Enough Entropy? Justify It!

Yi Mao, Ph.D., CISSP
CST Lab Manager
atsec information security corp.
Email: yi@atsec.com

ICMC 2015, November 4-6, 2015, Hilton Washington, D.C.

Agenda

- Before IG 7.14 and IG 7.15
- IG 7.14 Entropy Caveats
- IG 7.15 Entropy Assessment
- CPU Time Jitter as an Entropy Source
- Ring Oscillator as an Entropy Source
- After IG 7.14 and IG 7.15

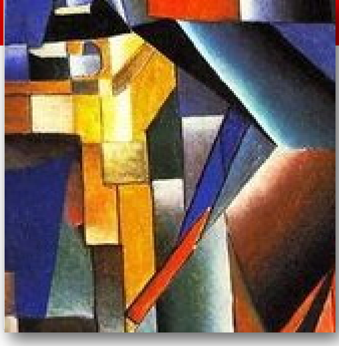
Let It Go (RNG Edition)

- Before IG 7.14 and IG 7.15

  Testing entropy is a chore

  Weak keys are out the door

FIPS 140-2 DTR AS.07.13



- **Guessing the seed value of an RNG (e.g. DRBG 800-90A) is at least as hard as guessing the generated key.**
- The burden of proof is on the *vendor.*

Vendor's wishes:

- We show you our design and data.
- You do the proof for us.

Entropy Test?

The Emotions of a CST Lab

Some Definitions of Key Strength

- NIST required minimum key strength: 112 bits
- Apparent key strength is determined by the key length according to the Table and formula in IG 7.5.
  - Triple_DES: 112 bits of security
  - AES 256: 256 bits of security
  - RSA 3072: 128 bits of security
  - RSA 4096: approximately 152 bits of security
- The real key strength is determined by the entropy in the RNG seed from which the key is generated.

How Much Entropy is Enough?

- At least 112 bits of entropy to seed the RNG that generates keys, <u>and</u>

- The amount of entropy in the RNG seed must be equal to or greater than all of the apparent strengths of the generated keys.

IG 7.14

Intended to answer the following questions:

- When is an entropy assessment necessary?
- How to handle cases when the entropy is insufficient?
  - Entropy doesn’t meet the minimum 112-bit strength, or
  - Is not sufficient to account for an apparent strength of the generated keys

When must a lab perform the entropy assessment?

- The entropy source is **within** the module boundary
  - Physical boundary for a hardware module
  - Logical boundary for a software module
  - Logical boundary for a sub-chip module (IG 1.20)
- The entropy source is **outside** the module boundary
  - Assess the entropy strength whenever possible.

Outcomes of in-Boundary Entropy Assessment

- ❑ Does not meet the minimum 112-bit strength
  - ➢ **STOP**: The module CANNOT be validated.
- ❑ Is not sufficient to account for an apparent strength of the generated keys
  - ➢ **GO**: The module CAN be validated.
  - ➢ **Caveat**: *The module generates cryptographic keys whose strengths are modified by available entropy.*
- ❑ Sufficient entropy: No entropy caveat on the certificate

Outcomes of out-Boundary Entropy Assessment (1)

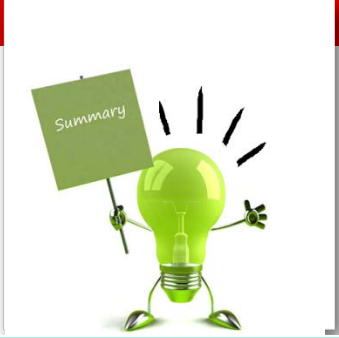
- Vendor/tester knows the entropy does not meet the minimum 112-bit strength.
  - **STOP**: The module CANNOT be validated.
- Vendor/tester doesn't know the minimum strength.
  - **GO**: The module CAN be validated.
  - **Caveat**: *There is no assurance of the minimum strength of generated keys.*
  - **In addition**, knows that the entropy is NOT sufficient to account for an apparent strength of the generated keys
  - **Caveat**: *The module generates cryptographic keys whose strengths are modified by available entropy.*

Outcomes of out-Boundary Entropy Assessment (2)

- Vendor/tester knows there are at least 112 bits minimum strength, but also knows the entropy is NOT sufficient to account for an apparent strength of the generated keys or doesn’t know if it’s sufficient.
  - **GO**: The module CAN be validated.
  - **Caveat**: *The module generates cryptographic keys whose strengths are modified by available entropy.*
- Vendor/tester knows there are at least 112 bits minimum strength and also knows there is sufficient entropy:

