CMUF Entropy Working Group Potpourri

Joshua E. Hill, PhD

CMUF Entropy WG
20230725

Presentation version 20230725-2

(Post Meeting notes in orange.)

An Arbitrary Selection of Topics

- Crediting the XOR of Ring Oscillators (following on to the 20230627 CMUF) Entropy WG Meeting).
- FIPS 140-3 IG D.K Resolution 19.
- GitHub NIST tool news.

XOR in ROs

Crediting the XOR of Ring Oscillators

- An important part of XOR analysis (and most analysis) is simplifying.
    - As an initial approach, look at the independent case.
    - Track the most likely symbol (MLS).
    - Use a fancy symmetry argument (e.g., relate the terms to symmetric polynomials) or alternately crank through a proof by cases.

Crediting the XOR of Ring Oscillators

- If you have two independent bits with known min entropy m_1 and m_2, then the most likely symbols have probability $p_1 = 2^{-m_1}$ and $p_2 = 2^{-m_2}$.
  - We are looking at the most likely symbol, it has probability $p_i \geq \frac{1}{2}$, so these probabilities can be written as $p_1 = \frac{1+\varepsilon_1}{2}$ and $p_2 = \frac{1+\varepsilon_2}{2}$ for some $0 \leq \varepsilon_i \leq 1$. (This represents the bias toward some particular output bit).
  - Note the most likely symbol need not be fixed!

Crediting the XOR of Ring Oscillators

- In each of the four possible cases (the most likely symbols for both bits are both 0, both 1, or mixed 0 and 1), the probability of the most likely symbol of the XOR of these two bits is $\frac{1+\varepsilon_1\varepsilon_2}{2}$, so the resulting min entropy is a direct result of this probability.

$$H_{\text{out}} = -\log_2\left(\frac{1+\varepsilon_1\varepsilon_2}{2}\right)$$

What About Independence?

- You don’t (almost certainly) have independent outputs.
- You could track the mutual information in all possible combinations and credit it correctly (using some sort of Principle of Inclusion-Exclusion argument) but this is unpleasant.
  - Particularly with lots of bits!
- A more practical approach is to credit only the variation that can be viewed as independent.
- The entropy due to the independent variation does not credit mutual information between RO outputs, so this component of the entropy can be viewed as independent.

FIPS 140-3 IG D.K Resolution 19

The Text (Part I)

- To receive full entropy from the output of a conditioning component, the following criteria must be met:
  - The conditioning component shall be vetted,
  - h_{in} shall be greater than or equal to $n_{\text{out}} + 64$ bits,
  - n_{out} shall be less than or equal to the security strength of the cryptographic function used as the conditioning component.

Issues (Part I)

- How is the “security strength” determined for:
    - CMAC?
    - CBC-MAC?
    - Hash_df?
    - Block_cipher_df?

Issues

- Some of these primitives (the derivation functions) do not have a fixed size of output.
    - [Meeting note: ESV enforces 90B Table 1 n_{out} as the fixed size of output.]
- The narrowest width is relevant to this argument.
- SP 800-90C pd3 deals with this by fixing the output size to the block size of the underlying primitive.
    - This step is necessary.
    - This is not done in Resolution 19.
    - [Meeting note: The SP 800-90B Table 1 n_{out} values are consistent with the SP 800-90C pd3 requirements.]

The Text (Part II)

- Note 1. If n_{in} bits of full entropy are provided to a vetted conditioning component, then the output of the conditioning component will maintain full entropy.

Issues (Part II)

- The use of n_{in} is wrong. When a conditioning function gets data, it gets (at least) n_{in} bits of it; that is the meaning of n_{in}.
- This note is true iff the vetted conditioning is a bijection.
  - None of the vetted conditioning functions are bijections in standard use.
- Even in the “ideal” case this can have problems.
  - If the output size is less than or equal to the input size, there are still collisions, and thus a min entropy reduction.

Issues (Part II)

- In the “less than ideal” case, this leads to insanity.
  - What if the output size is larger than the input size?
  - E.g., A 128 bit string of “full entropy” data goes into a SHA-512 vetted conditioner (so $n_{\text{in}} = 128$, $n_{\text{out}} = 512$). The result is clearly not a full entropy 512 bit string!

Post Meeting Notes

- [Meeting note: Chris (NIST ESV) stated that it was CMVP’s intent that this applied in addition to (perhaps some of?) the Resolution 19 criteria.]
- Should “Note 1” instead say:
    - Note 1. If n_{out} bits of full entropy are provided to a vetted conditioning component, then the output of the conditioning component will maintain full entropy.
        - If so, this is still not technically correct, but avoids the biggest problems.
        - The security strength still needs to be larger than n_{out}.

Proposed Resolution

- Mirror SP 800-90C PD3.
  - Fix the output size of derivation functions.
  - [Meeting note: ESV enforces 90B Table 1 n_{out} as the fixed size of output.]
- Publish a table providing “security strengths” for all vetted conditioning functions.
- Remove “Note 1”.
  - At least correct “Note 1”.

New GitHub PRs

New Grist for the NIST Tool Mill!

- PR #226 is essentially the same as PR #217.
  - This adds large file support to the t-tuple and LRS estimators.
  - Particularly useful for non-vetted conditioning analysis.
- PR #224 makes restart testing faster (and allows for larger-scale simulations for finding the X_{cutoff} parameter).
- PR #225 removes the “full entropy” criteria from earlier (2012 and 2016) drafts of SP 800-90C and adds the IG D.K Resolution 19 logic.