Tailored Health Tests for Physical Entropy Sources

John Kelsey [1] [2]   Johannes Mittmann [3]

[1]National Institute of Standards and Technology (NIST), United States

[2]COSIC, KU Leuven, Belgium

[3]Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany

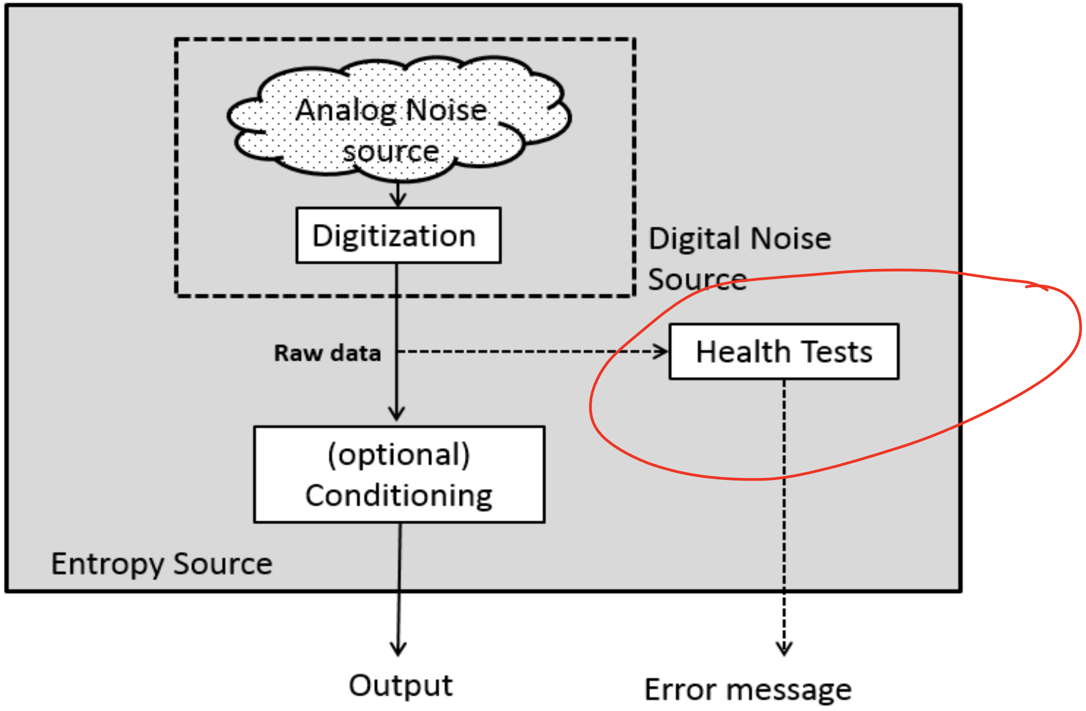
International Cryptographic Module Conference (ICMC23)
September 20, 2023

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Bundesamt
für Sicherheit in der
Informationstechnik

What are health tests?

Components of an entropy source



- Noise source = source of nondeterministic bits
- Conditioning = optional postprocessing of outputs
- Health tests = mandatory tests to detect failures/defects

How can an entropy source fail?

Total failures:

- Entropy very low
- Total loss of security
- Must be detected immediately

Non-tolerable entropy defects:

- Entropy below entropy claim
- Security guarantees violated
- Must be detected sufficiently soon

Health test requirements of SP 800-90B

- Continuous tests
  - Running all the time
  - Intended to detect a loss of entropy
  - Designer should test for any known failure modes
  - Two NIST provided tests
  - Use ours or show you own tests detect same problems
- Start-up tests
- On-demand tests
  - Both CAN just be running continuous tests over lots of data
  - Ideally device will do more

Health test requirements of AIS 20/31 for physical TRNGs

- Start-up test
  - Shall detect total failures and severe statistical weaknesses at start-up
  - Must pass before random numbers are output
- Total-failure test
  - Shall detect total breakdown of the noise source while the TRNG is in operation
  - Must react immediately
- Online test
  - Shall detect non-tolerable entropy defects while the TRNG is in operation
  - Must react sufficiently soon
  - Shall be tailored to the stochastic model

Why do entropy sources need health tests?

- Noise sources are fragile
  - Previously secure device can fail (aging, damage, environment)
  - A few devices may be insecure because of process variation
- Failures are invisible
  - Entropy source bits go into DRBG
  - Everything works fine....
  - ...it's just insecure
- Health tests are the only way to detect these failures!

