

SP 800-90B Non-Proprietary Public Use Document

Firebox Cryptographic Jitter Entropy Module

By WatchGuard Technologies, Inc.

255 S. King Street, Suite 1100

Seattle, WA 98104, United States

Document Version 1.3

July 9, 2024

Revision History

Version	Change
1.0	Initial draft.
1.1	Minor updates.
1.2	Updates after addressing CMVP 1st comments
1.3	Updates after addressing CMVP 2nd round comments

Table of Contents

1	Description	4
2	Security Boundary	4
3	Operating Conditions	5
4	Configuration Settings	6
5	Physical Security Mechanisms	6
6	Conceptual Interfaces	6
7	Min-Entropy Rate	7
8	Health Tests	7
9	Maintenance	7
10	Required Testing	7

1 Description

The Firebox Cryptographic Jitter Entropy Module v12.11 implements CPU Jitter version 3.4.0 without any modifications. CPU Jitter Entropy Source is a non-physical entropy source. It makes no IID claim and thus meets all the requirements for non-IID compliance. Table 1 summarizes the WatchGuard Firebox platforms and their respective processors for which testing was performed.

2 Security Boundary

The boundary of the CPU Jitter implementation is shown in Figure 1. CPU Jitter gets its entropy from the time deltas between repeated hash/memory collection operations. The CPU Jitter output is used to seed an SP 800-90A compliant DRBG, which is outside of the boundary of the entropy source. The security boundary of the CPU Jitter implementation contains the source code files provided as part of the software delivery. The source code contains API calls which are used by modules requesting entropy from CPU Jitter.

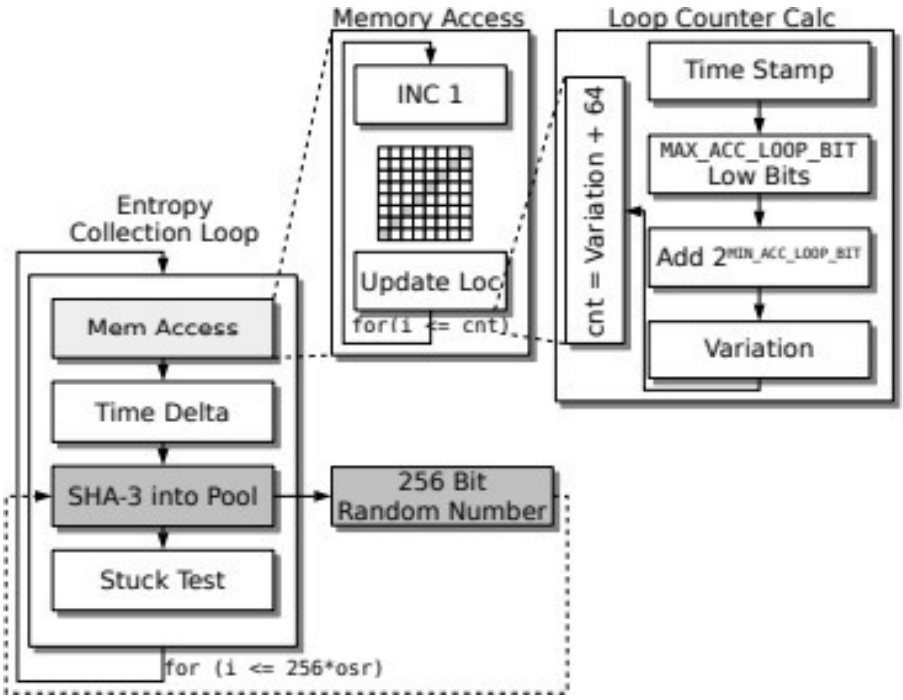


Figure 1 - Entropy Source Diagram

3 Operating Conditions

Table 1 summarizes the operating conditions for each of the tested platforms.

Platform	Processor	Clock Speed	Cache	Platform Temperature	Platform Voltage (AC)
NV5	NXP LS1012A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T20	NXP LS1023A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T20-W	NXP LS1023A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T25	NXP LS1043A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T25-W	NXP LS1043A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T40	NXP LS1043A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T40-W	NXP LS1043A	1.0GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T45	NXP LS1043A	1.6GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T45-PoE	NXP LS1043A	1.6GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T45-W-PoE	NXP LS1043A	1.6GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T45-CW	NXP LS1043A	1.6GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
T80	NXP LS1046A	1.2GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
T85-PoE	NXP LS1046A	1.8GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
M290	NXP LS1046A	1.8GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
M390	NXP LS2084A	2.0GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
M590	NXP LX2120A	1.8GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
M690	NXP LX2160A	2.2GHz	L1 = 32KB (data) L1 = 48KB (instruction)	0oC – 40oC	90V-264V
M4800	Intel E-2176G	3.7GHz	L1 = 32KB (data) L1 = 32KB (instruction)	0oC – 40oC	100V-240V
M5800	Intel 6230	2.1GHz	L1 = 64KB	0oC – 40oC	100V-240V

