SP 800-90B Non-Proprietary Public Use Document of Entropy Source for Quantum Origin

Version 3.4.1

Document Version: 1.2

Document Date: 2024-11-26

Prepared by:
atsec information security corporation
4516 Seton Center Parkway, #250
Austin, TX 78759
www.atsec.com

Table of Contents

1 Description

The Entropy Source for Quantum Origin is a non-physical entropy source based upon CPU Time Jitter RNG version 3.4.1 with two additional conditioning functions, thus forming a chain of conditioning functions. The chain of conditioning functions is thus:

1. A vetted SHA3-256 conditioning function.
2. A non-vetted, strong seeded randomness extractor with a quantum seed.
3. A vetted SHA3-512 conditioning function, which provides the output of the entropy source.

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 noise source was tested under the assumption that its output is non-IID.

The entropy source provides full entropy bits at its output, according to the NIST definition, of 512 bits of entropy per 512 bits of output.

The entropy source was tested on the operational environment listed in Table 1.

Table 1: Tested operational environments.

HW Platform	Operating System	Processor
Dell PowerEdge R650xs	Red Hat Enterprise Linux 9.4 (Plow)	Intel® Xeon® Ice Lake 4309Y

2 Security Boundary

The boundary for this non-physical entropy source is the Entropy Source for Quantum Origin executable binary file. It is compiled from the C code that implements it.

The noise source is implemented by collecting and accumulating time jitter variances caused by memory accesses and the execution time of a defined set of instructions. The accumulation of jitter variances in the form of concatenated time deltas is fed through the SHA3-256 vetted conditioning function of the entropy source. The outputs from this SHA3-256 function are accumulated as a concatenation until 1530 bits are available and are put through the second, non-vetted conditioning function, called an extractor function. The extractor uses a quantum seed as parameter that does not contribute entropy, and this seed does not influence the extractor's statistical properties. Thus, re-use of the evaluated quantum seed on all installations meets the requirements of SP 800-90B and does not affect the security of the entropy source.

The extractor provides 640-bit outputs. The outputs of the extractor function are processed by the third conditioning component, a vetted SHA3-512 function, which consumes 640 bits at a time and produces a 512-bit output per output of the extractor.

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

If the Repetition Count Test (RCT) or the Adaptive Proportional 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.