  No entropy caveat on certificate

Summary of Caveats



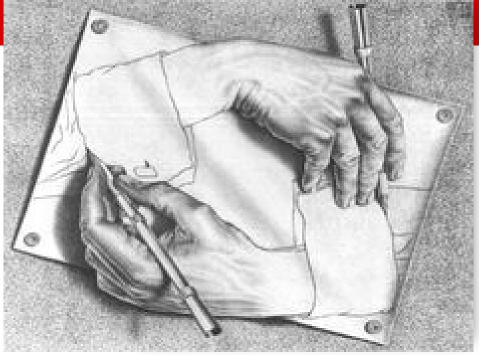
Is entropy source in or out of boundary?	Is minimum strength at least 112 bits?	Is sufficient to account for the apparent strength?	Can be FIPS validated? If yes, what caveats are applicable?
In, Out	No	No or Don't know	No
In, Out	No	Yes	Not logically possible
Out	Don't know	Don't know	Yes. "No Assurance" caveat
Out	Don't know	No	Yes. "No assurance" and "Modified Strength" caveats
Out	Don't know	Yes	Not logically possible
In, Out	Yes	No or Don't know	Yes. "Modified Strength" caveat
In, Out	Yes	Yes	Yes. No Caveat

“No assurance of minimum strength” Caveat for Porting

- ❑ Caveat in the module’s Security Policy:

  If porting to an untested platform is allowed when running a module on such an untested platform, the “No assurance of the minimum strength of generated keys” is applicable.

When I Know I Don’t Know



- The entropy source is **outside** the module boundary.
- The entropy input is passively loaded into the module.
- The module does NOT have control over the entropy input.

Warning sign to Federal Users:

- **“No assurance of the minimum strength” caveat**
- **Not all FIPS certificates are equal.**

Try to Know as Much as Possible

- ❑ Vendor: May I pretend I don't know the Entropy source?
  - ➢ **Within** the module boundary: No

    No knowledge of the entropy source,

    no FIPS certificate.
  - ➢ **Outside** the module boundary: Yes, if you wish.

    But … **caveats**!
- ❑ Better off: Have some control over the entropy source: actively getting, sanity checking, safeguards

To Know is to Verify

- Vendor: in the Security Policy:
  - State the minimum number of bits of entropy.
  - State the entropy estimate of the RNG seed.
- Lab: in a separate PDF report:
  - Confirm the entropy estimate by:
    - Reviewing the design of entropy source
    - Running statistical testing on the raw entropy data

7.15 Entropy Assessment

- Design Analysis First:
  - Not the comparison between the length of the seed and the length of a generated key,
  - But the comparison of the numbers of operations required to guess the seed and the generated key (i.e. the amount of entropy)
- Statistical Test Second:

NIST STS (SP 800-22)	http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html
NIST Python Script (SP 800-90B)	https://github.com/usnistgov/SP800-90B_EntropyAssessment
ENT	http://www.fourmilab.ch/random/

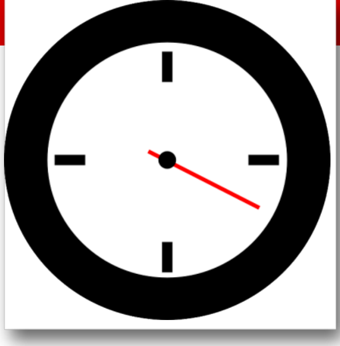
How to Report Entropy Estimation

- The lab **shall** provide a **PDF addendum** including:
  - A detailed logical diagram illustrating all entropy sources,
  - The tester’s arguments in support of the accuracy of vendor-provided rationale,
  - Results of statistical testing (optional but *strongly* recommended),
  - Specification of definition of entropy used (e.g. min-entropy, Shannon entropy).

NDRNG Approved for Use in Classified Applications

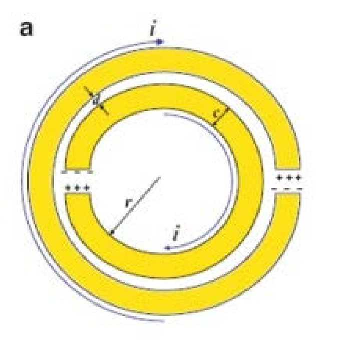
- The amount of entropy is assumed to be the length of the provided entropy string.
- No entropy estimation is required.
- The vendor may choose to claim a smaller amount of entropy.

CPU Time-Jitter Based Entropy



- Sampling frequency (e.g. once per second)
- Clock precision (e.g. nanosecond)
- How many bits are obtained from one clock reading (e.g. the least significant bit, the rightmost four bits)

Note: The sampling frequency shall be much slower than the clock frequency to ensure the independency of time readings.



Ring Oscillator Based Entropy

- ❑ Number of ring oscillators
- ❑ Positions of ring oscillators: no harmony over time
- ❑ The frequencies of ring oscillators: relative prime
- ❑ Sampling frequency of the ring oscillators

Note: The sampling frequency shall be much slower than the ring oscillators’ frequencies to allow the ring oscillators going through their full cycles.

Let the Tests Go On

- After IG 7.14 and IG 7.15

  An entropy test is a must

  The easy tester is gone

Visit atsec media webpage to see the video clips:

http://www.atsec.com/us/media.html

Thank you for your attention!