A useful framework

A useful framework

failure $\rightarrow$ signal $\rightarrow$ detection $\rightarrow$ reaction

- Failure or Defect: Entropy estimate no longer valid
- Signal: Observable change to some measurement
- Detection: Health test decides there is a problem
- Reaction: Module does something to prevent loss of security

Failure/defect: Entropy estimate no longer valid

failure $\rightarrow$ signal $\rightarrow$ detection $\rightarrow$ reaction

- Total failure = catastrophic loss of entropy
  - Examples: oscillator locks to clock, Geiger counter fails
- Defect (Non-tolerable entropy defect) = More subtle failure
  - Parameters out of range for entropy estimate
  - Assumption of entropy estimate falsified
  - Environmental conditions outside operating envelope
  - Attacks

Signal: change in observed measurements

failure $\rightarrow$ **signal** $\rightarrow$ detection $\rightarrow$ reaction

May be a change in statistics of:

- Raw output bits (raw random numbers)
- Internal values
- Conditioned outputs (internal random numbers)
- Environmental sensor

Need to show that failure will lead to signal!

Detection: Deciding there is a problem

failure $\rightarrow$ signal $\rightarrow$ **detection** $\rightarrow$ reaction

- Sometimes, signal unambiguous
  - Example: output stuck on very long run of zeros
- Usually involves choosing a cutoff value for some statistic
- ...and accepting a false positive/false negative tradeoff

False alarms vs silent failures

failure → signal → **detection** → reaction

- False alarm (false positive):
  - Entropy source operating correctly
  - Alarm raised
  - Consequence: Product reliability / usability problem
  - Especially bad for continuous tests
- Silent failure (false negative):
  - Entropy source producing less entropy than claimed
  - No alarm raised
  - Consequence: Security problem

Problem: Very low false positives $\rightarrow$ low power

failure $\rightarrow$ signal $\rightarrow$ **detection** $\rightarrow$ reaction

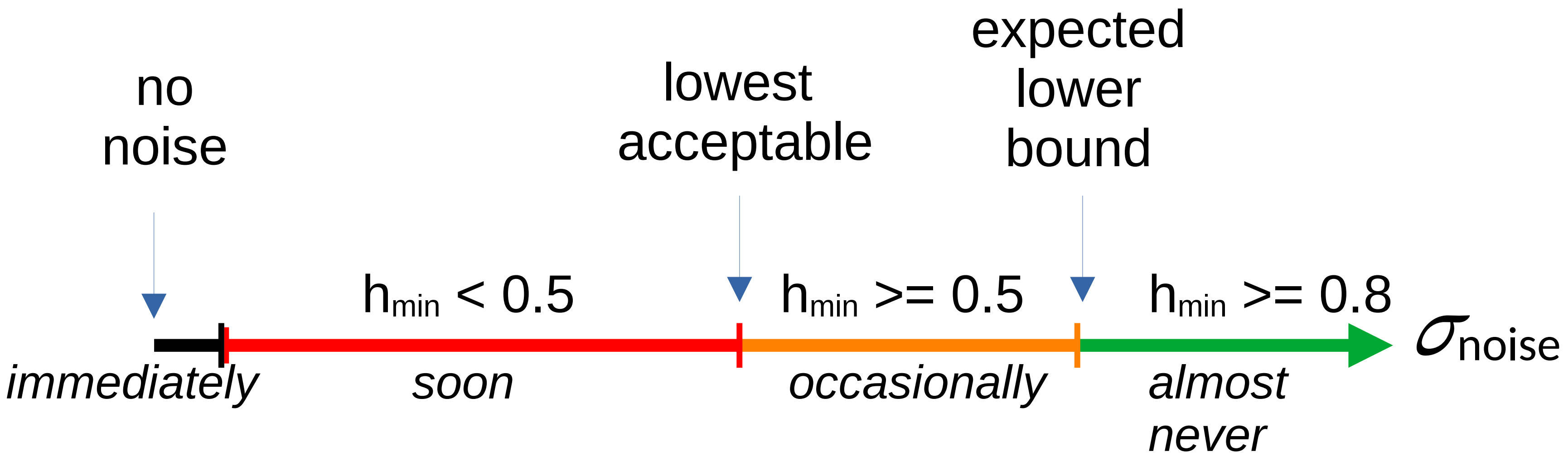
- Suppose we set test cutoff so high that false positives extremely rare

  Result: test can't detect much
- Some tests designed only to detect catastrophic failure
  - Example: Repetition count test
- ...but usually, we want to detect more subtle problems!

