

IBM Power HMC CPU Jitter Entropy Source version 3.4.0

SP 800-90B Non-Proprietary Public Use Document

Document Version: 1.1

Document Date: 2026-10-23

IBM Corporation

1 New Orchard Road

Armonk, NY 10504-1722

www.ibm.com

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

© 2026 IBM Corporation / atsec information security corporation

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Table of Contents

1 Description....2
2 Security Boundary....2
3 Operating Conditions....3
4 Configuration Settings....4
5 Physical Security Mechanisms....4
6 Conceptual Interfaces....4
7 Min-Entropy Rate....4
8 Health Tests....4
8.1 Start-up Health Tests....5
8.2 Continuous Health Tests....5
8.2.1 Repetition Count Test....5
8.2.2 Adaptive Proportion Test....5
8.2.3 Stuck Test....5
8.3 On-Demand Health Tests....5
8.4 Error Codes....6
9 Maintenance....6
10 User Verification....6
11 Vendor Permissions and Relationship....6

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

1 Description

The IBM Power HMC CPU Jitter Entropy Source version 3.4.0 is a non-physical, software-based entropy source. Its purpose is to seed the kernel DRBGs. The noise generation of this entropy source is based on the tiny variations in the execution time of the same piece of code. The execution time of this piece of code is made unpredictable by the complexity of the different hardware components that comprise modern CPUs and the different internal states that the operating system can have at a certain point in time.

The entropy source is validated as conformant to SP 800-90B (2018) and FIPS 140-3 IG D.J (November 5, 2021), D.K (April 16, 2026), D.O (May 04, 2021).

The entropy source was tested on the operational environments listed in Table 2. The noise source was tested under the assumption that its output is non-IID. This entropy source is within a multi-chip standalone embodiment.

Table 1: Entropy source and operational environment summary

Characteristic	Description
Vendor	IBM Corporation
Entropy source name	IBM Power HMC CPU Jitter Entropy Source
Standard, IGs	SP800-90B (2018) IG D.J (2021-11-05), IG D.K (2026-04-16), IG D.O (2021-05-04)
Entropy category	Non-Physical (NP)
Entropy source version	3.4.0
Operational environment	HMC 1121 on IBM POWER9
Embodiment and type	Multi-Chip Standalone

Table 2: Operational environment and version.

Manufacturer	Model	Operating System	Processor
IBM	7063-CR2	HMC 1121	POWER9

2 Security Boundary

The boundary for this non-physical, software-based entropy source is the executable binary file. It is compiled from the C code that implements it (i.e., kernel source code).

Figure 1 depicts the overall design of the entropy source and its core operations.

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

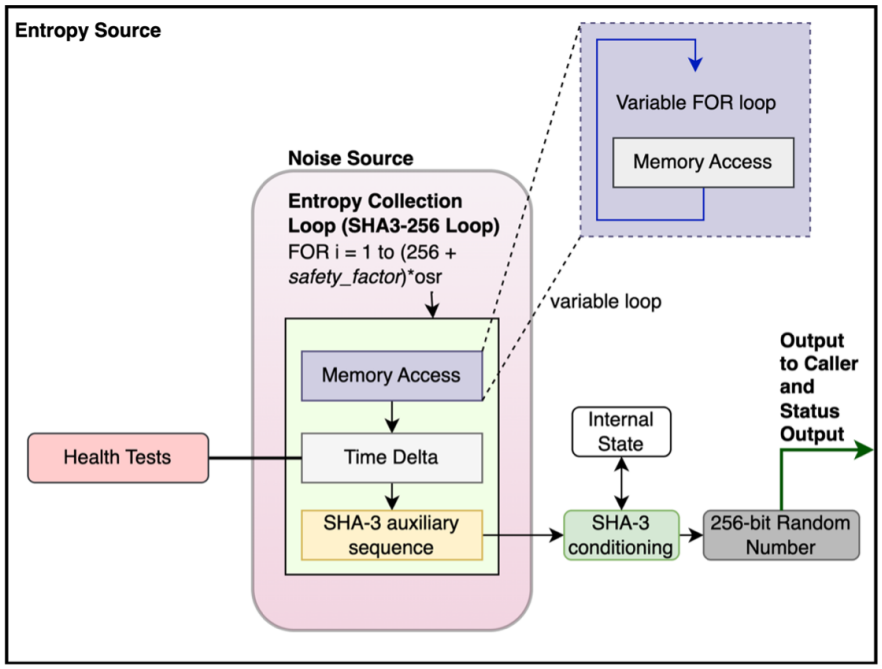


Figure 1: CPU Jitter 3.4.0 Design

The noise source is implemented by collecting and accumulating variances of the execution time of a defined set of instructions. The time jitter of the execution time is measured over time variances of variable memory accesses and variances in the execution time of a defined set of instructions, which includes an implementation of SHA3-256.

The output of the noise source is conditioned by a SHA3-256 conditioning component.

If the Repetition Count Test (RCT) or the Adaptive Proportion Test (APT) health tests fail, the noise data is discarded, the entropy source halts without outputting any data, and a failure code is returned to the caller.

3 Operating Conditions

The noise source is non-physical, and thus the operating conditions are inherited from the operational environment in which the entropy source resides, as shown in Table 3 below.

