Non-Physical Entropy Sources

RBG Workshop
01 June 2023

Chris Celi (NIST)
Tim Hall (NIST)
Outline

- Physical vs. Non-Physical
- Entropy Justification
- Health Tests
- Conditioning Components
### Physical versus Non-Physical

<table>
<thead>
<tr>
<th>Physical</th>
<th>Non-Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Depend on some natural phenomena, like thermal noise, quantum shot noise, ...</td>
<td>• Depend on timings available within a complex system</td>
</tr>
<tr>
<td>• <strong>25 Entropy Validations</strong></td>
<td>• <strong>20 Entropy Validations</strong></td>
</tr>
<tr>
<td>• Ring oscillators, meta-stable latches, ...</td>
<td>• 19 are CPU Jitter sources</td>
</tr>
</tbody>
</table>
Non-Physical Noise Sources

• Time it takes to process an interrupt request
  • E15, Apple CoreCrypto v11.1

• Time it takes to perform a complex operation
  • E1, NetApp CPU Jitter v3.4.0

• Race conditions between multiple threads

• The sample size may be arbitrary in relation to the noise source
• When using a clock to "capture" entropy, often this is done with time deltas rather than raw time stamps

• A delta may only contain 16 bits of information, easy to see any entropy is likely in the low order bits
  - But not always! Some lower precision system clocks may fix the least significant bits!

• Very dependent on the underlying hardware and operating system

```plaintext
clock[i] = current_time
complex operation
clock[i+1] = current_time
delta = clock[i+1] - clock[i]
```
• Software settings affect the entropy source...
  • Compilers
  • Configuration flags
  • Optimizations
  • Other processes handled by the OS

• Hardware settings affect the entropy source...
  • Clock speed
  • Cache sizes

• A validated non-physical entropy source must use specific configurations!
CPU Jitter

- Relies on memory access timings and unknown wait states of the memory buffer
- Uses cache misses by overflowing L1 cache, as L2 cache readings have more variance

https://www.chronox.de/jent/doc/CPU-Jitter-NPTRNG.pdf
• Difficult to claim large amounts of entropy per sample
• Typical estimates range from 0.33 bits to 1 bit per 64-bit sample
  • Remember a 64-bit sample may not contain 64 varying bits per sample
• Heuristics tend to make a claim that any entropy exists
• Joshua Hill – “What To Expect When You’re Expecting (to Evaluate JEnt Against SP 800-90B)”
  • https://www.untruth.org/~josh/sp80090b/What%20To%20Expect%20When%20You're%20Expecting%20(to%20Evaluate%20JEnt%20Against%20SP%20800-90B)%2020210904-1.pdf
  • For a presentation of a similar talk: CMUF Entropy Working Group-20221018 1702-1
• Base assumption for CPU Jitter is 1 bit of entropy per sample
  • [https://www.chronox.de/jent/doc/CPU-Jitter-NPTRNG.pdf](https://www.chronox.de/jent/doc/CPU-Jitter-NPTRNG.pdf)

• Reliance then becomes justifying that the hardware and software configurations allow the entropy source to make that claim

• Provide clock speed, cache sizes, amount of memory used in accesses, compiler flags, JEnt configuration parameters...
Health Tests – Failure Modes

- Failure modes are difficult for non-physical sources

- A similar operation is being run millions of times to gather entropy...
  - Periodicity?
  - Loss of entropy over time?

- Does the source behave well on idle or busy systems?

- Is this for a virtual environment where specific hardware isn’t a guarantee?
Because sources are software-based, there is a lot of freedom to add developer-defined health tests to mitigate concerns.

RCT and APT do not deal with periodicity
- Repetition Count Test and Adaptive Proportion Test are defined in SP 800-90B

CPU Jitter v3.4.0 adds a Lag Predictor Test that uses the 90B estimator to address potential periodicity
Conditioning Components

• CPU Jitter times the full operation of the noise source and conditioner

• CPU Jitter v2 – LFSR on a primitive polynomial
  • 64-bit sample processed one at a time, 64 times
  • Non-bijective, despite the primitive polynomial being injective and surjective
  • Nin = 4096, NOut = 64

• CPU Jitter v3 – SHA-3-256
  • Much easier claim on full-entropy
  • 64-bit sample processed one at a time, 256+ times
  • Still considered vetted with a loop around SHA-3
In Linux systems, an entropy source might be shared between kernel space and user space.

If CPU Jitter is used in kernel space with a DRBG, the entropy source for user space will chain the DRBG.

Considered two separate entropy sources for certification due to the differences in the conditioning component chain.
Questions?