Strategy: Under promise, over deliver

Example: Source promises 0.5 bits h_{min} per bit

- Design source to deliver $h_{\min}$ 0.8
- If source working as expected, *almost no false alarms*
- If source drops below $h_{\min} = 0.5$, *test will fail soon*

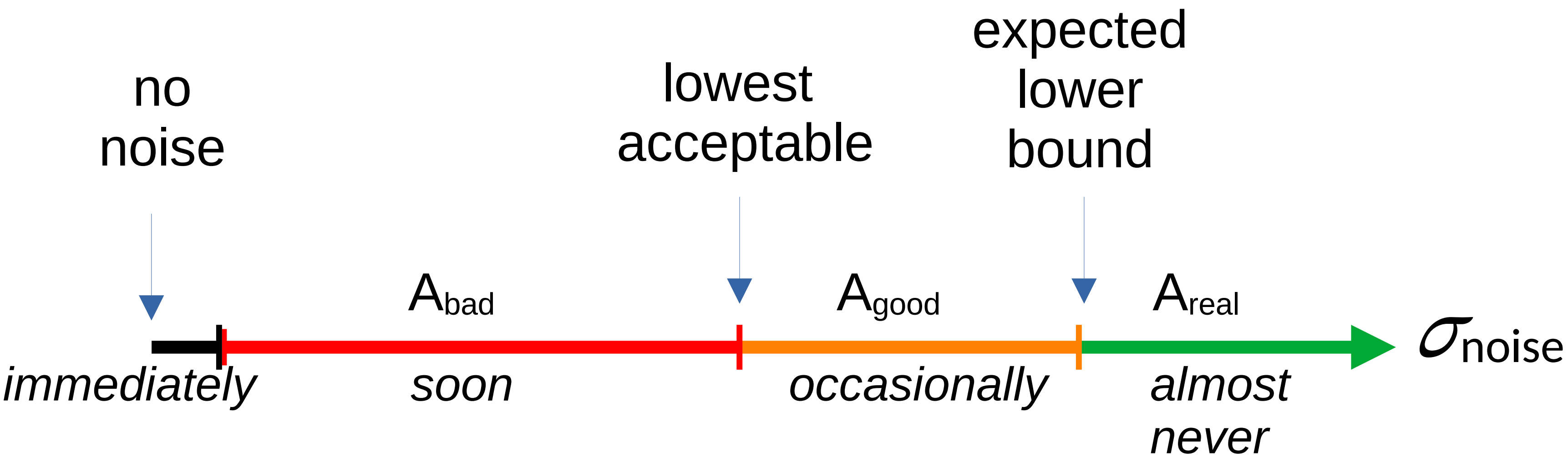


When should health tests fail?

Stochastic model parameters

Tests often estimate parameters of stochastic model.

- A_{real} – parameters that guarantee h_{min} 0.8
- A_{good} – parameters that guarantee h_{min} 0.5
- A_{bad} – parameters that don't guarantee h_{min} 0.5



When should health tests fail?

Reaction: What happens when test triggers?

failure $\rightarrow$ signal $\rightarrow$ detection $\rightarrow$ **reaction**

- Obvious answer:
  - Signal error condition
  - Shut down device
- Often makes sense for total failure tests
- Plays well with "under promise, over deliver" strategy

- There are some alternatives....
- ...but if tests keep failing, eventually source must signal failure and shut down!

Reaction: Alternatives to immediate shutdown

failure $\rightarrow$ signal $\rightarrow$ detection $\rightarrow$ **reaction**

- Stop output and run additional tests
- Suppress output until test stops failing
- Don't count outputs as having entropy
- Keep track of failed tests: too many $\rightarrow$ fatal error
  - E.g., window of last 64 tests run
  - Too many fails $\rightarrow$ shut it down
- Alter some internal parameters
  - E.g., slow down sampling

Must be some point at which signal error and stop entropy source

Example: Radioactive decay

Example: Radioactive decay

Loosely based on Alkassar et al. (ICCSA 2005) and Park et al. (ICMC19, TVLSI 2020)

- The noise source contains 0.3 µg of Americium-241 with total activity 37 kBq.
- A photodiode is used to detect emitted alpha particles.
- The photodiode detects 40% of the emitted particles and has negligible dead time.
- Detected alpha particles are counted in time intervals of length s.

