# Comments Received on
# NIST SP 800-90B:
# *Recommendation for the Entropy Sources Used for Random Bit Generation*

Round two of comments

Comment due date:  May 9, 2016

From: Tim Myers
**Sent:** Tuesday, April 19, 2016 7:15 PM

| # | Organization | Com. | Type | Page # | Line # | Section | Comment (Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | Microsoft | Niels Ferguson | T | 11 | 545 | 3.1.4.3 | The term Z_(1-\alpha) does not seem to be defined anywhere. | Define Z_(1-\alpha). |
| 2 | Microsoft | Niels Ferguson | T | 22 | 926 | 4.4.2 | What is 'the next sample' ? | |
| 3 | Microsoft | Niels Ferguson | T | 26 | 1015 | Figure 5 | The While loop condition should be >=1, not >1. As-is, the algorithm never considers swapping s_0 and s_1, leading to a non-uniform permutation distribution. | While (i >= 1) |

From: Buller, Darryl M
Sent: Wednesday, April 13, 2016 1:23 PM

The following comments and suggestions pertain only to the entropy estimators described in Section 6 of SP 800-90B. This is the only topic that we have focused on.

**Most Common Value Estimate Comments:**

- In Section 6.1, the following is stated:

  > "It is important to note that the [MCV] estimate provides an overestimation when the samples from the source are not IID".

  This is usually the case, but is not necessarily the case. We propose replacing the above sentence in Section 6.1 with:

  > "It is important to note that this estimate typically provides an overestimation when the samples from the source are not IID."

  And adding the following explanation as a footnote:

  > "However, it is actually possible for this estimate to slightly underestimate the true min-entropy. It is believed that this underestimation is likely to not exceed one bit because of the relationship between min-entropy and expected guessing work derived in Appendix D. Of course, such an underestimate would not indicate that a guessing attack which ignores dependencies could be less costly than one that takes the dependencies into account. As an example, consider a data sample consisting of pairs of bytes generated from the joint distribution on two bytes X and Y, each having possible values A and B, where P(X=A,Y=A)=0.104, P(X=A,Y=B)=0.332, P(X=B,Y=A)=0.239, and P(X=B,Y=B)=0.325. The min-entropy according to the MCV estimator is 0.712, while the true min-entropy is 0.795."

**Hagerty/Draper Estimator Comments:**

- In Step 9 of the Collision Estimate, $k$ is defined to be "the number of possible values." Should "possible" be changed to "uniquely observed"? We only know the observed values from the data, but not necessarily all possible values that the entropy source can produce.

- In Step 6 of the Compression Estimate, the $n$ should be changed to $k$ in the equation $\bar{X}' = G(p) + (n-1)G(q)$.

- For the binary search used in the Collision and Compression estimates, the document should say to initialize the binary search lower and upper bounds to $\frac{1}{k}$ and 1 respectively. If 0 is used as the initial lower bound, the search can miss the correct value of $p$.

- All of the Markov estimate example values appear to be inaccurate, except for $\alpha$, and should be changed to the following:

- After Step 2
    - $\varepsilon = 0.1054$
    - $P_1 = 0.4863$
    - $P_2 = 0.4387$
    - $P_3 = 0.3911$
- After Step 4, the bounding matrix $T$ has values:

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0.4682 | 0.7540 | 0.3254 |
| 1 | 0.6111 | 0.3254 | 0.6111 |
| 2 | 0.6972 | 0.5305 | 0.3638 |

- After Step 5a, the iteration for $j = 1$ has completed:
    - $P_1 = 0.2727$
    - $P_2 = 0.3667$
    - $P_3 = 0.2681$
- After Step 6, the highest probability of any chain of length 128 generated by this bounding matrix is $7.0706 * 10^{-22}$, yielding an estimated min-entropy of $0.5489$.

- In your "Predictive Models for Min-Entropy Estimation" paper, where you introduce the new predictor estimates (now included in 90B), you tested the Hagerty/Draper estimators against the predictor estimators using data that is produced from real-world RNGs. For the Random.org data, you state that the Hagerty/Draper tests produce min-entropy estimates between 5.1830 and 5.6662. However, the majority of predictor estimates gave results that are closer to 8, which is expected. You concluded the following: "Although we cannot prove it, we suspect that this discrepancy comes from the inaccuracy of the [Hagerty/Draper] estimators, rather than a weakness of the source." To follow up with your observation, we generated a large collection of IID data files and found that the Collision and Compression estimators dominate the results so that the minimum estimator value is almost always one of these two values. These estimators were the minimum estimator value approximately 25% and 73% of the time, respectively, and each of these estimators tends to significantly underestimate the true min-entropy for this type of data file. In these cases the output of the other estimators, although more accurate, are ignored. Although these tests were not designed for IID data, it seems plausible that similar results could also occur for data that is very nearly IID but is deemed to be non-IID; e.g., if only one test or a small number of tests for IID reject the IID hypothesis. Therefore we are basically pointing out the same inaccuracy that you observed as well.

## Tuple Estimator Comments:

- The t-tuple estimate has been changed to at least a maximum 35 samples per tuple size. The upper bound for the largest number of occurrences for a given tuple size in the LRS estimate is still set to less than 20 and should be changed to less than 35.

- Why do the t-tuple and LRS estimates not use the upper-end of a 99% confidence interval to estimate the probability of each tuple? This appears to be inconsistent with the Most Common Value ($p_u$) and predictor estimates ($P'_{global}$).

## Predictor Estimator Comments:

- For the four predictor estimates, instead of having

$$P'_{global} = P_{global} + 2.576\sqrt{\frac{P_{global}(1-P_{global})}{N-1}}$$

we suggest

$$P'_{global} = min\left(1,\ P_{global} + 2.576\sqrt{\frac{P_{global}(1-P_{global})}{N-1}}\right)$$

since $P_{global} + 2.576\sqrt{\frac{P_{global}(1-P_{global})}{N-1}}$ might be greater than 1.

- In the “Predictive Models for Min-Entropy Estimation” paper and in NIST’s implementation, $P'_{global}$ is set to $1-\left(\frac{\alpha}{2}\right)^{\frac{1}{N}}$ when $P_{global} = 0$. This should also be stated in the standard for each of the predictor estimates when computing $P'_{global}$. This computation seems to be finding the value of $P_{global}$ that makes the probability of no correct predictions be $\frac{\alpha}{2}$. However, should the $\frac{\alpha}{2}$ be $\alpha$? This would be consistent with the one-tail test implied by the value of 0.99 used in the estimate of $P_{local}$.

- In the predictor estimates, it is possible that $\max(P'_{global}, P_{local})$ could be less than $\frac{1}{k}$, where $k$ is the alphabet size, which results in the min-entropy estimate being greater than $\log_2 k$, which is not possible. In this case, we would consider the estimate to be inconclusive, and $\log_2 k$ should be returned as the estimate. This can easily be implemented by replacing $\max(P'_{global}, P_{local})$ with $\max(P'_{global}, P_{local}, \frac{1}{k})$.

- For the four predictor estimates, $P_{local}$ appears to be inaccurate in the provided examples. We believe the following are more accurate values:

  - MultiMCW: $P_{local} = 0.0306$
  - Lag: $P_{local} = 0.1115, minentropy = 0.7349$
  - MultiMMC: $P_{local} = 0.1167$
  - LZ78Y: $P_{local} = 0.1001$

  We also note that the function to which the binary search is being applied to find $P_{local}$ apparently has a very small slope in the region of interest, so that any value in a relatively large range will return the desired value 0.99.

- The MultiMMC estimate defines the variable $M_d$ to denote the number of observed transitions for the Markov model. However, this variable is later referred to as $MMC_d$ in Step 4a. This should be changed so that the names are consistent.

- For both the multiMMC and LZ78Y estimates, it is not clear how to choose *ymax* if multiple counts have the same max value. For instance, when $i = 9$ in the multiMMC example, the predicted value is "2". The transition counts for $M_1$ at this point are: {1->3: 3}, {2->1: 2}, {3->1: 1}, {3->2: 1}, and $s_8 = 3$. Therefore, the two possible transitions are {3->1: 1} and {3->2: 1}. However, both transitions have one count. So either "1" or "2" could be predicted. Since "2" is the predicted value, should we just take the output with highest lexicographical (or numeric) ordering if there is a count tie? This seems to be the case here, since 2 > 1. In the code from NIST we saw that the multiMMC in fact does break the tie according to the highest byte value; e.g., if D[prev][y1]=D[prev][y2], and y1=0x40 and y2=0x0a, then *ymax*=y1. We also suggested that LZ78Y code break ties this way as well.

- The multiMMC estimate uses the concept of building Markov models of order 1 through 16. However, this estimate does not limit the number of observed previous states that can be incorporated into each model, whereas the LZ78Y allows for a maximum of 65536 previous states across all of the 16 models. We have observed that it is sometimes infeasible to run the multiMMC on larger files (e.g., 10MB) because there are too many previous states to store. Although we do not have a concrete suggestion at this time, we suggest considering a transition limit for this estimate. This limit should be chosen such that the multiMMC estimate is feasible to compute across large files.

