Userspace CPU Time Jitter RNG Entropy Source

SP 800-90B Public Use Document

Document Date: 25 Nov 2025
Document Version: 2

Prepared by

EWA-Canada
1223 Michael Street North, Suite 200
Ottawa, Ontario, K1J 7T2, Canada

On behalf of

CACI
44590 Guilford Dr.
Ashburn, VA, 20147, USA

Contents

1 Introduction 3
2 Security boundary 3
3 Operating Conditions 4
4 Configuration Settings 4
5 Physical Security Mechanisms 4
6 Conceptual Interfaces 5
7 Min-Entropy Rate 5
8 Health Tests 5
9 Maintenance 5
10 User Verification 5

1 Introduction

This public use document describes the design of the Userspace CPU Time Jitter RNG Entropy Source (referred to as the "entropy source" in this document) by CACI. This entropy source is an implementation of CPU Jitter Random Number Generator by Stephan Mueller, a non-physical entropy source which extracts noise from small externally unpredictable variances in the execution time of the same set of instructions that consists of the SHA3-256 conditioner and other supportive functions in the JENT library.

This entropy source was tested against the requirements prescribed within the Jan2018 publication of SP 800-90B and the 02Sep2025 publication of the FIPS 140-3 Implementation Guidance[1].

Table 1 below contains the tested oeprating environments which contain implementations of the entropy source.

Table 1: Tested Operating Environments

E-OE Name	OS	Processor	CAVP
Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)	Debian 12 on Red Hat RHEL v9.6 KVM	Intel Xeon Silver 4514Y (Raptor Cove)	A7546
Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)	Debian 12 on VMware ESXi v8.0	Intel Xeon Silver 4514Y (Raptor Cove)	A7546

2 Security boundary

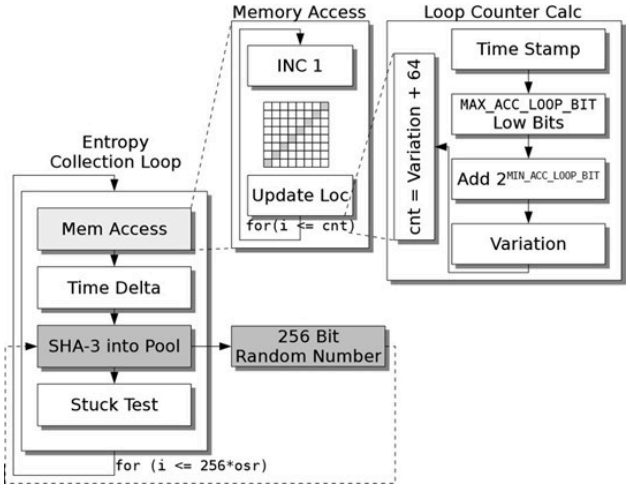

Figure 1: *Block Diagram* Of JENT

The entropy source gathers its entropy from the time deltas between memory access operations. The security boundary of the entropy source is the compiled binary of the entropy source of the C code that implements it as found in the specific version of JENT used and any memory allocated to it by the underlying operational environment. From a physical perspective, the security boundary is the processor that executes the compiled code and its operational environment.

The security boundary includes the Repetition Count Test (RCT) or the Adaptive Proportional Test (APT) health tests. If there is any failure any data gathered from the noise source is discarded without outputting any data and a failure code is returned to the caller.

[1]where the FIPS 140-3 IG D.J was last modified on November 5, 2021, and D.K was last modified on March 17, 2023

3 Operating Conditions

Because this is a non-physical noise source, the operating conditions are inherited from the operational environment which hosts the entropy source.

4 Configuration Settings

The configuration parameters of this entropy source are listed in Table 2 below.

Table 2: Parameter and Configuration Settings

Parameter/Configuration	Value
JENT Version	3.4.1
JENT_CONF_DISABLE_LOOP_SHUFFLE	Enabled
JENT_MEMORY_ACCESSLOOPS	128
JENT_MEMORY_BLOCKS	512
JENT_MEMORY_BLOCKSIZE	128
JENT_MIN_OSR	3
JENT_CONF_ENABLE_INTERNAL_TIMER	Defined
JENT_RANDOM_MEMACCESS	Defined
JENT_MEMORY_BITS	17

5 Physical Security Mechanisms

The noise source is non-physical. It will inherit the physical security mecanisms of the environment within which it is implemented.

6 Conceptual Interfaces

NIST defines three conceptual interfaces within SP 800-90B:

GetNoise An interface with which the consumer of the entropy source can get material from the noise source, without conditioning.

GetEntropy An interface with which the consumer of the entropy source can get entropy material from the entropy source.

HealthTest An interface with which the entropy source communicates information about the health of the noise source.

Table 3 below lists how the consumer of the entropy source can access these interfaces, and a description of the response format.

Table 3: Entropy Source Interfaces

Interface	Access	Response Format
GetNoise	Access to this interface is not available to the consumer of the implemented entropy source.	–
GetEntropy	Calling jent_read_entropy()	The function returns a 256-bit block of data from the SHA3-256 Vetted Conditioning Component.
HealthTest	A separate interface is not provided by the entropy source, when the entropy source fails it will either be unavailable or provide an error in response to jent_read_entropy() calls.	Implicit or function error

7 Min-Entropy Rate

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

8 Health Tests

SP 800-90B health tests are performed using the Adaptive Proportion Test (APT) and Repetition Count Test (RCT) and are run at startup and continuously under the normal operating conditions listed previously. Any errors detected are returned for handling by the caller. There are no known suspected noise source failure modes.

9 Maintenance

There are no maintenance requirements for this entropy source.

10 User Verification

The user can check the installation of the entropy source by comparing their device's components against the tested operating environment's OS, CPU, and by checking that the version of the JEnt service is 3.4.1.

During operation, the entropy source is responsible for its own performance and will fail if the entropy rate is not maintained.