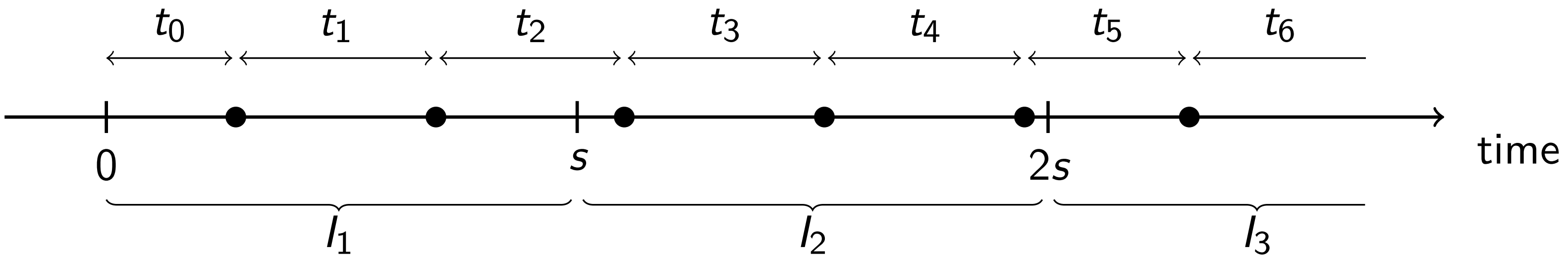


Figure: Radioactive decays (•) occurring in time intervals (⊢⊣)

Noise source

- Detected alpha particles are counted in time intervals I_n of length s.
- Raw random numbers: $r_n := \#\{\text{detected particles in } I_n\}$
- Internal random numbers: $y_n := r_n \bmod 2$

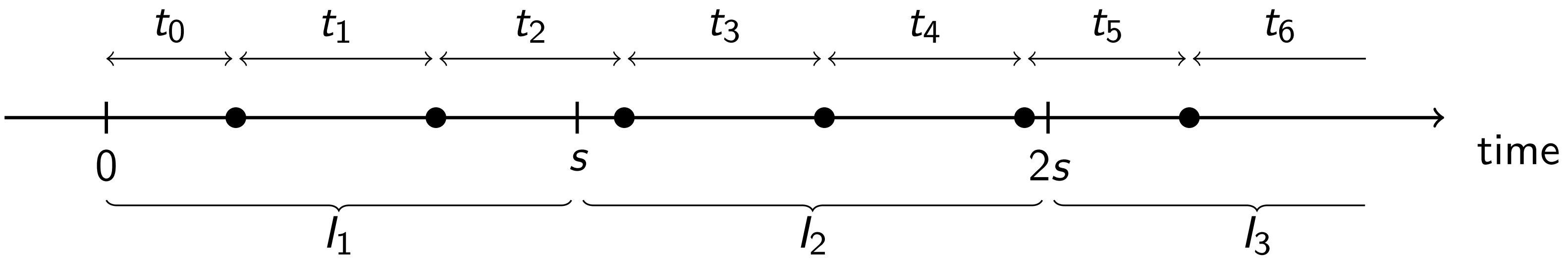


Figure: Radioactive decays (•) occurring in time intervals (⊢⊣)

Stochastic model

- Time t_0 and intermediate times $t_1, t_2, \ldots$ between detected alpha particles are interpreted as realizations of iid Exp( )-distributed $T_0, T_1, T_2, \ldots$ with rate

$$0.4 \cdot 37\,000\,\mathrm{s}^{\;1} = 14\,800\,\mathrm{s}^{\;1}\,.$$

- The raw random numbers $R_1, R_2, \ldots$ are iid Pois(s)-distributed with

$$\mathrm{E}(R_n) = \mathrm{Var}(R_n) = \;\; s\,.$$

- The parameter of this model can be taken as s.

Entropy of the internal random numbers

- The internal random numbers $Y_n = R_n \bmod 2$ are iid $B(1, p)$-distributed with

$$p = \Pr(Y_n = 1) = \quad_{i=0} \Pr(R_n = 2i + 1) = \frac{1}{2} \quad \frac{e^{\ 2\ \ s}}{2} .$$

- The min-entropy per internal random bit is given by

$$\mathrm{H} \quad (Y_n) = \quad \log_2 \max_{y_n \quad \{0,1\}} \Pr(Y_n = y_n) = 1 \quad \log_2 \ 1 + e^{\ 2\ \ s} \ .$$

- The min-entropy is determined by $\mathrm{E}(R_n) = \quad s$.