- We understand the purpose of $P_{local}$ in the predictor estimates. However, it was not initially obvious to us that choosing $\max(P'_{global}, P_{local})$ is indeed a one-tail hypothesis test that rejects $P'_{global}$ in favor of $P_{local}$ at the 99% significance level. To make this clearer, we suggest adding the following paragraph in Appendix H.2 (Predictors) after the sentence ending with "theory of runs and recurrent events [Fel50]."

In order to make the predictor estimates lean toward a conservative underestimate of min-entropy, $P_{global}$ is replaced by $P'_{global}$, the proportion corresponding to the 99th percentile of the number of correct predictions based on the observed number of correct predictions. Note that the order in which correct predictions occur does not influence the min-entropy estimate based on $P_{global}$. For example, a predictor could always be correct for the first half of the outputs in a data set, and always incorrect for the second half of the outputs. The min-entropy estimate of this sequence, based on $P_{global}$, is half the data length in bits. On the other hand, for another sequence, the predictor could have a 50% chance of being correct for every sample in this sequence. The min-entropy estimate of this second sequence, based on $P_{global}$, is the same as that of the first sequence. However, the typical successful prediction run lengths are very different for these two sequences. Therefore, the proposed scheme takes the local prediction performance into account in order to conservatively decrease the min-entropy estimate if the observed local prediction behavior is statistically significant given the global prediction success rate. The predictor estimates accomplish this by basing the min-entropy estimate on $\max(P'_{global}, P_{local})$, where $P_{local}$ is the successful prediction proportion for which the observed longest run of correct predictions is the 99th percentile. This is effectively a one-tail hypothesis test that rejects $P'_{global}$ in favor of $P_{local}$ if the observed longest run, given a success probability of $P'_{global}$, is beyond the 99th percentile.

From: Jose Emilio Rico
**Sent:** Thursday, May 05, 2016 4:22 AM

| # | Org. | Com. | Type | P# | L# | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | E&E | MGR | T | 3 | 320 | 2.1 | $H = -\min_{1\le i\le k}(-\log_2 p_i)$<br>(The formula is incorrect) | $H = \min_{1\le i\le k}(-\log_2 p_i)$ |
| 2 | E&E | MGR | E | 4 | 338 | 2.2.1 | sourse<br>(Typo) | source |
| 3 | E&E | MGR | G | 9 | 459 | 3.1.1 | jth<br>(to be consistent with the same expression along document) | $j^{th}$ |
| 4 | E&E | MGR | G | 9 | 460 | 3.1.1 | ith<br>(to be consistent with the same expression along document) | $i^{th}$ |
| 5 | E&E | MGR | G | 22 | 938 | 4.4.2 | where p = $2^{-H}$<br>(misplaced) | Delete the sentence. |
| 6 | E&E | MGR | T | 29 | 1113 | 5.1.7 | i=i+j+1<br>(The formula is incorrect) | i=i+j |
| 7 | E&E | MGR | T | 30 | 1132 | 5.1.8 | i=i+j+1<br>(The formula is incorrect) | i=i+j |
| 8 | E&E | MGR | T | 32 | table below line 1196 | 5.2.1 | The values $e_{i,j}$ shown in the table have been calculated according to formula $e_{i,j} = p_i p_j(L)$ instead of $e_{i,j} = p_i p_j(L-1)$. | Calculate the $e_{i,j}$ values according to formula $e_{i,j} = p_i p_j(L-1)$. For example, $e_{1,1} = 0.21 \cdot 0.21 \cdot (100-1) = 4.37$ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | E&E | MGR | G | 33 | 1221 | 5.2.2 | The second bin contains sample 2<br>(wrong sentence) | The second bin contains sample 3 |
| 10 | E&E | MGR | G | 33 | 1222,<br>1223 | 5.2.2 | and the last bin contain 3.<br>(wrong sentence) | and the last bin contain 2. |
| 11 | E&E | MGR | G | 33 | 1225 | 5.2.2 | the remaining bits<br>(not accurate sentence) | the remaining samples |
| 12 | E&E | MGR | T | 34 | 1247 | 5.2.3 | the value of *m* is selected as 3<br>(considering $p_0$=0.14 and L=1000, if *m* is selected as 3, the inequation $(p_0)^m$>5/L is not met) | the value of *m* is selected as 2 |
| 13 | E&E | MGR | T | 39 | 1424 | 6.3.3 | *Define the confidence level to be* $\alpha = \min(0.99^{k^2}, 0.99^{d})$<br>(concept error) | *Define the significance level to be* $\alpha = \min(0.99^{k^2}, 0.99^{d})$ |
| 14 | E&E | MGR | T | 40 | 1457 | 6.3.3 | ε = 0.0877<br>(wrong ε value, so subsequent values of $P_1$, $P_2$, $P_3$, etc. are wrong too) | ε = 0.1054 |

From: Albert MARTINEZ
**Sent:** Wednesday, May 04, 2016 9:35 AM

| # | Org. | Com. | T | P # | L # | Sec. | Comment(Include rationale for comment) | Sugg. change |
|---|---|---|---|---|---|---|---|---|
| 1 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | | | 3.2.2 | *Post-processing functions (Section 3.2.2): We provided a list of approved post-processing functions. Is the selection of the functions appropriate?*<br>It would be good to have the freedom of choosing a post-processing and then possibly showing why it's sound, as already done for the conditioning. It would then require to establish/describe the required properties of such functions so that the applicant is able to show how it fulfills the requirements | |
| 2 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | | | 3.1.5 | *Entropy assessment (Section 3.1.5): While estimating the entropy for entropy sources using a conditioning component, the values of n and q are multiplied by the constant 0.85. Is the selection of this constant reasonable?*<br>Besides the value itself, and before answering on this, we first need to understand how this constant has been built (rationale) as mentioned in our comments below | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | ST | Albert Martinez,Jean Nicolai,Yannick Teglia | | | | | *Multiple noise sources: The Recommendation only allows using multiple noise sources if the noise sources are independent. Should the use of dependent noise sources also be allowed, and if so, how can we calculate an entropy assessment in this case?*The key word here is “independent”. We need to understand what it means and how the applicant has to measure and prove this independence. And in case of “soft dependence”, if allowed, what would be the authorized maximum ? | |
| 4 | ST | Albert Martinez,<br>Jean Nicolai,<br>Yannick Teglia | | | | | *Health Tests: What actions should be taken when health tests raise an alarm? The minimum allowed value of a type I error for health testing is selected as 2-50. Is this selection reasonable?*<br>If an alarm is raised, it has to be reported to the upper layers ; it’s up to them to decide the action to take depending on the context. The choice of 2**-50 is reasonable if we the objective is to discard obvious and intolerable statistical weakness of the RNG. It’s not if the purpose of such tests is to estimate the statistical quality of the RNG. | |
| 5 | ST | Albert Martinez,Jean Nicolai,Yannick Teglia | | 4 | 340 | | There is a new item called « post processing » that is intended for reducing the bias. It is mentioned as optional in the text but is not mentioned as such in the drawing. Shall we consider it optional or not? | |
| 6 | ST | Albert Martinez,<br>Jean Nicolai,<br>Yannick Teglia | | 4 | | | Health tests are supposed to be processed after the post-processing, but before the conditioning, according to the drawing. It is the case?<br>We don’t understand the rationale of performing the tests in between two processing; is it because the first one is supposed to be simple and therefore won’t hide statistical weaknesses? | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | ST | Albert Martinez,Jean Nicolai,Yannick Teglia | | 5 | 338, 356 | | Two elements are intended for reducing the bias and increasing the entropy per bit, the post-processing (338) and the conditioning (356).Regarding the handle “get_entropy()”, it is not clear to us if the provided string shall contain the required entropy or the required entropy per bit. Said differently, shall we provide a string S that will contain the required entropy but possibly with a post-processing to be performed by the customer to get the required entropy per bit, or alternatively, the requested entropy per bit (i.e. the compression of the string has already been performed).We think that, especially in the latter case, having such a function could be a “self-test” for an attacker, enabling him to know if its current attack (for instance through electromagnetic fault injection) is having an effect on the entropy by getting information on the updated value of the entropy. | |
| 8 | ST | Albert Martinez,Jean Nicolai,Yannick Teglia | | 9 | 445 | | Raw sample are to be taken for validation testing. Are we dealing here with sample before or after this new post-processing item ?Suppose we’re using several ring oscillators as a noise sources and exclusive-oring (xor) them to get a single one; if the sampling is done after the xor, it is considered to be a single noise source. What if the sampling is done on each ring oscillator before the xor; is it still considered to be a single noise source or the combination of several noise sources? In the latter case how do we consider the xor? Do we consider it as a post-processing, a conditioning, something else? We would like the NIST to advise on this point. | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 10 | 494 | | The submitter has to provide an entropy estimate. Is it supposed to be based on a characterization to be performed in the submitter premises or rather based on theoretical elements?<br>Besides the value, what clues have to be provided to the evaluator to assess the soundness of the estimate? | |
| 10 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 10 | 507 | | It is mentioned that the entropy source shall be restarted. Does it mean that it has to be powered-off ?<br>In the affirmative, what should be the maximum delay between the power-off and the following power-on? | |
| 11 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 14 | | | Could you explain/detail the rationale for the value 0.85 beyond the note p14 | |
| 12 | ST | Albert Martinez,Jean Nicolai,Yannick Teglia | | 14 | | 3.1.6 | Paragraph 3.1.6 is not clear to us; in case of k entropy source, each having a width of d bits, shall we consider the input to the conditioning as being the sum (over GF(2)) of the k sources that yields a d-bit wide input or rather the concatenation of the k sources, yielding a k*d-bit wide input? | |
| 13 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 16 | 678 | | You’re mentioning the exclusive use of independent sources;<br>Could you provide a definition of this independence?<br>How will we have to provide evidence of this independence? | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 14 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 16 | 700 | | Do we also need to consider physical tampering?<br>In the affirmative, do we need to also consider tampering when device is powered-off? | |
| 15 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 19 | 802 | | It is mentioned that the entropy source is required to be capable of performing on-demand health tests. We understand that this exclude the possibility of using a hardware/software partitioning where for instance a central processor is collecting the stream for the noise source and then performing the tests. Are we correct? | |
| 16 | ST | Albert Martinez, Jean Nicolai, Yannick Teglia | | 22 | | 4.4.2 | Adaptive Proportion test triggers an error if number B of occurrences of a value in a window W is greater than cutoff C. In the particular case of binary data, we think we could extend with low cost the test by checking that B =< C but also B > W-C.<br>This would guarantee that a binary value occuring too frequently will be caught on the first test window, no need to rely on the chance that the too frequent value is in first position in the test window | |