Table 1 – Operating Environment and Conditions

4 Configuration Settings

Table 2 summarizes the configuration settings used by the WatchGuard Firebox Cryptographic Jitter Entropy Module implementation of CPU Jitter. These settings must be preserved to comply with the CPU Jitter ESV certificate.

Parameter	Value	Description
JENT_RANDOM_MEMACCESS	Enabled	Enables random memory access using bit versus blocks
JENT_MEMORY_BITS	17	Number of bits in a memory block (i.e., the memory block size in bits is 2^JENT_MEMORY_BITS)
JENT_MEMORY_SIZE	131072	Memory block size
JENT_MEMORY_ACCESSLOOPS	128	Memory access loops
JENT_MIN_OSR	3	Set in code by JENT_CONF_DISABLE_LOOP_SHUFFLE (Enabled: 3 (default), Disabled: 1)
JENT_CONF_DISABLE_LOOP_SHUFFLE	Enabled	Disables pseudo-random looping e.g.,lower boundary entropy equals upper boundary entropy.
JENT_FORCE_FIPS	Enabled	Fips mode flag is set
ENTROPY_SAFETY_FACTOR	64	Entropy safety factor

Table 2 - Configuration Settings

5 Physical Security Mechanisms

The Firebox Cryptographic Jitter Entropy Module v12.11 is enclosed entirely within the module's cryptographic boundary which is protected by tamper evident seals, anti-probing barriers and electronic tamperdetection and is assessed for FIPS 140-3 level 2 Physical Security compliance.

6 Conceptual Interfaces

The entropy source provides the following interfaces:

- jent_read_entropy(): Obtains conditioned entropy for the caller. This is the main function from the entropy source, the one that shall be used to request entropy data. The entropy gathering logic creates 256 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

FireBox Cryptographic Jitter Entropy Module generates an output that is considered to have full entropy. A request for 256 bits of entropy results in 256 bits of entropy per output sample, or full entropy.

8 Health Tests

Per the SP 800-90B requirements, health tests are run when CPU Jitter starts, and are then run continuously while it is operating. All tests check for persistent failures based on their respective "cutoff" values, which represent expected error thresholds.

CPU Jitter implements four types of health tests:

- Stuck Test
- Repetition Count Test (RCT)
- Adaptive Proportion Test (APT)
- Lag Predictor Test

All health test failures are considered permanent failures. If one is triggered, the current instance of the CPU Jitter will always remain in error state. The documentation of the API call jent_read_entropy(3) explains that the caller can only clear this error state by deallocating the CPU Jitter instance followed by an allocation of a new CPU Jitter Entropy Source instance to reset the noise source.

9 Maintenance

There are no maintenance requirements for FireBox Cryptographic Jitter Entropy Module.

10 Required Testing

The entropy report contains the results of the testing Intel did on their raw and restart data. Raw data was collected by running the invoke_testing.sh script included in the CPU Jitter package. This script gathers 1,000,000 samples of 8-bits of raw time stamp data. It also gathers 1000 samples of 8-bits of rawdata after restarting CPU Jitter each time, thus eventually gathering 1,000,000 samples of raw data for the restart tests.

Raw Data Analysis

Each raw data sample consists of one time stamp, which is 64 bits long. The CPU Jitter design states that the source can deliver full entropy if and only if the min-entropy is at least 1/osr (osr=3) bits of entropy per time stamp. This CPU Jitter implementation uses an oversampling rate of 3.

Restart Data Analysis

The CPU Jitter Entropy does not have a stochastic model, but an $H_{submitter}$ estimate of 1 was determined using analysis as described in SP 800-90B Section 3.2.2, Requirement 3. The CPU Jitter Entropy Source implementation uses an oversampling rate of 3. For that reason, the initial entropy estimate is set to 0.333 (1/3) for the purpose of running the NIST restart tests.

For the restart tests, the raw entropy data is collected for 1,000 CPU Jitter Entropy Source instances allocated sequentially. That means, for one collection of raw entropy, one CPU Jitter Entropy Source

instance is allocated. After the conclusion of the data gathering it is deallocated and a new CPU Jitter Entropy Source instance is allocated for the next restart test round.