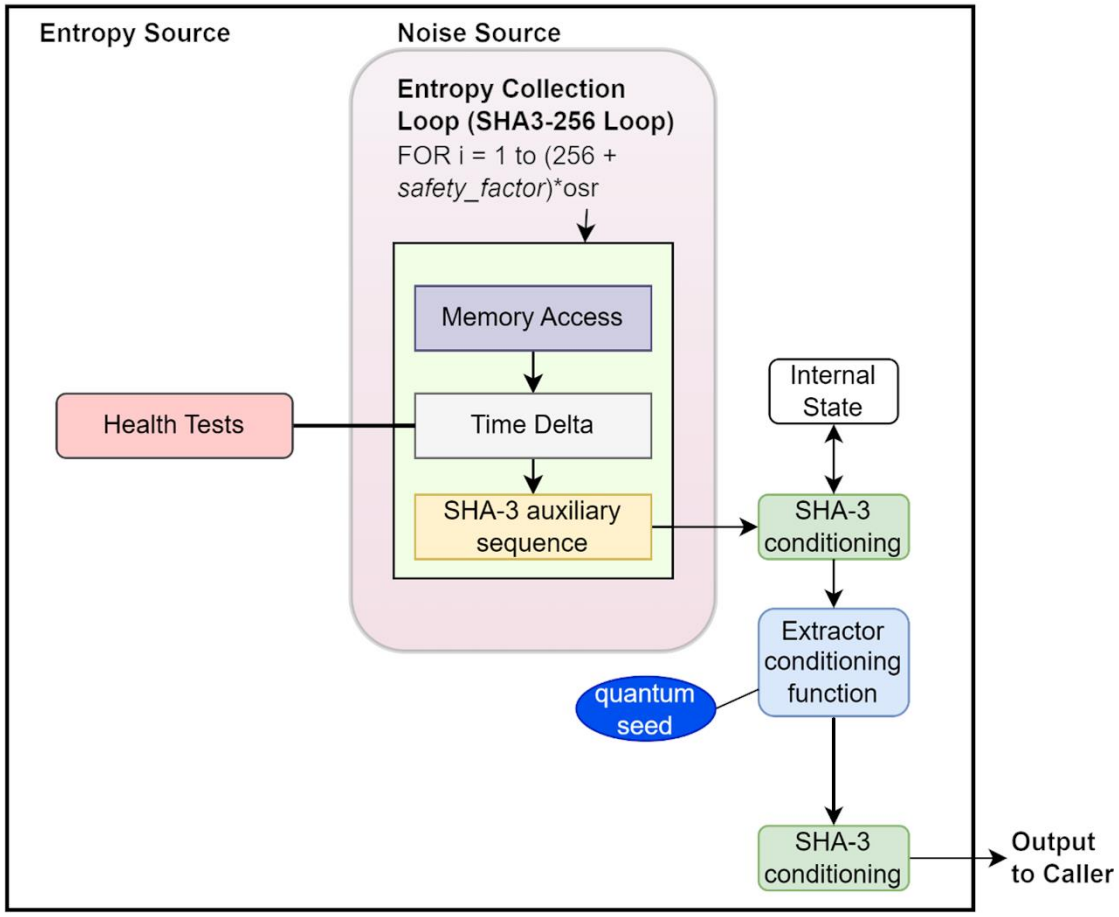

Figure 1: Security boundary of the entropy source.

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 is installed, as shown in Table 2.

Temperature	Humidity (non-condensing)	Voltage
4 to 40°C	8% to 80%	100 to 240 VAC 240 VDC

Table 2: Operating Conditions of Dell PowerEdge R650xs.

4 Configuration Settings

The entropy source has configurable settings, but the tested configuration is described here. Any modification of any configuration item listed here results in a non-validated instance of the entropy source, thus no modification of the configuration settings is allowed.

The entropy source was tested using the external CPU timer which exists on most modern CPUs. In this entropy source, only the external CPU timer is allowed as validated. Enabling the internal timer results in a non-validated entropy source.

A default value of the quantum seed is hard-coded in the entropy source. This is the only value of the quantum seed that can be used.

The timer settings are defined in the file jitterentropy.h:

#define JENT_CONF_ENABLE_INTERNAL_TIMER /* enables internal non-CPU timer. This setting shall be disabled */

#define JENT_FORCE_INTERNAL_TIMER (1<<3) /* forces the use of the internal timer. This setting shall be disabled */

#define JENT_DISABLE_INTERNAL_TIMER (1<<4) /* disable the potential use of the internal timer. This setting shall be enabled */

The other settings in jitterentropy.h are listed below. Any changes invalidate the ESV certificate.

#define JENT_FORCE_FIPS (1<<5) /* Force FIPS compliant mode including full SP 800-90B compliance */

#define ENTROPY_SAFETY_FACTOR 0 (only affects the first conditioner, SHA3-256)

#define JENT_MEMORY_BITS 17

#define JENT_MEMORY_SIZE (UINT32_C(1)<<JENT_MEMORY_BITS)

#define JENT_MEMORY_ACCESSLOOPS 128

5 Physical Security Mechanisms

The noise source is non-physical. The physical security mechanisms 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_read_entropy(): Obtains conditioned entropy for the caller. This is the main function of the entropy source, the one that shall be used to request entropy data. The entropy gathering logic creates up to 512 bits per invocation. This interface corresponds to the GetEntropy() conceptual interface from SP800-90B.
- jent_hash_time(): Obtains raw noise data for testing purposes. This interface corresponds to the GetNoise() conceptual interface from SP800-90B.

7 Min-Entropy Rate

$H_{submitter}$ = 1/3 bit per 64 bits of time delta.

The time delta noise sample size is 64 bits; however, this value is modified by dividing by a fixed greatest common divisor (GCD) value computed from some initial time deltas. This has the effect of shifting varying bits towards the least significant bit (LSB) positions of the time delta and eliminating non-varying bit positions. It is this value that is used by the health tests.

The modified 64-bit time deltas are concatenated to gather sufficient entropy for the SHA3-256 vetted conditioning function. 1530 bits of output from the SHA3-256 are input to the non-

vetted extractor function which outputs 640 bits. These are then input to the final SHA3-512. Each 512-bit output from the SHA3-512 has 512-bits of entropy.