From: John Leiseboer
Sent: Tuesday, May 03, 2016 2:42 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | QuintessenceLabs | John Leiseboer | T | 23 | 946 | 4.4.2 | Some values in tables 2 and 3 differ when calculated using both Excel and also Mathematica. In Table 2, Row 3 (H=0.4), Column 2 (Cutoff value). Cutoff value = 867 | In Table 2, Row 3 (H=0.4), Column 2 (Cutoff value). Cutoff value = 867 |
| 2 | QuintessenceLabs | John Leiseboer | T | 23 | 946 | 4.4.2 | In Table 2, Row 5 (H=0.8), Column 2 (Cutoff value). Cutoff value = 697 | In Table 2, Row 5 (H=0.8), Column 2 (Cutoff value). Cutoff value = 697 |
| 3 | QuintessenceLabs | John Leiseboer | T | 23 | 947 | 4.4.2 | In Table 3, Row 2 (H=0.2), Column 2 (Cutoff value). Cutoff value = 492 | In Table 3, Row 2 (H=0.2), Column 2 (Cutoff value). Cutoff value = 492 |
| 4 | QuintessenceLabs | John Leiseboer | T | 23 | 947 | 4.4.2 | In Table 3, Row 3 (H=0.5), Column 2 (Cutoff value). Cutoff value = 430 | In Table 3, Row 3 (H=0.5), Column 2 (Cutoff value). Cutoff value = 430 |
| 5 | QuintessenceLabs | John Leiseboer | T | 37 | 1367 | 6.3.1 | There can be two solutions for the parameter p: In Example (Line 1378) the two resulting solutions for p are p=0.0205 and p=0.7062. Should the parameter search space for p be refined? | Should the parameter search space for p be refined? |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | QuintessenceLabs | John Leiseboer | T | 38 | 1382 | 6.3.3 | Since an alphabet size of more than 2^6 cannot be used, the entropy estimate provided by the implementation of the Markov test is upper-bounded by 6. For samples with large alphabet size (e.g. 16-bit) this test can be the limiting test.  Is it possible to provide either a) a sample requirement to estimate the entropy according to the Markov test for larger alphabet sizes, or b) the option to not use the Markov test if the alphabet size is more than 2^6? | Is it possible to provide either a) a sample requirement to estimate the entropy according to the Markov test for larger alphabet sizes, or b) the option to not use the Markov test if the alphabet size is more than 2^6? |
| 7 | QuintessenceLabs | John Leiseboer | T | 40 | 1434 | 6.3.3 | Is the following condition necessary  (T_(i,j)=1,if o_i=0)? Since in previous paragraph (Line 1417) the output is adjusted to have consecutive values. | Is the following condition necessary  (T_(i,j)=1,if o_i=0)? |
| 8 | QuintessenceLabs | John Leiseboer | T | 40 | 1457 | 6.3.3 | Should the Logarithms in Equations Line 1430 and Line 1439 be base-2 or base-e? The example test vectors can be replicated by replacing the base-2 with a base-e Logarithm in these two equations. | Should the Logarithms in Equations Line 1430 and Line 1439 be base-2 or base-e? |
| 9 | QuintessenceLabs | John Leiseboer | T | 42 | 1514 | 6.3.4 | Parameter n should be replaced with k. | Parameter n should be replaced with k. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 10 | QuintessenceLabs | John Leiseboer | T | 45 | 1595 | 6.3.7 | Taking the log of this equation on both sides and solving for P_local is robust to overflows, otherwise x^(N+1) overflows. | Taking the log of this equation on both sides and solving for P_local is robust to overflows |
| 11 | QuintessenceLabs | John Leiseboer | T | 43 | 1531 | 6.3.5 | A complete Example should be included for the t-Tuple Estimate method. | A complete Example should be included for the t-Tuple Estimate method. |
| 12 | QuintessenceLabs | John Leiseboer | T | 43 | 1547 | 6.3.6 | An Example should be included for the LRS Estimate method. | An Example should be included for the LRS Estimate method. |
| 13 | QuintessenceLabs | John Leiseboer | T | 48 | 1663 | 6.3.9 | For consistency change notation “MMCd” to “Md” | For consistency change notation “MMCd” to “Md” |
| 14 | QuintessenceLabs | John Leiseboer | T | 48 | 1665 | 6.3.9 | Include the condition: “If d<i, ….” | Include the condition: “If d<i, ….” |
| 15 | QuintessenceLabs | John Leiseboer | T | 63 | 1855 | Appendix E | Can we apply the post-processing functions to non-binary noise sources? | Can we apply the post-processing functions to non-binary noise sources? |
| 16 | QuintessenceLabs | John Leiseboer | T | 63 | 1855 | Appendix E | We propose an approach to whiten a non-uniform symmetric noise source (e.g. a Gaussian) by dropping the most significant bits. Is this a suitable post-processing function for non-binary noise sources? | We propose an approach to whiten a non-uniform symmetric noise source (e.g. a Gaussian) by dropping the most significant bits. |
| 17 | QuintessenceLabs | John Leiseboer | T | 46 | 1606 | 6.3.7 | We found some inconsistencies regarding the computation of P_local in all examples. We found P_local=0.03597 | We found P_local=0.03597 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 18 | QuintessenceLabs | John Leiseboer | T | 47 | 1644 | 6.3.8 | We found P_local=0.1167 and thus H=0.7349 | We found P_local=0.1167 and thus H=0.7349 |
| 19 | QuintessenceLabs | John Leiseboer | T | 49 | 1696 | 6.3.9 | We found P_local=0.1307 | We found P_local=0.1307 |
| 20 | QuintessenceLabs | John Leiseboer | T | 52 | 1740 | 6.3.10 | We found P_local=0.1230 | We found P_local=0.1230 |
| 21 | QuintessenceLabs | John Leiseboer | T | 41 | 1466 | 6.3.4 | The compression test appears to perform badly when the alphabet size is very large, and the alphabets are non-uniformly distributed. For example, in the case where the distribution over the sample space is Gaussian, some symbols will be much more unlikely than others, and for large alphabet sizes this causes the entropy estimates to fall drastically when the full space is used. This creates an effect where including more significant bits in the sample reduces the entropy estimate (which does not reflect reality, where including more bits should not reduce the entropy of a sample). | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 22 | QuintessenceLabs | John Leiseboer | T | 10 | 508 | 3.1.4.1 | We would like some clarification on the following statement: "For each restart, c=1000 consecutive samples shall be collected directly from the noise source". This statement does not explicitly say when the samples shall be collected after restart. After restarting a noise source, firstly the sub-systems must first come-up, then be retrained and built-in tests run. This takes time. Would noise samples after this startup process be sufficient for the restart data test? | |

From: Harris, Michael W.
**Sent:** Monday, May 02, 2016 4:58 PM

CDC has no comments to provide on the *Draft NIST Special Publication 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation*.

Thank you for the opportunity to review and comment.

