SP 800-90B Non-Proprietary Public Use Document for Summit CPU Time Jitter RNG Entropy Source

Entropy Source Version 11.0

Document Version: 1.0

Document Date: 2024-01-22

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1 Description

The Summit CPU Time Jitter RNG Entropy Source v11.0 is a non-physical entropy source, part of the kernel binary, that feeds the kernel and user space 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 was tested on the operational environments listed in Table 1 below. The noise source was tested under the assumption that its output is non-IID.

Table 1: Operational environment and version.

Manufacturer	Model	Operating System	Processor
Laird Connectivity	WB50NBT MPU (Microprocessor Unit)	Summit Linux 11.0	Microchip/Atmel ATSAMA5D31, ARM Cortex A5-based (ARMv7)
Laird Connectivity	SU60-SOMC 60 Series SOM (System on Module)	Summit Linux 11.0	Microchip/Atmel ATSAMA5D36, ARM Cortex A5-based (ARMv7)
Laird Connectivity	WB45NBT MPU (Microprocessor Unit)	Summit Linux 11.0	Microchip/Atmel AT91SAM9G, ARM9-based (ARMv5)

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.

The noise source is implemented by collecting and accumulating 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 time variances, in the form of time deltas, are collected and conditioned to 256-bits by the SHA3-256 vetted conditioning function. The output of this conditioning function is 256 bits of full entropy.

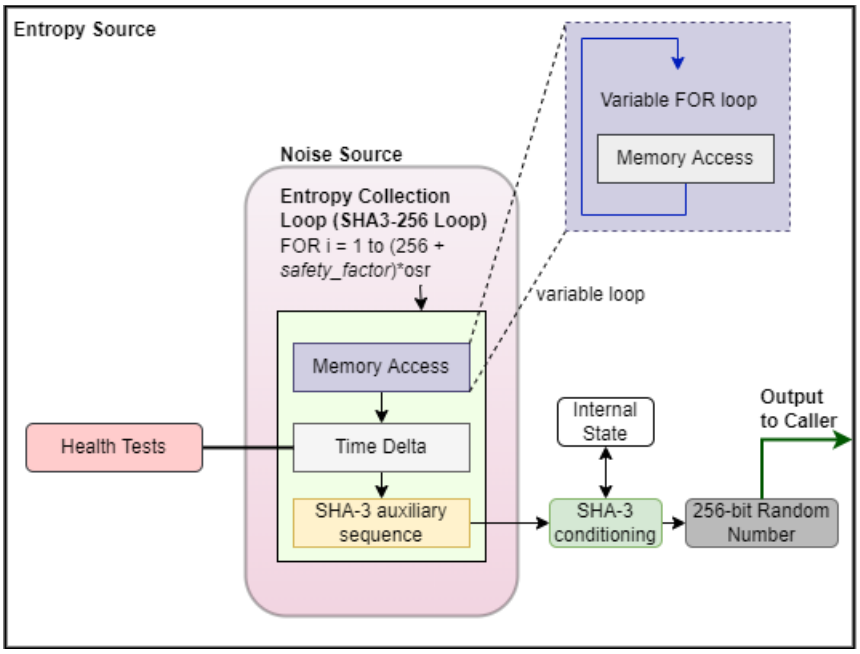

Figure 1: Entropy Source Design.

3 Operating Conditions

The noise source is non-physical, and thus the operating conditions are inherited from the operational environment. The tested operational environments and their operating conditions are shown in Table 2.

Table 2: Operating Conditions for each Operational Environment.

Manufacturer/Model	Temperature Range	Voltage Range	Humidity Range
Laird Connectivity/WB50NBT	-30°C – 85°C	3.3V±10% 1.8V±10%	10% - 90%
Laird Connectivity/SU60-SOMC 60	-30°C – 85°C	3.3V±5%	10% - 90%
Laird Connectivity/WB45NBT	-20°C – 70°C	3.3V±5% 1.8V-2%+5%	10% - 90%

4 Configuration Settings

No configuration settings are required.

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.
- jent_kcapi_random(): 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. This interface corresponds to the GetEntropy() conceptual interface from SP 800-90B.
- 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 noise source provides an entropy rate for each time delta of $H_{submitter}$ = 1 bit per 64-bit sample.

The entropy source collects 320 time deltas of 64 bits each from the noise source as input to the SHA3-256 vetted conditioning component. This input corresponds to at least 320 bits of entropy. The output entropy rate of the SHA3-256 vetted conditioning component, and the output of the entropy source, is assessed to be 1 bit/bit.

The input to the conditioning component considers and oversampling rate (osr) of 1 and a safety factor of 64 bits of entropy (so that the conditioning component may output full entropy). The oversampling rate and safety factor are fixed values defined in the implementation and cannot be altered.

8 Health Tests

The entropy source implements the following categories of health tests:

- Startup health tests.
- Continuous health tests.
- On-demand health tests.

The Continuous health tests implement a Repetition Count Test (RCT), an Adaptive Proportion Test (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 $\alpha_i = 2^{-30}$, which lies within the recommended range of $2^{-20} \leq \alpha \leq 2^{-40}$. 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. The only exception to this rule is during the start-up tests, where intermittent failures will be treated as permanent.

Permanent failures are 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 Startup Health Tests

Start-up tests conduct the same set and parameters of the continuous health tests on 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 the permanent shutdown of the entropy source.

8.2 Continuous Health Tests

8.2.1 Repetition Count Test

The Repetition Count Test conforms to SP 800-90B section 4.4.1.

- $H = 1$ bit of entropy per 64-bit time delta/.
- Intermittent failure alpha value $\alpha_i = 2^{-30}$.
- Permanent failure alpha value $\alpha_p = 2^{-60}$.
- Intermittent failure cutoff value $C_i = 31$.
- Permanent failure cutoff value $C_p = 61$.

8.2.2 Adaption Proportion Test

The Adaptive Proportion test conforms to SP 800-90B section 4.4.2.

- $W = 512$.
- $H = 1$ bit of entropy per 64-bit time delta.
- Intermittent failure alpha value $\alpha_i = 2^{-30}$.
- Permanent failure alpha value $\alpha_p = 2^{-60}$.
- Intermittent failure cutoff value $C_i = 325$.
- Permanent failure cutoff value $C_p = 355$.

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 conditioning function, and another sample needs to be fetched. 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.

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 $\alpha_i = 2^{-30}$, which lies within the recommended range of $2^{-20} \leq \alpha \leq 2^{-40}$. 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. The only exception to this rule is during the start-up tests, where intermittent failures will be treated as permanent.

Permanent failures are 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.

8.3 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. Similar to the start-up tests, the data used for the on-demand health tests are discarded after successful completion.

8.4 Error Codes

When entropy is requested from through the jent_kcapi_random() 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 failureThe following error codes are defined for jent_read_entropy():

- -1 entropy_collector is NULL or the generation failed.
- -2 Intermittent health failure.
- -3 Permanent health failure.

9 Maintenance

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

10 Required Testing

To test the entropy source, raw data samples must be collected using both timer options using a test harness that is capable of accessing the jent_measure_jitter() noise interface from the entropy source. 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_measure_jitter() 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.