Jitter RNG

Stephan Müller <smueller@chronox.de>

Agenda

- Noise Source Phenomenon
- Architecture of Jitter RNG
- Conditioner
- Standards Compliance
- Documentation
- Testing Interfaces

CC BY 4.0

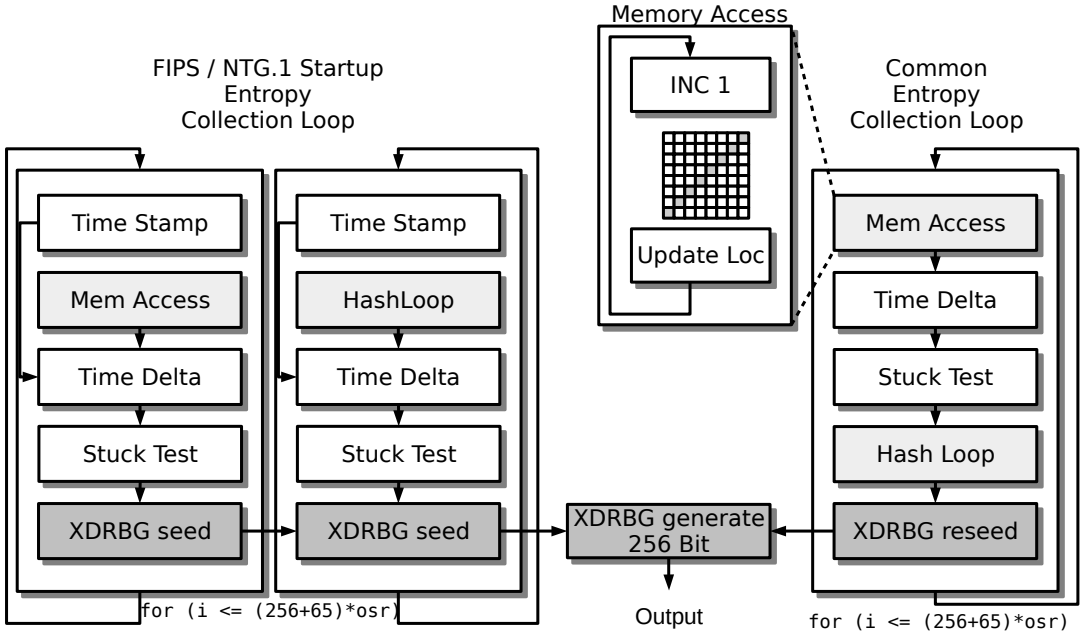
Noise Source Phenomenon

- **Execution time of instructions not deterministic**
  - The more complex a CPU is the larger the non-determinism
  - Non-determinism based on wait states between CPU components → CPU state defines the non-determinism
  - Serializing instructions (e.g. cpuid) may reset CPU state → timing of first instruction afterwards is deterministic
- **Memory access time is not deterministic**
  - The slower the memory compared to CPU, the larger the variations → wait states define the non-determinism
  - Variations(L1-access) < Variations(L2-access) < Variations(L3-access)

CC BY 4.0

Architecture of Jitter RNG

- **Measure execution time:**
  - Memory Access Loop
  - Hash Loop
- **Time delta subject to health test**
- **Time delta added to conditioner**
- **Random output block:**
  - Conditioner output
  - Always 256 bits
  - Input entropy (256+65)*OSR time deltas
- **Heuristic Entropy Rate:**
  - 1/OSR bits of entropy per time delta
  - Oversampling of 65 bits to offset entropy loss by conditioning components



CC BY 4.0

Architecture: Noise Source Sampling

- **Initialization:**
  - 3.7.0: FIPS 140 / NTG.1 mode
    - Memory / Hash loops sampled independently
    - Each loop type must deliver (256 + 65) * OSR time deltas
    - Each loop type operates on independent health test state
    - Loops are processed serially
  - < 3.7.0: see runtime
- **Runtime**
  - Memory / Hash loops sampled concurrently
  - Concurrent loops must deliver (256 + 65) * OSR time deltas
  - Common health test