**Michael Harris, CISSP**

From: Jw Choi
**Sent:** Monday, May 09, 2016 5:48 AM

| # | Organization | Com. | Type | P# | L# | Sec. | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| | SKtelecom | Jeong Woon Choi | T | 37 | 1340 | 6.3.1 | I suggest that an Asymptotic Limit of entropy estimation should be strongly considered. It is because the current version is unfair to pretty good entropy sources. Lets consider an ideal binary source with even prob. of 0 and 1. In this case, can the estimated min-entropy reach to the ideal min-entropy 1? It is impossible even though it is ideal. Of course, it can fail by a very small gap like 0.000xxx. However, this small gap make a big difference and a big lost of entropy when we use a conditioning function. It is needed to have in mind the definition of full entropy again here. Full entropy does not require an exact value, but an asymptotic value which allows a small lack of entropy, 2^-64 times n. Conservative estimation is necessary for security. However, reasonable and fair approach is also important. When an entropy of some source goes up to a certain value A as the size of dataset gets bigger, my suggestion is to give it A as an estimated entropy, even though in a finite set it does not reach to A. This approach can be fairly used in SP800-90B/C with respect to definition of (full) entropy. | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SKtelecom | Jeong Woon Choi | G | | | | [Question] Why do you allow a full entropy only based on a conditioning or a  DRBG? | |
| | SKtelecom | Jeong Woon Choi | G | | | | [Question] Why is only a half of security strength of a conditioning or a DRBG allowed for full entropy? | |

From: Alessandro Tomasi, Alessio Meneghetti
**Sent:** Monday, May 09, 2016 11:02 AM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | UniTN | AM, AT | | 14 | 637 | 3.1.6 | Conceptually there is little difference between two sources producing individual dependent bits and a single source producing two non-independent bits in a single sample. | Add: 'Multiple dependent noise sources may be used as long as they can be, and shall be, considered as producing a single output in a larger sample space. The burden shall be on the vendor to provide a convincing estimate of the entropy defined on the larger sample space so that it may be tested accordingly. |
| 2 | UniTN | AM, AT | | 4 | 339 | 2.2.1 | We absolutely need a way to distinguish between the data coming from the noise source and that being output by any subsequent processing algorithm, and the former should be subjected to testing to validate the entropy it provides. | The output of the digitized noise source is called the digitized noise. |
| 3 | UniTN | AM, AT | | 52 | 1761 | 6.4 | Algorithm 6.4 is open to interpretation and would not in general yield an optimal choice. | [...] where n exceeds the bit length m that can be handled by the test, the submitter shall indicate and motivate a selection of the m-tuple they deem to have the highest joint entropy. [cut to line 1781] |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | UniTN | AM, AT | | 64 | 1868 | Appendix E | It should be noted that the bias in sources producing samples with correlated bits might increase after applying this technique. | [Add:] It should be noted that the bias in sources producing samples with correlated bits might increase after applying this technique. |

Multiple noise sources (6.3.1)

Dependent noise sources should be allowed, in general; conceptually there is little difference between two sources producing individual dependent bits and a single source producing two non-independent bits in a single sample. There are however so many ways in which two dependent sources might be embodied and combined that we suggest the following criterion: multiple dependent noise sources should be allowed, as long as they can be, and shall be, considered as producing a single output in a larger sample space. The burden shall be on the submitter to provide a convincing estimate of the entropy defined on the larger sample space so that it may be tested accordingly.

For example, in the case of sources operating at the same frequency, the sample space will be the Cartesian product of the individual sample spaces, the distribution will be the distribution of a single variable over the larger space, and hence the entropy should be computed as the entropy of a variable in the larger sample space.

This criterion is motivated by a need to be able to quantify the joint entropy of the sources. It should be possible to satisfy this requirement for sources that are synchronous or at least have a common frequency; in the case of sources operating at different frequencies there may be a least common multiple among the frequencies at which the sources produce output, so that there will exist a frequency at which they all output an integer number of sub-samples.

Entropy source model (2.2), in particular: definition of noise source (2.2.1)

Line 339: The output of the digitized and optionally post-processed noise source is called the raw data. The following observations stand out to us.

1. Conceptually, it makes little sense to refer to post-processed data as ‘raw’.

2. We also lack a term to refer to the digitized sequence before the application of post-processing, which makes little sense since an estimate of the entropy after post-processing cannot be reliable unless there is a thorough description of the signal before it.
3. There is no explanation as to why certain functions are referred to as post-processing and others as conditioning, so we don't see the need for this distinction.
4. It makes no sense to design health tests to be applied before conditioning but after post-processing. Either the intent is to measure the entropy provided by the noise source as accurately as possible, in which case it is surely best to act on the digitized sequence directly, or the intent is to simply check that the final output is passable enough, in which case the tests should be applied on the output of the conditioning function, since both conditioning and post-processing are stated to be optional, and since this would be less computationally intensive given that the conditioning is a further compression.

We accept that certain implementations combine the digitization and post-processing steps - which is actually not foreseen by this draft, since digitization is assumed to occur before post-processing - but a proper description of the source still requires a full understanding of the sample distribution without post-processing.

Based on these observations, we strongly recommend taking post-processing outside the definition of noise source, and requiring that the noise source shall be described in sufficient detail to make an entropy estimate before the application of any post-processing, including approved ones. We find this to be the only way to obtain a reliable entropy estimate after post-processing. In cases in which the digitization and post-processing are combined it would of course be acceptable to place health tests afterwards, but this should not be considered the norm in the standard.

We recommend that the functions currently considered post-processing functions be considered non-vetted conditioning components, as the Von Neumann algorithm was in the previous draft; we also recommend explicitly stating that the use of more than one conditioning component is admissible, and that the given equations for entropy estimation apply to each step. We think there is a case to be made for not applying the 0.85 entropy estimation constant on those functions that permit an exact quantification of the output distribution function, provided the submitter is indeed able to give the exact input distribution function that should have been provided as part of the noise source model.

Post-processing: Von Neumann

Nowhere is it stated that the individual bits in a sample need be independent in the case of non-binary samples. It is, however, stated that Von Neumann is an allowed processing. This would (a) lead to raw data of a non-fixed size, and (b) possibly a deterioration of the output, as remarked in Appendix E. If the recommendations on the noise source model above are not accepted, this should be clarified.

### Post-processing: Linear filtering

In the same way as has been noted for the Von Neumann processing, linear filtering can lead to a deterioration of the quality of the output if applied to non-independent bits. By way of example it is enough to consider two identical bits. If the recommendations on the noise source model above are not accepted, the same warning as for the Von Neumann processing should be applied.

### Post-processing: runs method

If the recommendations on the noise source model above are not accepted, we would recommend more information on this be added, at least to the level of detail of the other two in the same category.

### Algorithm 6.4

We find the statement ‘Choose an output bit a from M such that no other bit in S is assumed **to contribute more entropy to the noise source samples** than a‘ (our emphasis) to be too ambiguous. Is the vendor meant to calculate the entropy of each individual bit on its own, or the difference between the joint entropy of the whole sample and the whole sample minus each single bit, or something else? This would not be a big issue if the use of this algorithm were not mandatory.

Considering the entropy of each bit on its own, this ranking algorithm works fine if each bit in the sample is independent of the others. If however there happen to be bits with a high individual entropy but a strong correlation, the joint entropy of the highest-ranking bits together may be lower than the joint entropy of bits with lower individual entropy. In order to output the m-tuple with the highest joint entropy one would have to evaluate the joint entropies of all possible combinations of m bits.

By way of example, consider a sample of three bits, (a,b,c), of which a has full entropy, b = a in all cases, and c has any other entropy less than one. Reducing the sample size from 3 to 2 with the given algorithm would be disadvantageous.

If the purpose of this test is to reduce the sample size to the largest value m that can be handled by the test, we recommend allowing the vendor to choose whatever approach they see fit to select the m-tuple they deem best, and then provide a full and proper justification for their reasoning. There is a good deal of leeway in the algorithm as defined already, hidden in the expression ‘assumed to contribute’. No rationale to provide a full ranking is readily apparent.

Entropy assessment constant

Line 604, footnote 2: citing ‘empirical studies’ without references makes it hard to judge how appropriate the coefficient really is. Please add references.

From: Surdhar, Pali
**Sent:** Monday, May 09, 2016 11:07 AM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | Thales e-Security | Pali Surdhar | G | | 142 | | When considering the actions to be taken when health tests raise an alarm, it is important to consider the following:<br>1. catastrophic failure<br>2. recoverable failure : can the system operate in degraded mode and what are the actions to remedy this state; this is particularly relevant in systems with multiple, validated h/w entropy sources where the failure of one is not necessarily catastrophic.<br>Is a separate threshold required for the above states?<br>How do we deal with this in a lights-out operation? (E.g. system is in a remote datacentre). Implication here is | |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  | to consider the use cases and how actions will impact. |  |
| 2 | Thales e-Security | Pali Surdhar | E |  | 338 |  | source is misspelt |  |
| 3 | Thales e-Security | Pali Surdhar | T |  | 349 |  | "If noise sources are independent, their entropy assessments can be added"<br>Comment required on acceptable ways to combine different sources ( implementation rather than the entropy assessment). Is there guidance for this? does adding the entropy assessments still work for the different ways of combining the sources? (XOR, concatenate, ...) |  |
| 4 | Thales e-Security | Pali Surdhar | T |  | 372 |  | Continuous testing may affect the entropy rate of a system. Can these be periodic rather than continuous? |  |
| 5 | Thales e-Security | Pali Surdhar | T |  | 407 |  | HeathTest(): Action on false?<br>any retry attempts allowed?<br>error thresholds? |  |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | Thales e-Security | Pali Surdhar | T | | 636 | | Does this take into account how the entropy sources are combined \|\| is not equal to ^.<br>Additionally, what happens in this instance if the rates that entropy is produced is measurably different between the two sources? Does this cover combining data at the slowest rate such that the device is discarding entropy from one source, or perhaps the combination is interleaved? | |
| 7 | Thales e-Security | Pali Surdhar | G | | 600 | | given the assurance of correct implementation by CAVP testing'Clarify if CAVP testing of the conditioning function is mandatory? Will it be treated as not vetted - i.e. in the same category as a wrongly implemented conditioning component. | |
| 8 | Thales e-Security | Pali Surdhar | G | | | | Will NIST be providing an entropy test suite? | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | Thales e-Security | Pali Surdhar | G | | | | Will NIST be providing a reference for health test implementation? How do you check that the health tests are implemented correctly? do these require code inspection or some CAVP type of test? | |
| 10 | Thales e-Security | Ignacio Dieguez | G | | | | How do we harmonise between the different standards and their test requirements - for instance can the online tests from AIS be combined with those from SP800-90B? | |