Table 3: Operating Conditions for the Operational Environment

Manufacturer / Model	Temperature	Voltage	Relative Humidity	Base Clock Speed	Cache Sizes
IBM / 7063-CR2	5°C - 40°C	100 - 127 V ac or 200 - 240 V ac	8% - 80%	2.3 GHz	L1i: 6 x 32 KiB L1d: 6 x 32 KiB L2: 4 x 512 KiB L3: 4 x 10 MiB

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

4 Configuration Settings

None of the analysis-relevant parameters can be configured, and no configuration is required for the entropy source to operate as validated.

5 Physical Security Mechanisms

The noise source is non-physical. The physical security mechanisms only apply to the hardware component of the operational environment in which the entropy source is installed, and thus the entropy source inherits those mechanisms.

6 Conceptual Interfaces

The entropy source provides the following interfaces:

- jent_kcapi_init(): Initializes the entropy source context, including the configuration of the CPU timer or the internal timer. This function returns 0 if the entropy source context is initialized successfully.
- crypto_rng_generate(): Obtains conditioned entropy from the Jitter RNG entropy source and returns the associated health test status. This interface corresponds to the GetEntropy() and HealthTest() conceptual interfaces defined in SP 800-90B. To request entropy, a consumer within the kernel space shall first obtain a handle to the jitterentropy_rng algorithm through the Linux Kernel Crypto API, and then invoke crypto_rng_generate() using that handle. Internally, the Linux Kernel Crypto API invokes the Jitter RNG implementation (jent_kcapi_random()) to generate the requested entropy. The function returns 0 if the entropy is generated successfully.
- jent_measure_jitter(): Obtains raw noise data for testing purposes. This interface corresponds to the GetNoise() conceptual interface from SP 800-90B.
- jent_kcapi_cleanup(): Zeroizes and frees the given entropy collector instance.

7 Min-Entropy Rate

The entropy source provides 256 bits of entropy per 256-bit conditioned output block (i.e., full entropy).

8 Health Tests

The entropy source implements the following continuous health tests:

- Start-up health tests
- Continuous health tests
- On-demand health tests

The continuous health tests implement a Repetition Count Test (RCT), an Adaptive Proportion Tests (APT), and a developer-defined Stuck Test.

As allowed by Section 4.3 of SP 800-90B, the entropy source defines two types of health test failures for the RCT and the APT: intermittent failures and permanent failures.

An intermittent failure is characterized by a false positive probability of $\alpha = 2^{-30}$, which lies within the recommended range of $2^{-40} \leq 2^{-20}$. When an intermittent failure is detected, the CPU Jitter RNG is automatically reset (which includes clearing the entropy pool and resetting the conditioning component), and the caller is notified of this failure.

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

A permanent failure is characterized by a false positive probability of $\alpha_p = 2^{-60}$. When a permanent failure is detected, the CPU Jitter RNG is also reset, but the Linux kernel that contains this entropy source immediately enters the error state. In practice, this results in a kernel panic.

8.1 Start-up Health Tests

Start-up health tests conduct the Adaptive Proportion Test and the Repetition Count Test across 1024 time deltas. The data is discarded after the start-up tests have completed successfully. Any health test failure during the start-up tests will always be treated as a permanent failure, which results in a kernel panic.

8.2 Continuous Health Tests

8.2.1 Repetition Count Test

The Repetition Count Test conforms to SP800-90B, Section 4.4.1.

- Intermittent RCT failure alpha value $\alpha = 2^{-30}$.
- Permanent RCT failure alpha value $\alpha = 2^{-60}$.
- Intermittent RCT failure cutoff value $C = 31$.
- Permanent RCT failure cutoff value $C = 61$.

8.2.2 Adaptive Proportion Test

The Adaptive Proportion Tests conforms to SP800-90B, Section 4.4.2.

- Window size $W = 512$.
- Intermittent APT failure alpha value $\alpha = 2^{-30}$.
- Permanent APT failure alpha value $\alpha = 2^{-60}$.
- Intermittent APT failure cutoff value $C = 325$.
- Permanent APT failure cutoff value $C = 355$.

8.2.3 Stuck Test

The stuck test computes the first, second, and third discrete derivatives of the time delta (the noise sample). If any of these derivatives are zero, then the received time delta is considered stuck. In this case, the noise sample is not used as input to the SHA3-256 conditioning function, and another sample needs to be fetched (the entropy value is not counted). The stuck test then triggers the RCT for further processing, such that the stuck test itself does not trigger a health test failure. The second derivative is in fact the RCT itself.

8.3 On-Demand Health Tests

On-demand health tests of the noise source may be performed by rebooting the operational environment, which results in the immediate execution of the start-up tests. The data used for the on-demand health tests is discarded after successful completion.

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

8.4 Error Codes

When entropy is requested through the crypto_rng_generate() function, the return code indicates the status of the entropy source. A return code of -EAGAIN indicates an intermittent health test failure, whereas a return code of -EFAULT indicates a permanent health test failure. The following error codes are defined:

- -3: Permanent health test failure.
- -2: Intermittent health test failure.
- -1: All other errors.

9 Maintenance

There are no maintenance requirements as this is a software-based entropy source.

10 User Verification

The user shall verify that the Operating Environment matches a tested configuration listed in Table 2. This verification should confirm the processor model, the operating system and its version. The operating system version can be identified using the `lshmc` command:

```
> lshmc -V
"version= Version: 11
Release: 2
Service Pack: 1121
...
```

Use of an OE not listed in Table 2 is outside the scope of this certificate.

11 Vendor Permissions and Relationship

The ESV certificate is “Reuse restricted to vendor”. Parties other than the vendor may use the certificate only with written and signed permission from the vendor’s point of contact (as indicated on the ESV certificate).

© 2026 IBM Corporation / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.