KlasOS Jitter Entropy Source

SP 800-90B Non-Proprietary Public Use Document

Document Date: 5th February 2025

Document Version: 1.0

Prepared by

EWA-Canada

1223 Michael St North, Suite 200

Ottawa, Ontario, K1J 7T2, Canada

On behalf of

Klas Telecom, Inc

1101 30th Street NW, Suite 500

Washington, DC 20007

Table of Contents

1 Description

This public use document describes the design of the KlasOS Jitter Entropy Source by Klas (referred to as the 'entropy source' for the remainder of the document). The entropy source extracts entropy from a non-physical noise source in the form of a userspace CPU Jitter Random Number Generator (JEnt). The JEnt version is 3.4.1.

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

Table 1: Tested Operating Environments

E-OE Name	OS	Processor	CAVP Cert
KlasOS Keel 5.4.3 on Intel Xeon D-1746TER (VoyagerVM 4.0)	KlasOS Keel 5.4.3	Intel Xeon D-1746TER	A6552
KlasOS Keel 5.4.3 on AMD V2516 (VoyagerVMm 2.0)	KlasOS Keel 5.4.3	AMD V2516	A6552

Figure 1 - VoyagerVM 4.0 is an enterprise grade server for extreme edge environments, built on the latest Intel Xeon D processor.

Figure 2 - VoyagerVMm 2.0 is an ultra-slim compute module powered by the AMD V2516 processor.

Table 2 below contains the minimum amount of entropy material from this entropy source to compliantly seed a CAVP DRBG.

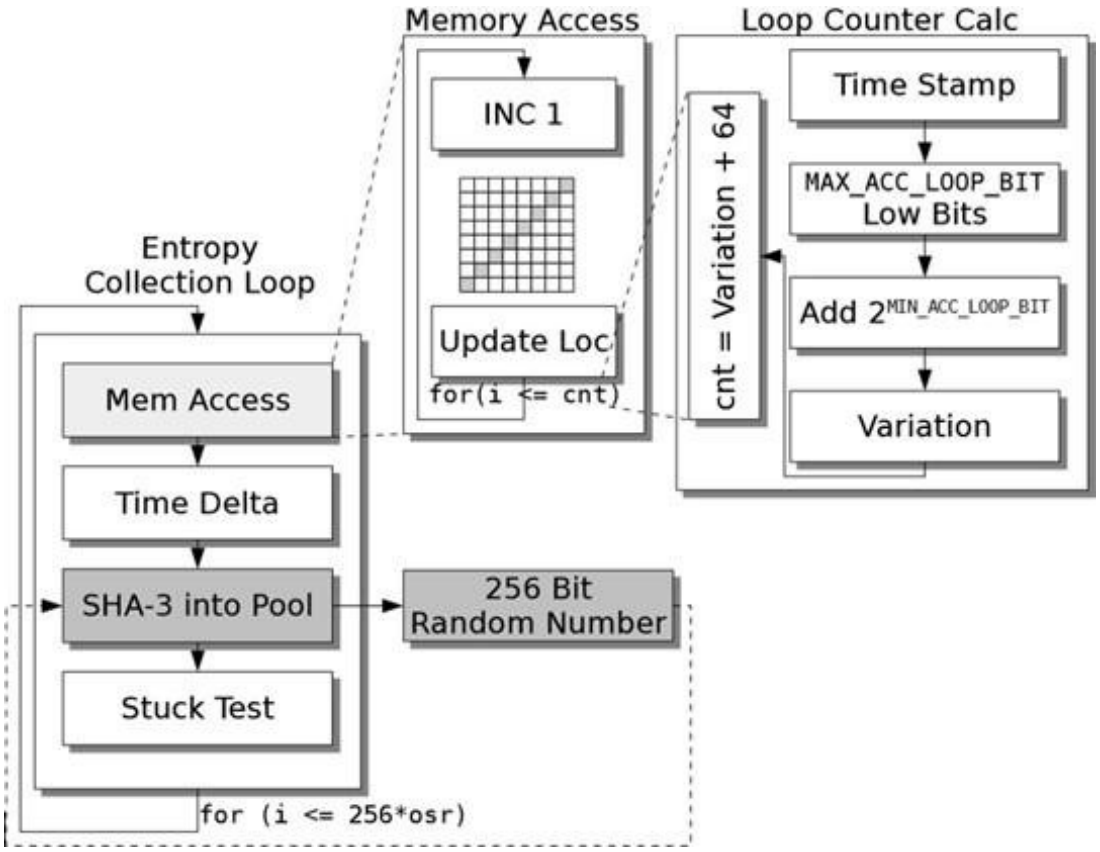
Table 2: DRBG Seeding Requirements

DRBG Security Strength	**Entropy Material Required**	
	with random nonce	*with fixed nonce*
128-bit	196-bits	128-bits
192-bit	288-bits	192-bits
256-bit	384-bits	256-bits

2 Security Boundary

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.



The figure above depicts a block diagram of the entropy source.

3 Operating Conditions

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

4 Configuration Settings

The entropy source's configuration settings are defined in Table 3 as follows.:

Table 3: Compliant Configuration Settings

Configuration item	Setting	Purpose
JENT_CONF_DISABLE_LOOP_SHUFFLE	Defined/enabled	Only execute one hash loop

JENT_MEMORY_ACCESSLOOPS	128	Memory access loops
JENT_MIN_OSR	3	Oversampling rate
JENT_CONF_ENABLE_INTERNAL_TIMER	Defined	hardware fine-grained timer is used
JENT_RANDOM_MEMACCESS	defined	Defines specifying the number of bits of the raw entropy data of the time delta value used for the APT.
JENT_MEMORY_BITS	17	Number of bits for mask

5 Physical Security Mechanisms

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

6 Conceptual Interfaces

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

1. GetNoise: An interface with which the consumer of the entropy source can get material from the noise source, without conditioning.
2. GetEntropy: An interface with which the consumer of the entropy source can get entropy material from the entropy source.
3. HealthTest: An interface with which the entropy source communicates information about the health of the noise source.

Table 4 below lists how the consumer of the entropy source can access these interfaces, and a description of the response format.

Table 4: Direct Entropy Source Interfaces

Interface	Access	Response format
Get Noise	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.

The entropy source additionally performs a stuck test and the lag predictor test.

9 Maintenance

There are no maintenance requirements for this entropy source.

10 Required Testing

No additional testing is required by the consumer.