From: Schindler, Werner
Sent: Monday, May 09, 2016 12:25 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | BSI | Aron Gohr | ed | 1 | 275 | 1.1 | It is not completely clear what a „consistent“ source of entropy is. It would be useful to briefly define some minimal criteria for being a consistent source, e.g. near-stationarity or a near-iid property.. | Add a few words after „consistent source“ to explain what is expected at a minimum. |
| 2 | BSI | Aron Gohr | ed | 4 | 338 | 2.2.1 | „the noise sourse“ | Sourse → source |
| 3 | BSI | Aron Gohr | te | 9 | 452 | 3.1.1 | It is not obvious what the purpose of gathering validation data for a non-vetted conditioning component is. Such data can only reveal either implementation errors in the conditioning component (if it is compared to test vectors for the conditioning component under evaluation), or weaknesses in the conditioning component that are so severe that they can be found by a predefined set of statistical tests. | Define the purpose of gathering test data for the conditioning component. From a security point of view, it would in addition be very helpful if unvetted conditioning components had to be supported by a security argument showing that they will produce output indistinguishable from an ideal distribution if cryptographic standard assumptions hold (note that this standard is easily met by all the vetted conditioning components). |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | BSI | Werner Schindler | te, ed | | | Appendix B-Glossary | The glossary contains the definition of a non-deterministic Random Bit Generator (NRBG). However, the class of NRBGs falls into two natural subclasses, the class of physical RNGs and the class of non-physical non-deterministic Random Bit Generators. Both classes have very different features, and their security evaluation is rather different. Appropriate definitions should be added to the glossary.. | Proposed Definitions:<br>**Physical non-deterministic Random Bit Generator (PTRBG) (**or for short: **Physical Random Bit Generator):** The entropy source uses dedicated hardware or uses a physical experiment (noisy diode(s), oscillators, event sampling like radioactive decay etc.)<br>**Non-physical non-deterministic Random Bit Generator (NPTRBG):** The entropy source does not use dedicated hardware but uses system ressources (RAM content, thread number etc.) or the interaction of the user (time between keystrokes etc.) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | BSI | Werner Schindler | te, ed | | | Appendix B-Glossary | The glossary does not contain the term stochastic model. | Proposed definition: **Stochastic model:** A stochastic model is a mathematical description (of relevant properties) of a PTRBG using random variables, i.e., a model of the reality under certain conditions and limitations. A stochastic model used for the PTRBG analysis shall support the estimation of the entropy of the digitized data and finally of the raw data. In particular, it shall provide a family of distributions, which contains the true (but unknown) distribution of the digitized data or of the raw data. Moreover, the stochastic model should allow to understand the factors that may affect the entropy. The distribution of the PTRBG shall remain in the family of distributions, even if the quality of the digitized data goes down. |
| 6 | BSI | Werner Schindler | ed | | | | The document uses the terms 'digitized data' and 'raw data'. In the literature usually different terms are used. In particular, 'digitized data' are often denoted by 'raw random numbers' (e.g. in the ISO draft ISO /IEC WD 20543.2). The notation corresponding to 'raw data' in the sense of this document is not unique, e.g. 'internal random | It should be considered to at least change one of the terms 'digitized data' or 'raw data'; e.g. to ('digitized data' and 'internal random numbers'), ('raw data' and 'internal random numbers') or ('raw data' and 'postprocessed data') etc. |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  | numbers' or 'postprocessed data'. The used terms might lead to confusion. |  |
| 7 | BSI | Aron Gohr, Werner Schindler | te | 9 | 465 | 3.1.2 | The distinction between an IID track and a non-IID track does not seem to be appropriate. First of all, hardly any digitized data or even raw data will be IID in the strict mathematical sense of the word, and if they were, a rigorous proof would hardly be feasible. Moreover, from a practical point of view, one might speculate that the current design of the two tracks creates unintended incentive structures: a vendor who submits under the IID track has to do submit a rationale showing that their source is IID, but the only advantage this gives them over a competitor who submits an equivalent design under the non-IID track seems to be that their design has to undergo slightly less statistical tests, which are presumably associated with little cost to them. | Instead of disinguishing between an IID track and a non-IID track one should distinguish between physical RBGs and Non-physical non-deterministic RNGs since the evaluation of both types of RNGs are very different (see comments 9 and 10) |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 8 | BSI | Werner Schindler | te |  |  |  | In Comment 7 it is proposed to distinguish between physical RBGs and non-physical non-deterministic RBGs. If NIST follows this proposal the differences should be worked out. Finally, one is interested in the entropy per raw data bit. In fact, a lower entropy bound suffices, ideally one for all exemplars of the given RBG design under all allowed conditions of use and environmental conditions. In this comment we address Physical Random Bit Generators.<br><br>Any PTRBG design should allow to formulate a stochastic model, which in particular provides a family of probability distributions, which contains the true (but unknown) distribution.of the digitized data or raw data. Usually, this family depends on one or several parameters. (Due to tolerances of components and ageing effects exemplars of the same design may belong to different parameters.) After having specified this family of distributions one may estimate the parameters, which in turn yields an estimate of the | Additionally to 3.2.2. the documentation shall contain a stochastic model of the digitized data or of the raw data. In both cases a lower entropy bound for the raw data shall be derived. The stochastic model shall be supported by physical and engineering arguments and by experiments. The distribution of at least the digitized data shall be stationary. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | entropy. If the stochastic model considers the digitized data the post-processing shall be taken into account to obtain a lower entropy bound per raw data bit. The distribution of the digitized data shall be stationary since a verification of the stochastic model seems to be very difficult for non-stationary distributions.<br>The (blackbox) entropy estimators in Section 6 may be used to check the entropy claim, which is derived from the stochastic model. In fact, if any of these entropy estimators yields significant less entropy per bit (i.e., beyond 'usual' statistical deviations) than the stochastic model this is a serious indicator that the stochastic model is not valid. (If so desired I could provide one or two text proposals for stochastic models.)<br>NOTE: Comment 22 also refers to the blackbox entropy estimators. It in particular covers the case when no stochastic models are applied. | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | BSI | Werner Schindler | te | | | | In Comment 8 we addressed PTRBGs. In this comment we consider NPTRBGs. Unlike PTRBGs NPTRBGs usually do not allow a precise stochastic model. In many cases, the entropy source is not under the control of the designer (e.g., system data of a standard PC). The best one can do is to provide conservative entropy estimates for strings.of digitized data. NPTRBGs de facto always need considerable data compression. | Additionally to 3.2.2. the documentation shall contain a conservative entropy bound for the digitized data. The digitized data must be compressed at least by a factor such that the length of the resulting bit string is not larger than the established lower entropy bound.. |
| 10 | BSI | Aron Gohr | te | 11 | 526 | 3.1.4.2 | This criterion appears to be a bit arbitrary. If, for instance, H_I were half of H_r, should we have higher confidence in the result than if it were the other way around?<br>It is also worth noting that if H_r is half of H_I, then the probability of observing the initial dataset under the distribution implied by the row dataset will fall off exponentially with increasing dataset size. Since dataset size is here 1000000, this means that this is a very restrictive condition for rejecting that both datasets come essentially from the same sample. | It would seem safer for example to calculate confidence intervals around H_r, H_c and H_I to a confidence level of, say, 99 percent and rejecting if the intersection of these intervals is empty. |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 11 | BSI | Aron Gohr | ed | 11 | 538 | 3.1.4.3 | The use of „sample size“ to denote „alphabet size“ is, while it is explained in the terms and notations section of the document and consistently applied, somewhat confusing at least to me. In most of the statistical literature, sample size denotes the size of a set of observations and not the size of the space each observation may be sampled from (i.e. if I look at ten people for the purpose of e.g. polling statistics, my sample size is ten and not seven billion).. | Replace „sample size“ in the relevant places of the document with „alphabet size“. |
| 12 | BSI | Aron Gohr | ed | 11 | 544 | 3.1.4.3 | The percent sign in „(1-alpha)%“ effectively multiplies the confidence level by 1/100. | Remove percent sign. |
| 13 | BSI | Aron Gohr | te | 12 | 558 | 3.1.5 | One might want to emphasize again at this point that these are supposed to be independent noise sources. | Add a sentence to the effect that this of course assumes that there is a good reason to presume the noise sources independent. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 14 | BSI | Aron Gohr | te | 14 | 620 | 3.1.5.2 | Suppose we have an IID source with an unvetted conditioning component. Without analyzing the conditioning component, it is then dangerous to use the IID track on the output of the conditioning component, because it is entirely possible that an unvetted conditioning components may introduce (possibly very complicated) dependencies when processing IID input.<br><br>Note that problems of this kind can happen due to implementation errors, i.e. there is no need to assume that the vendor is not entirely acting in good faith. | From the point of view of security, the best way to ensure that a non-vetted conditioning component will produce cryptographically strong output is to demand that the vendor provide mathematical evidence that it will do so, i.e. a design rationale for the conditioning component that shows it secure under reasonable cryptographic assumptions. Testing can then address the issue of correctness of implementation and act as a sanity check of the design evaluation. |
| 15 | BSI | Aron Gohr | te | 15 | 650-653 | 3.2.1 | In the same general context, it would make sense to also require any self-protection measures and physical health tests to be documented. Also, a discussion of the effects of ageing on the entropy source and its operating conditions might be helpful in assessing any security claims. | Add remarks on these issues. |
| 16 | BSI | Aron Gohr | te | 16 | 704-708 | 3.2.2 | It would be helpful for evaluation purposes if a submitter using the non-IID path would have to submit evidence similar to the rationale needed in the IID path | See remark 14. |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 17 | BSI | Aron Gohr | te | 19 | 815-816 | 4.3 | It would make sense to explicitly include here failure modes due to an adversary. Otherwise, it is easy to read this as „failure modes due to natural causes“, which would be overly narrow for most cryptographic systems. | Explicitly include fault attacks as a possible cause of failure to be considered here. |
| 18 | BSI | Aron Gohr | te | 20 | 829-830 | 4.3 | In the case of intermittent failures, it would be helpful for security evaluation if also a rationale were submitted explaining why a particular failure mode might appear intermittently and why upon spontaneous recovery the noise source can still be relied upon. | Add a remark to the effect that failure modes that allow for spontaneous recovery have to be justified by supporting evidence of their intermittent nature. |
| 19 | BSI | Aron Gohr | te | 20 | 859-860 | 4.4 | This is in apparent contradiction to 815-816, where the vendor is required to include vendor-defined continuous tests to detect failure modes specific to the device under evaluation. | Add a statement saying that the tests required in 815-816 are required independently of 859-860. |
| 20 | BSI | Aron Gohr | ed | 25 | 995-996 | 5.1/Figure 4/2.2.2 | t_i = t_i' |  |
| 21 | BSI | Aron Gohr | te/ed | 32 | 1187-1188 | 5.2.1 | Isn't "for each pair" redundant here? The number of observed values for each bin does not depend on the (s_j, s_{j+1}) pairs except in the sense that we have to loop over them once in the counting process. |  |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 22 | BSI | Aron Gohr | te | 36 | 1318-1319 | 6.2 | All of the listed entropy estimation methods are generic methods agnostic to the workings of the noise source. If one of these methods produces an entropy estimate that is much lower than the others, it seems reasonable to worry that that method has discovered some regularity in the tested data but quite possibly not fully exploited it. From the security point of view, one would gain much higher assurance if the developer had to submit a rationale indicating why any of these tests is expected to fully exploit regularities in the random data produced. | Add a requirement to the effect that vendors have to show why they have confidence that the entropy estimates produced by the non-IID track tests will not overestimate the entropy of their source. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 23 | BSI | Aron Gohr | te | 55 | | Appendix B | The definition given of a confidence interval is arguably incorrect. Having a confidence interval to confidence level alpha around an observed distribution parameter does not mean that the actual parameter is with probability alpha in that interval, but that a distribution from the same family of distributions with the parameter to be estimated outside the interval will produce this parameter estimate or a more extreme one in a ratio of less than 1-alpha to all cases.<br>To give an admittedly trivial example, if one treats the output of an RBG with a vetted conditioning component as a B(1,p)-distributed random source and finds with a sample size of 1000000 that the value 0.5 for p is not in the confidence interval around the observed frequency to a confidence level of 0.95, then this is not evidence at all that the true value for p is not 0.5, because there is strong prior information available indicating that it is. | Amend the definition given. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 24 | BSI | Aron Gohr | ed | 57 | | Appendix B | From a strictly mathematical point of view the first sentence of the definition of independence seems to be wrong, because it omits the possibility that independent random variables can be linked on sets of zero measure (e.g. two real-valued random variables X and Y with continuous cumulative distribution functions can be independent even though for x a realisation of X and y a realisation of Y we may know a priori that if x is rational, then x=y).<br>Note that I do not disagree on this being a technicality with zero practical impact. | |
| 25 | BSI | Aron Gohr | te | 68 | | Appendix B | The definition of „sample size“ does not seem congruent with common usage. If I sample ten items out of an underlying set of one million, then my sample size is ten, not one million. | |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 26 | BSI | Werner Schindler | te |  |  | 3.1.5.1.2 | The entropy estimate is mulitplied by factor 0.85 to be on the safe side. However, if the entropy estimate is derived from a sound stochastic model (PTRBG evaluation) there is no need for a 'security factor' smaller than 1. For NPTRBGs the entropy estimate for the digitized data shall be conservative. Of course, one might demand a compression factor which exceeds the minimum value addressed in Comment 9, which would implicitly introduce a 'security factor'. | cancel the 'security factor' 0.85 |
| 27 | BSI | Werner Schindler | te |  |  | Section 4 | Subsection 4.3 demands that health tests shall be tailored to the RBG. For physical RNGs the stochastic model allows to specify appropriate health tests, which shall detect possible defects of the noise source. | For PTRBGs the stochastic model shall be used to tailor the health tests to the noise source and to justify their appropriateness. In particular the health test shall detect all cases of possible failures. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 28 | BSI | Werner Schindler | te | | | 6.3.4. | It is pointed out that Maurer's paper does not require any independence assumptions. However, Maurer assumes that the random source is binary-valued, stationary and ergodic with finite memory. Moreover, under these assumptions Maurer's test statistic is asymptotically related to the Shannon entropy. Further, [CoNa98] computes the correction factor c = c( b,\nu) more precisely; c.f. also the remarks concerning the entropy conjecture. A further paper of Coron might also be relevant [Coro99].<br>[CoNa98] J._S. Coron, D. Naccache: An accurate evaluation of Maurer's universal test. SAC 98, Springer, LNCS, Vol. 1556, Berlin, 1999, pp. 57-71.<br>[Coro99] J.-S. Coron: On the Security of Random Sources. PKC 1999, Springer, LNCS, Vol. 1560, Berlin, 1999, pp. 29-42. | It should be checked whether the mentioned papers are relevant for the entropy estimator in Subsection 6.3.4. |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 29 | BSI | Werner Schindler | te | | | 3.1.5.1.2. | If the entropy estimate is based on a sound stochastic model (PTRBG case), it should not be necessary to multiply by a 'security constant' 0.85.<br>Instead, in this case the constant might be set to 1. | |

