [Attacker Assumption.3] Note 86 [One-way function]
	CCN-DRNG-ALGORITHMIC/DRG.3.8
	Notes Note 81 [Algorithmic Properties.1]
	CCN-DRNG-ALGORITHMIC/DRG.3.9
	CCN-DRNG-ALGORITHMIC/DRG.3.10
	Notes Note 87 [Justification]
	CCN-DRNG-ALGORITHMIC/SecurityStrength

Table 59. Summary of CCN-DRNG Evaluation Tasks

7.5 CCN-EDRNG EVALUATION TASKS

335. The following table summarizes the evaluation tasks to be performed for TOEs belonging to the CCN-EDRNG functionality class:

Summary of CCN-EDRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
CCN-EDRNG-IMPLEMENTATION	CCN-EDRNG-IMPLEMENTATION/MEMeC
	Notes Note 70 [Self-Tests]
CCN-EDRNG-ENTROPY	CCN-EDRNG-ENTROPY/Source
	CCN-EDRNG-ENTROPY/Seeding
	Notes Note 72 [Seeding Evidence]
	CCN-EDRNG-ENTROPY/Reseeding
CCN-EDRNG-ALGORITHMIC	Notes Note 73 [Reseeding Evidence]
	CCN-EDRNG-ALGORITHMIC/90A
	CCN-EDRNG-ALGORITHMIC/FormalModel
	Notes Note 74 [Algorithmic Structure Formal Model] Note 75 [Seeding and Reseeding Formal Model] Note 76 [Parameters] Note 77 [Link to Implementation]
	CCN-EDRNG-ALGORITHMIC/DRG.3.2
	Notes Note 78 [Request Counter] Note 79 [Zeroize Non-Output Bits]
	CCN-EDRNG-ALGORITHMIC/DRG.3.3
	Notes Note 80 [Attacker Assumption.1]
	CCN-EDRNG-ALGORITHMIC/DRG.3.5
	Notes Note 81 [Algorithmic Properties.1] Note 82 [Attacker Assumption.2] Note 83 [Security Strength]

Summary of CCN-EDRNG Evaluation Tasks	
Evaluation Task	Evaluation Test Code
	CCN-EDRNG-ALGORITHMIC/DRG.3.6 Notes Note 81 [Algorithmic Properties.1] Note 82 [Attacker Assumption.2] Note 83 [Security Strength]
	CCN-EDRNG-ALGORITHMIC/DRG.3.7 Notes Note 81 [Algorithmic Properties.1] Note 83 [Security Strength] Note 84 [Algorithmic Properties.2] Note 85 [Attacker Assumption.3] Note 86 [One-way function]
	CCN-EDRNG-ALGORITHMIC/DRG.3.8 Notes Note 81 [Algorithmic Properties.1]
	CCN-EDRNG-ALGORITHMIC/DRG.3.9
	CCN-EDRNG-ALGORITHMIC/DRG.4.10 Notes Note 88 [DRG.4.10]
	CCN-EDRNG-ALGORITHMIC/DRG.3.10 Notes Note 87 [Justification]
	CCN-EDRNG-ALGORITHMIC/SecurityStrength

Table 60. Summary of CCN-EDRNG Evaluation Tasks

8 REFERENCES

[CCN-MEGA-EV]	CCN-MEGA Evidence Matrix v1.0. July 2025.
[CCN-MEMeC]	CCN Cryptographic Mechanisms Evaluation Methodology v2.0. June 2025.
[CCN-STIC-221]	CCN. Guía de Mecanismos Criptográficos autorizados por el CCN v2.1. October 2024.
[FCRNG]	BSI. A Proposal for Functionality Classes for Random Number Generators, version v3.0, 2024.
[JEPROC]	<i>Jitter Entropy Library v3.4.0–v3.6.12 Entropy Source Opportunistically Essentially-IID Heuristic Assessment Procedure</i> , Version 1.01, KeyPair Consulting Inc., 31 March 2025
[SP 800-90A]	NIST. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. National Institute of Standard and Technology, Special Publication, SP 800-90A, Rev. 1, 2015.
[SP 800-90B]	NIST. Recommendation for the Entropy Sources Used for Random Bit Generation. National Institute of Standards and Technology, Special Publication SP 800-90B, January 2018.
[SP 800-90C]	NIST. Recommendation for Random Bit Generator Constructions. National Institute of Standards and Technology, Special Publication SP 800-90C, Fourth Draft, 2024.

9 ABBREVIATIONS

CCN	Centro Criptológico Nacional
CL2	MEMEC Certification Level 2
CL3	MEMEC Certification Level 3
CPU	Central Processing Unit
DRBG	Deterministic Random Bit Generator
DRNG	Deterministic Random Number Generator
HDL	Hardware Description Language
I_HEALTH	Health Testing Evidence
IID	Independent and Identically Distributed
I_IMP	Implementation Evidence
I_IRN	Output Entropy and Post-Processing Justification
I_SM	Stochastic Model
I_TDS	Technical Design Specification
I_TEST	Black-Box Testing Evidence
I_VAL	Stochastic Model Validation Report
KAT	Known Answer Test
LFSR	Linear Feedback Shift Register
NPTRNG	Non-Physical True Random Number Generator
PTRNG	Physical True Random Number Generator
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman Mechanism
SSP	Sensitive Security Parameter
TOE	Target Of Evaluation
TRNG	True Random Number Generator