Therefore, the entropy source output has full entropy.

8 Health Tests

The entropy source implements the following continuous health tests[1]:

- Repetition Count Test conforming to SP 800-90B section 4.4.1.
  - osr =3
  - $H = 1/3$ bit of entropy per modified time delta
  - alpha value of $\alpha = 2^{-30}$.
  - Cutoff value $C = 91$
- Adaptive Proportion test conforming to SP 800-90B section 4.4.2.
  - osr =3
  - $W = 512$
  - $H = 1/3$ bit of entropy per modified time delta
  - alpha value of $\alpha = 2^{-30}$.
  - Cutoff value $C = 458$.
- Stuck (Non-Permanent) Test: The stuck test computes the first, second and third discrete derivatives of the time value that will be processed by SHA3-256. If any of these derivatives are zero, then the received time delta is considered stuck. In this case the input state to SHA3-256 is not updated, and the entropy value is not counted. The stuck test then triggers the RCT for further processing. The second derivative is in fact the RCT itself.
- Lag Predictor Test: The goal of this test is to detect a failure mode in which the outputs may become mostly deterministic. In essence, this test constructs a scoreboard and tracks the number of times that a subpredictor was correct. The subpredictor that scored the most correct predictions is used to predict the next value of a series. The lag predictor test is configured in this entropy source with the following parameters:
  - $\alpha = 2^{-22}$
  - Window size: $window\_size = 131072$
  - Lag history size: $lag\_history\_size = 8$
  - Global cutoff = $InverseBinomialCDF = CRITBINOM\left(n = window_{size} - lag_{history_{size}}; p = 2^{-\frac{1}{osr}}; 1 - \alpha\right) = 104761$
  - Local cutoff = 111

The continuous-health tests are applied to each new sample obtained from the noise source. Whenever a failure is detected during the health testing specifically for the RCT and APT, entropy data is not returned to the caller; instead, a failure code is returned to enable the caller to acknowledge the failure. The entropy source then halts and will refuse new requests for entropy. Upon return of the failure code, the caller shall attempt to reset or reboot the

[1] osr = oversampling rate

entropy source or return an error to its own operator. The stuck test is considered non-permanent, as positive stuck tests will be registered but will not immediately halt the entropy source.

Startup tests conduct the same set and parameters of the continuous health tests on 1024 samples of noise data. The data is discarded after the startup tests have been completed successfully.

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. Typically, this entropy source designed for user space cannot be reloaded without restarting the executable. Similarly, the data used for the on-demand health tests are discarded after successful completion.

The following error codes are defined for jent_read_entropy():

-1 entropy_collector is NULL

-2 RCT failed

-3 APT test failed

-4 The timer cannot be initialized

-5 LAG failure.

-38021 Memory allocation for seed failed

-38180 Memory allocation for polynomial coefficients in extractor failed

-38270 Memory allocation for extractor context failed

9 Maintenance

There are no maintenance requirements as the entropy source is software based.

10 Required Testing

To test the entropy source, raw data samples must be collected using a test harness that is capable of accessing the jent_read_hashtime() noise interface from the entropy source, and also the final entropy output of the entropy source using the jent_read_entropy() interface. The test harness and accessory tools must be supplied by the vendor.

Raw noise data samples consisting of at least 1,000,000 bits must be collected from the operational environment at its normal operating conditions and processed by the SP 800-90B entropy tool that is provided by NIST. The expected min-entropy rate must approach the one in Section 7.

Restart data must be collected at normal operating conditions through the jent_hash_time() interface following the restart procedure specified in SP 800-90B (i.e., 1,000 samples from 1,000 restarts each) and processed by the NIST SP 800-90B entropy tool. The minimum of the row-wise and column-wise entropy rate must be more than half that of the raw noise entropy rate.

In addition, at least 1,000,000 outputs must be collected from the extractor (due to it being a non-vetted conditioning component) using the jent_read_entropy_qo_callback() interface. The estimated min-entropy rate of the 1,000,000 extractor outputs is a factor in determining its assessed min-entropy rate.

All vetted and unvetted conditioners underwent a code review and mathematical review to ensure compliance with SP 800-90B.

11 Vendor Permissions and Relationship

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