From: Jonathan Smith
**Sent:** Monday, May 09, 2016 2:20 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | Cygnacom Solutions | Jonathan Smith | suggestion | 1 | 280 | Introduction | Add short "Testability" section to the intro to alert developers to access needed for data collection. I realize this is somewhat covered in the data collection section, but that's 17 pages in and a developer may well not look at it until ready to actually collect data; long after the chip has been designed and likely fabbed. | Altering chip, entropy source, or DRBG system designers as early and clearly as possible about where they'll need to pull samples from will save much pain down the road when they only discover during validation that they provided no test or debug mechanism to actually pull the necessary samples from before the optional conditioning function (or in the case where the entropy source w/o conditioning and consuming DRBG are on the same chip the pull samples before they're consumed by the DRBG) |
| 2 | Cygnacom Solutions | Jonathan Smith | suggestion | 10 | 499 | Restart Tests | Standard doesn't define "restart". For some entropy noise sources a "warm" restart may produce different results than allowing the system to shutdown, cool off, and have to power-up from an equilibrium state. | If real world start-up is the concern clarify that the restart test must simulate that. If either case is the concern consider requiring both "warm" and "cold" restart tests. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | Cygnacom Solutions | Jonathan Smith | correction | 18 | 781 | Types of Health Test | Current wording, implies a requirement to retain samples from start-up test until the test has completed; even if the intent is to then discard them. A potential future CMVP or CAVP validatior might feel the need to force a vendor to comply with this as written simply because its a written requirement in the standard. | Reword "until the tests are completed; after testing, these samples may simply be discarded." to "until the tests are complete; however these samples may be discarded at any time". |
| 4 | Cygnacom Solutions | Jonathan Smith | comment | 18 | 777 | Types of Health Test | As you may know ISO1970, which may form the basis of FIPS 140-3, largely moved away form power-up self-tests and instead requires that algorithms be conditionally tested prior to their first use. I think the current wording in the first sentence (and definition in the glossary on page 59) is broad enough not to be incompatible with this, but wanted to make sure you were alert to the change so edits would accidently force the entropy source to be tested instantly on module power-up even if the next 140 standard might otherwise permit delaying the test a bit if you didn't | Just keep future 140-3 or ISO19790 compatibility in mind |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  |  |