CC BY 4.0

Conditioner

- **Intermediate[RATE] = timedelta||domain sep||digest**
  - Exactly one Keccak operation per intermediate buffer injection into conditioner → monomodal behavor
- **3.7.0: XDRBG**
  - V_t = SHAKE-256(512, V_{t-1}||Intermediate1||…||Intermediate963)
  - (V_{t+1}, S) = SHAKE-256(768, V_t)
  - Output = S of 256 bits
- **< 3.7.0: SHA3-256**
  - S_n = SHA3-256(S_{n-1}||Intermediate1||…||Intermediate963)
  - Output = S_n of 256 bits

CC BY 4.0

Standards Compliance

- **FIPS 140-3:**
  - CAVP testing of conditioner
- **SP800-90B: compliant entropy source**
- **SP800-90C: combination with DRBG may lead to RBG2(NP)**
- **AIS20/31 v3.0:**
  - 3.7.0: NTG.1
  - < 3.7.0: no compliance

CC BY 4.0

Guidance: Entropy Rate Configurations

- **Oversampling Rate: Global heuristic entropy rate is 1/OSR**
  - Runtime: osr parameter during initialization
  - Compile time: -DJENT_MIN_OSR=<VALUE>
  - Default: 3
  - Note: Runtime configuration value takes precedence over compile-time value
- **Memory Access Buffer: The larger the buffer, the higher the entropy rate.**
  - Runtime: JENT_MAX_MEMSIZE_*
  - Compile time: -DJENT_DEFAULT_MEMORY_BITS=<VALUE> (memory size is $2^{\text{JENT\_DEFAULT\_MEMORY\_BITS}}$)
  - Default: 18 (resulting in 2^{18} bytes)
  - Note: Runtime configuration value takes precedence over compile-time value and L1-cache value
  - Note: Compile-time value applied only if no L1-cache size is detected
- **Hash Loop Count: The larger the iteration count, the higher the entropy rate.**
  - Runtime: JENT_HASHLOOP_*
  - Compile time: -DJENT_HASH_LOOP_DEFAULT=<VALUE>
  - Default: 1
  - Note: Runtime configuration value takes precedence over compile-time value

CC BY 4.0

SP800-90B Guidance

- **Jitter RNG configuration:**
  - JENT_NTG1 may be set
  - JENT_FORCE_FIPS must be set or base OS is in FIPS mode
  - Status must show:
    - FIPS mode enabled
- **Test Evidence**
  - Apply heuristic analysis mandated by NIST on common behavior
  - Ignore NTG.1 startup procedure of independent hash/memory loop
  - Obtain CAVP certificate for SHAKE-256 conditioner (use ACVP-Parser at https://github.com/smuellerDD/acvpparser)