$\longrightarrow$ Online test should be based on $\mathrm{E}(R_n)$.

Online test

- The targeted min-entropy $\mathrm{H}\quad(Y_n) > 0.98$ can be achieved with $s = 0.15\,\mathrm{ms}$ which corresponds to bitrate $6.6\,\mathrm{kb/s}$.
- The designer picks $s = 0.2\,\mathrm{ms}$ which yields min-entropy $\mathrm{H}\quad(Y_n) > 0.995$ and bitrate $5.0\,\mathrm{kb/s}$.

$$
\begin{aligned}
A_{\text{bad}} &= \{\mathrm{H}\quad(Y_n) < 0.98\} &&= \{\mathrm{E}(R_n) < 2.14\} && \text{(bad parameters)} \\
A_{\text{good}} &= \{\mathrm{H}\quad(Y_n) > 0.98\} &&= \{\mathrm{E}(R_n) > 2.14\} && \text{(good parameters)} \\
A_{\text{real}} &= \{\mathrm{H}\quad(Y_n) > 0.995\} &&= \{\mathrm{E}(R_n) > 2.84\} && \text{(realistic parameters)}
\end{aligned}
$$

- Raise noise alarm if $\mathrm{SampleMean}(r_1, \ldots, r_m) < 2.49$.

Online test efficiency

- The efficiency of the online test can be determined by studying the distribution of the random variable $\mathrm{SampleMean}(R_1, \ldots, R_m)$.
- The random variable $m \cdot \mathrm{SampleMean}(R_1, \ldots, R_m)$ is $\mathrm{Pois}(m \;\; s)$-distributed.
- We can compute upper bounds for the failure probabilities:

Test sample size m	Pr(silent failure)	Pr(false alarm for params in A_{real})
128	$< 4.2 \cdot 10^{\;3}$	$< 8.1 \cdot 10^{\;3}$
256	$< 9.2 \cdot 10^{\;5}$	$< 3.5 \cdot 10^{\;4}$
512	$< 6.8 \cdot 10^{\;8}$	$< 7.7 \cdot 10^{\;7}$
1024	$< 4.4 \cdot 10^{\;14}$	$< 5.5 \cdot 10^{\;12}$
2048	$< 2.5 \cdot 10^{\;26}$	$< 4.0 \cdot 10^{\;22}$

A less efficient online test

- The internal random numbers $Y_1, Y_2, \ldots$ are iid $\mathrm{B}(1, p)$-distributed with $p = \Pr(Y_n = 1) = \mathrm{E}(Y_n)$.
- A test based on $\mathrm{SampleMean}(y_1, \ldots, y_m)$ requires way larger sample sizes for comparable failure probabilities:

Test sample size m	Pr(silent failure)	Pr(false alarm for params in A_{real})
262 144	$< 3.6 \cdot 10^{3}$	$< 3.6 \cdot 10^{3}$
524 288	$< 7.3 \cdot 10^{5}$	$< 7.3 \cdot 10^{5}$
1 048 576	$< 3.9 \cdot 10^{8}$	$< 3.9 \cdot 10^{8}$
2 097 152	$< 1.5 \cdot 10^{14}$	$< 1.6 \cdot 10^{14}$
4 194 304	$< 3.3 \cdot 10^{27}$	$< 3.3 \cdot 10^{27}$

Further health tests

- Repetition count test for the raw random numbers
- Check for extreme values of the online test
- Sample variance of the raw random numbers
- Correlation of the raw random numbers

Wrap-up

Summary

- Entropy sources:
  - Are fragile
  - Can have a few bad instances in large production run
  - Can fail silently – no way to see problem from outside
  - Total failure or entropy defect

- Health tests:
  - Are the only way to detect failures and defects
  - Based on design of entropy source
  - Closely tied to the stochastic model
  - Usually testing raw random numbers or internal values

Thank you for your attention!

Questions?

Further information and contact:

- SP 800-90: `https://csrc.nist.gov/Projects/random-bit-generation`
- AIS 20/31: `https://www.bsi.bund.de/dok/randomnumbergenerators`
- `john.kelsey@nist.gov`
- `johannes.mittmann@bsi.bund.de`