|  |  |  |
|---|---|---|
|  | actually need to use the entropy source yet. |  |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | Cygnacom Solutions | Jonathan Smith | suggestion | 17 | 745 | Requirements for Data Collection | This covered invasinve actions that might alter the behavior of the data source, but there's nothing that addresses general actions that might tend to do so. One example of a noise source used elsewhere was hard drive access times; if a naieve developer or tester wrote the samples to a file as they were captured that would alter the behavior of the drive access times. | Add a short section requiring an explanation of why the sampting method used isn't believed likely to interfere with the noise source. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | Cygnacom Solutions | Kirill Sinitski | | 17 | 719 | Requirements on Noise Source | Post-processing (specifically, unbiasing) can substantially reduce entropy (e.g. Von-Neumann and non-iid data) ; if we analyze just raw data our analysis might not correlate to collected entropy. At the same time, depending on post-processing technique used, the data might already be whitened and result in meaninglessly high min-entropy score. We can't call anything but actual direct source output "the raw data", and we shouldn't make any conclusions about the source unless direct output is tested. Failing to do this will result in inadequate entropy sources passing the test. To adequately determine available entropy both noise source and post-processing and conditioning techniques must be considered. | |
| 7 | Cygnacom Solutions | Kirill Sinitski | | 9 | 458 | Data Collection | Data Collection methodology is overly restrictive, there might not be 1000 consecutive samples available during restart. Instead, I suggest defining end of restart | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | sequence as the first instance of seeding. | |
| 8 | Cygnacom Solutions | Kirill Sinitski | | 9 | 459 | Data Collection | Creating M matrix is unnecessary overcomplicating this analysis; existing statistical analysis toolkits take sequential data set, by introducing this matric concept you will prevent their use and only introduce mistakes into process. | |
| 9 | Cygnacom Solutions | Kirill Sinitski | | 17 | 742 | Requirements on Data Collection | The requirement to always analyze raw data creates difficulties for embedded headless devices (e.g. switches, terminal servers) that rely on built-in CPU entropy source (e.g. Intel RdRand) that cannot be directly analyzed. Since such implementations represent majority of networking infrastructure, IoT, and SCADA solutions this requirement will preclude certification. Consequently, these devices won't get tested and certified and the vendors will keep relying on waivers to sell their uncertified and possibly flawed products. | |