CC BY 4.0

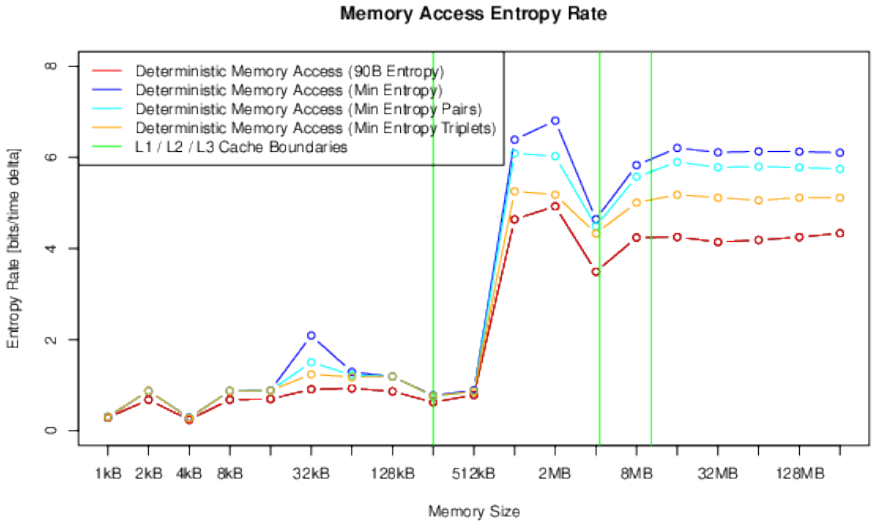
NTG.1 Guidance

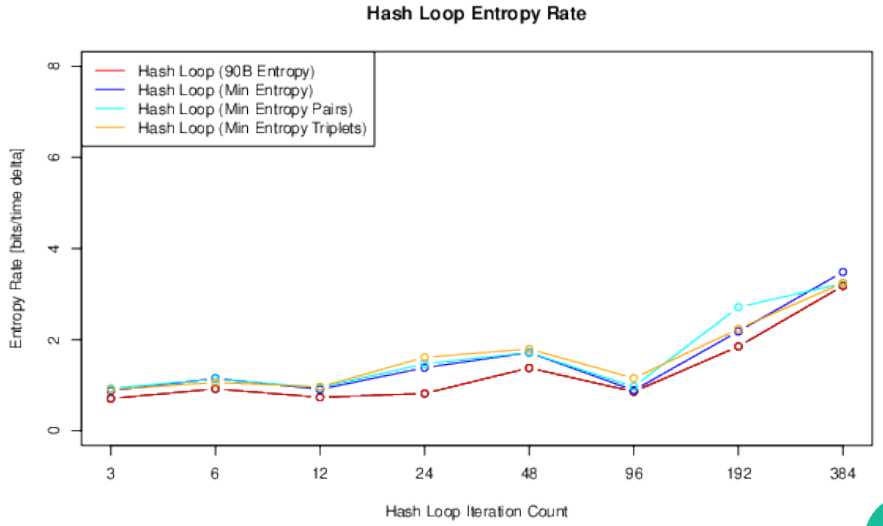
- **Jitter RNG configuration**
  - Memory Access Loop: Memory buffer size at least 4 times larger than L1 cache size
  - JENT_NTG1 flag must be set (this flag automatically disables the internal timer)
  - Status must show:
    - AIS 20/31 NTG.1 mode enabled
    - Memory Block Size equal or larger than four times L1 cache
    - Secure Memory enabled
    - Internal timer disabled
- **Test Evidence**
  - Measured entropy rate must show rate 8/OSR or higher
    - Hash loop
    - Memory access loop
    - Common behavior (SP800-90B restart + runtime tests)

CC BY 4.0

Entropy Rate Analysis

- What to do when required entropy rate is not achieved?
- Adjust the OSR to a higher value to lower targeted entropy rate.
- Test tools to independently measure memory access and hash loops → determine applicable memory sizes, hash loop iteration counts for targeted entropy rate.





CC BY 4.0

Documentation

- https://chronox.de/jent
- **General description**
- **Explanation of raw noise phenomenon**
  - Testing on 250+ different systems of all big CPU types
  - X86 Intel/AMD, ARM 32/64 bit, RISC-V, Sparc, IBM P, IBM Z, MIPS
- **SP800-90B compliance: section 6.1**
- **NTG.1 compliance: section 6.6**
  - 9/4 tuples provided with XDRBG specification: https://leancrypto.org/papers/xdrbg.pdf

CC BY 4.0

Testing Interfaces

- Raw Entropy Gathering tools for kernel/user space in Jitter RNG library code base
- Tools to obtain SP800-90B statistical min-entropy values
- Kernel Space: Requires enabling of kernel test interface support

CC BY 4.0

Test Results Interpretation

- **Heuristic entropy rate: 1/OSR bits of entropy per time delta**
- **Statistical entropy rate: SP800-90B tools**
- **Check:**

  min(statistical entropy rates) > heuristic entropy rate
- **Test to be conducted for each timer**
  - On each CPU
  - On each different timer on one CPU, if applicable
  - Internal timer, if applicable
- **BSI methodology: Apply safety factor of 8**

  min(statistical entropy rates) > 8 * heuristic entropy rate

CC BY 4.0

Questions?
Thank you for listening.

CC BY 4.0