From: Eckgren, Stephanie
Sent: Monday, May 09, 2016 2:32 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | InfoGard Laboratories | Joshua Hill | E | 2 | | 1.2 | The symbol L should be defined. | Add a definition for the symbol L (the sample size). |
| 2 | InfoGard Laboratories | Joshua Hill | G | 5 | 354 | 2.2.1 | The absence of specificity in the definition of "noise source" makes this poorly defined. It's not clear when multiple noise sources are in use, or when it is actually a single (more complicated) noise source. Common examples would be multiple ring oscillators which may or may not interact; are these multiple noise sources, or just one complicated noise source. | Define "noise source" and "multiple noise sources" so that the labs can distinguish between these two cases, or remove the requirements around multiple noise sources. |
| 3 | InfoGard Laboratories | Joshua Hill | E | 9 | 444 | 3.1.1 | Use of "shall" here only works if "consecutive" is removed, as this requirement is later softened. | Remove "consecutive" from this first statement, and add this requirement it in the next sentence. |
| 4 | InfoGard Laboratories | Joshua Hill | E | 9 | 445 | 3.1.1 | Add the "consecutive" requirement here. | Change the sentence starting with "If the generation of 1,000,000" to read "These 1,000,000 samples shall be consecutive outputs of the noise source, or, if the generation of 1,000,000 consecutive samples is not possible, …" |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | InfoGard Laboratories | Joshua Hill | G | 9 | 444, 458, 508 | 3.1.1, 3.1.4.1 | The phrase "directly from the noise source" suggests that this information should not be post-processed, but this is contradicted by the definition of "raw data" on line 340, the text of line 743, and the glossary entry for "raw data". It must be clear where various samples should be taken from. The phrase "directly from the noise source" surely sounds like prior to post-processing, but it isn't clear that this meaning was intended. | Make it clear where samples for entropy and statistical testing must be drawn from. |
| 6 | InfoGard Laboratories | Joshua Hill | E | 9 | 445 | 3.1.1 | "raw samples" is used where "raw data" was intended. "raw samples" isn't defined or used elsewhere in the document. | Replace "raw samples" with "raw data". |
| 7 | InfoGard Laboratories | Joshua Hill | E | 8 | 439 | 3.1 | Figure 2, the "Determining the track" reference should refer to 3.1.2, not 3.1.1. | Change the reference to 3.1.2. |
| 8 | InfoGard Laboratories | Joshua Hill | G | 8, 11 | 439, 519, 525, 535 | 3.1, 3.1.4.2, 3.1.4.3 | It seems like there should be some path forward in the instances currently described as "validation fails" and "no entropy estimate is awarded" (and similar variations). These instances do not generally signal a case where there is expected to be no entropy; rather, they are an indication that the assessed entropy should be (perhaps dramatically) lower than the current estimate. | One approach is to ask the vendor to reduce H_{submitter}, and try again. |
| 9 | InfoGard Laboratories | Joshua Hill | T | 11 | 538 | 3.1.4.3 | This statistical test appears to involve producing the upper bound of a confidence interval under the assumption that the underlying counts follow the binomial distribution with 1000 trials each (where the outcomes are the most likely symbol and then any other symbol), and failing in the instance where any of the 2000 observed data sets is above this bound. If that's not what's going on, then much of the rest of this comment won't make sense. It's not clear why this value for alpha has been selected. If we treat each test as | Assuming that I'm correctly interpreting the statistical test that is used here, we should set alpha to 1-(.99)^(1/2000). It is unclear if using the binomial distribution in this case is reasonable; I suggest running simulations to see how the binomial |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | independent, and want an overall probability of 0.01 of false reject, then we would want a 2000th root of an expression, not divide by 2000; we are effectively performing 2000 separate statistical tests (one per column, and one per row) with the one comparison. Treating the tests as independent, if alpha is to be the probability of false reject for one distinct test, then alpha = 1-(.99)^(1/2000). I also don't know why k is in this expression. If k=2, then each test is a true binomial distribution, so there should be no k term. Otherwise, we are really interested in the maximum of the multinomial distribution, not the binomial distribution (as we don't know a priori that the most commonly observed symbol for each test is actually the most likely symbol). I suspect that using the binomial distribution rather than the maximum of the multinomial distribution increases the probability of a false reject in many types of sources, but perhaps this is acceptable. For k>2, I don't know what meaning dividing by the number of symbols in the alphabet for the calculation of alpha; perhaps this is an approximation that I'm unfamiliar with... | distribution does for reasonable values of k. Alternately, for the distribution of the maximum of the multinomial distribution, see Corrado's 2001 paper "The exact distribution of the maximum, minimum, and the range of Multinomial/Dirichlet and Multivariate Hypergeometric frequencies", though this only will allow calculations with explicit parameters (many of which are unknown in this instance!). Approximations can also be found in Fuchs and Kenett's "A test for detecting outlying cells in the multinomial distribution and two-way contingency tables". |
| 10 | InfoGard Laboratories | Joshua Hill | T | 11 | 545 | 3.1.4.3 | Assuming that we are calculating the upper bound for a binomial distribution, then there is a typo in this calculation. The "1000" in the square root should be in a divisor. | U= 1000 p + Z_{1-\alpha} \sqrt{\frac{p (1-o)}{1000}} |
| 11 | InfoGard Laboratories | Joshua Hill | E | 11 | 545 | 3.1.4.3 | Z_{1-\alpha} is not defined in this document. | Define Z_{1-\alpha} as the (1-\alpha) quantile |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | of the standard normal distribution. |
| 12 | InfoGard Laboratories | Joshua Hill | G | 13 | 581 | 3.1.5.1.1 | This requirement seems to preclude the device from generating its own key, which is undesirable; some properties of the conditioners as entropy extractors require that the attacker not know the key, and it seems prudent to make this settable by the device. | Allow the keys used by conditioning functions to be generated by the device. |
| 13 | InfoGard Laboratories | Joshua Hill | T | 13 | 602 | 3.1.5.1.2 | The equation for h_{out} is not continuous (in h_in). This leads to some artificial behavior in the instance where h_in value can wander above and below this 2 times margin. | I suggest that this piecewise function be made continuous in h_in. (e.g., make a linear transition between the cases). This generation event should probably not be considered "key generation" with respect to FIPS 140 (to avoid circular requirements), so perhaps some other term should be used. |
| 14 | InfoGard Laboratories | Joshua Hill | G | 14 | 616 | 3.1.5.2 | Certain non-vetted conditioners can act poorly when the noise source data is not independent. | The vendor should only be able to use a non-vetted conditioning function if they can argue that, for the raw data source in use, there is a minimum min-entropy that can be output from the conditioning function (i.e., the vendor needs to argue that, for the particular |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  | noise source, the conditioning function won't result in the entropy output from the function being unboundedly low); this minimum min-entropy should be taken into account in the equation on line 627. |
| 15 | InfoGard Laboratories | Joshua Hill | E | 15 | 661 | 3.2.1 | Item #5 isn't a requirement. | Remove this item, or combine it with item 6. |
| 16 | InfoGard Laboratories | Joshua Hill | G | 3, 16 | 354, 678 | 2.2.1, 3.2.1 | In item #8, the requirement "only independent noise sources are allowed by this Recommendation" seems problematic. Any noise source (irrespective of its assessed independence) should be allowed to be integrated using an approved conditioning function. In this setting, inclusion of non-independent noise data shouldn't generally undermine the security of the noise data that one can credit as assessed entropy, and this additional (possibly non-independent) noise data has the possibility of adding significant strength (even if independence is difficult to formally justify). | Allow non-independent noise sources to be used, so long as they are combined a using approved conditioning function. If a noise source cannot be argued to be independent, assess it as providing no additional entropy. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 17 | InfoGard Laboratories | Joshua Hill | G | 17 | 717-722 | 3.2.2 | In the instance where the noise source isn't independent, some of the approved post-processing functions are inappropriate to use. Most of these post-processing systems reduce the entropy output from the noise source. | If post-processing functions are ultimately allowed, there should be some requirements surrounding the particular approaches adopted, e.g., "Von Neumann de-biasing shall only be allowed if the vendor has argued that the data input into the post-processing function is statistically independent." In addition, the vendor discussion of the entropy produced by the noise source ought to take the post-processing into account. |
| 18 | InfoGard Laboratories | Joshua Hill | G | 17 | 717-722 | 3.2.2 | Use of the cited post-processing functions is likely to obscure the statistical characteristics of the digitized noise source. | The vendor should be able to justify why the statistical testing performed in the continuous test and entropy assessment is still meaningful, even after the digitized noise source has passed through the post-processing function. In particular, the vendor ought to be required to argue that the |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  | continuous tests are likely to detect all expected failure modes, even after processing by the post-processing function. |
| 19 | InfoGard Laboratories | Joshua Hill | T | 20 | 844 | 4.3 | The continuous testing reduces entropy; when the false positive probability is low (e.g., 2^(-50)), this reduction is negligible. There is no upper limit on this probability, so this could be an issue if a vendor forms a test that is likely to fail. | Either account for this loss (e.g., adding log_2 (1-alpha) ), or (more reasonably) provide an upper bound for the probability of false reject along with the lower bound that is provided. |
| 20 | InfoGard Laboratories | Joshua Hill | G | 21 | 891, 927 | 4.4.1, 4.4.2 | The meaning of C in these two places is different; it is the first failing number in the RCT, and the last passing number in the APT. | It seems reasonable to make this consistent. Either make C the first failure value across the board, or make it the last passing value across the board. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 21 | InfoGard Laboratories | Joshua Hill | T | 25 | 996 | 5.1 | This permutation test doesn't have a consistent probability of false reject, and indeed such a probability can't be assured in the test style. In instances where the probability of false reject isn't the desired 0.1%, it is greater than this level in the test as described. This is particularly important when the number of distinct scores resulting from the statistical tests is small. In this instance, it may be preferable to sort the resulting scores (including the reference value), and consider this test as passed if the middle 9990 data elements (chop off the top 5 and bottom 5) contain the reference score. The down side of this approach is that the resulting chance of false reject is then less than or equal to 0.1%. (the currently specified approach and the approach outlined here are the two approaches nearest to a 0.1% false reject rate). | Either approach fails to yield a 0.1% chance of false reject, so it's just a matter of preference: do you want to error toward rejecting the hypothesis that the data set is IID, or toward accepting this hypothesis. |
| 22 | InfoGard Laboratories | Joshua Hill | T | 26 | 1015 | 5.1 | The described Fisher-Yates algorithm isn't quite correct; there should be some chance that a shuffled element remains fixed after shuffling, but this can't happen in the algorithm provided. (Look at the end points of the generated number in step 2a). | Change the range for 2a to "between 1 and i (inclusive)" |
| 23 | InfoGard Laboratories | Joshua Hill | E | 26 | 1019, 1024 | 5.1 | Both conversion I and conversion II should mention that they are padded with 0s. | Mention in Conversion I that the value is 0 padded when the last block has less than 8 bits. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 24 | InfoGard Laboratories | Joshua Hill | E | 37, 42, 45, 46, 48, 50 | 1339, 1342, 1364, 1512, 1591, 1628, 1676, 1726 | 6.3.1, 6.3.2, 6.3.4, 6.3.7, 6.3.8, 6.3.9, 6.3.10, | The numerical constant "2.576" is included; it would be better to instead reference the relevant Z_(1-.005} value. | Replace "2.576" with "Z_(1-.005}". |
| 25 | InfoGard Laboratories | Joshua Hill | E | 39 | 1397 | 6.3.3 | "output samples" suggests that "L", rather than "k". | Use "output symbols" instead. |
| 26 | InfoGard Laboratories | Joshua Hill | T | 39 | 1424 | 6.3.3 | The calculation for alpha here is not consistent with the Hagerty-Draper paper, which states that the overall probability (which we want to be .99) is as follows (notation updated for consistency): .99 = \alpha^{\min{k^2, d}}, where alpha is the bound necessary for application of Hoeffding's Inequality. Also, note that in proposition C.2, they are estimating only the k^2 elements in the transition matrix; we need to further estimate the additional k elements in the initial probabilities, so the appropriate term here is k^2 + k (note, the necessity of a change here is mentioned at the bottom of page 27 of Hagerty-Draper). Finally, I think that the inclusion of d in this statement is erroneous (though is included within Hagerty-Draper); I believe the underlying rationale for its presence is that the resulting product is well estimated when its terms have all been successfully bounded. This is true, however, the resulting product is not known a priori, it must be solved for using a dynamic programming algorithm which may select the incorrect maximal probability string in the | Set \alpha = (.99)^1/(k^2 + k), which behaves in a much more intuitive way than the formulation included in the current draft. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | instance where the other values are not well estimated; in particular, without successfully bounding all k^2 + k values, one does not satisfy the hypotheses of proposition C.1. | |
| 27 | InfoGard Laboratories | Joshua Hill | T | 40 | 1430, 1439 | 6.3.3 | These are both applications of Hoeffding's Inequality, which is in terms of the natural log (not log base 2). | Change both \log_2 instances to \log (or however the natural log is to be denoted). |
| 28 | InfoGard Laboratories | Joshua Hill | G | 52 | 1767 - 1776 | 6.4 | It isn't clear what the specification of this algorithm accomplishes; it's just a restatement of the more readable description the precedes this algorithm. The use of "rank" here is confusing (due to the mathematical sense of this word). | Remove the specification of this algorithm. If this algorithm is retained, please change "rank" to "ranking". |
| 29 | InfoGard Laboratories | Joshua Hill | G | 64 | 1873 - 1874 | Ap. F | The definition of narrowest internal width isn't clear. "Maximum amount of information" invites misreading, as it relies on a sort of information-theoretic view of this statement, which isn't likely to be common. | A definition along the lines of "The minimum, across all steps making up the conditioning function, of the number of bits of the state data that is dependent on the input to the function and influences the output of the function." |
| 30 | InfoGard Laboratories | Joshua Hill | G | 64 | 1864 - 1868 | Ap. E | The "linear filtering method" is very broadly specified; at present, this could possibly include XOR of some fixed number of outputs together (if iterative application of rules is allowed), multiplying by a matrix, application of a CRC, application of various algo-geometric codes, and processing through an LFSR. (As a note: this list could surely be much broader, but I listed only approaches that I've seen used). These major approaches | Either list major classes of this linear post-processing and include reasonable requirements for each, or rephrase the description of linear filtering so that such approaches are explicitly disallowed. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | should be separately listed so that relevant requirements can be included for each. | |
| 31 | InfoGard Laboratories | Joshua Hill | G | 37 | 1358 - 1359 | 6.3.2 | The failure if v<1000 should have some more explicit procedure associated. | "If v<1000, map down the noise source outputs by removing the lowest ranking bit (see Section 6.4), based on the ranking provided, and retest the data." |
| 32 | InfoGard Laboratories | Joshua Hill | G | | | | I appreciate you hard work on these documents, and suggest that you attempt to finalize these documents as soon as feasible. It is possible to dwell on both possible major additions and refinement of what is here, but various programs would benefit from speedy release of this document. | |

From: Chris Brych
**Sent:** Monday, May 09, 2016 4:45 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | Oracle | Dr. Paul Dale | | 9 | 444, 457 | 3.1.1 | In 3.1.1  the amount of data required to be collected is 1,000,000 samples (line 444) plus 1,000 times 1,000 samples (line 457) | This is twice what it was previously.  There is no provision for collecting fewer samples than this.  For a very slow source, this could take a very long time.  There is provision(line 447) for a source where you can't collect 1,000,000 samples in a sitting, to allow the concatenation of multiple samples of 1,000 – this would mean you could use the 1,000 times 1,000 samples as your 1,000,000 samples.  The main concern is the time it will take to collect the data. |
| 2 | Oracle | Scott Ellett | | 9 | 457 | 3.1.1 | | Will the environmental conditions (temperature, voltage, etc.) need to be kept constant for each restart? For a hardware entropy source, will a power cycle be required for each restart or will a reset without removing power be sufficient? |

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 2 | Oracle | Chris Brych |  | 15 | 654, 655 | 3.2.1 | "The entropy source shall have a well-defined (conceptual) security boundary, which should be the same as or be contained within a FIPS 140 cryptographic module boundary." This is not always the case where an entropy source will be contained within a FIPS 140 module boundary. FIPS has a concept of a logical cryptographic boundary for software modules which entropy is not included within this cryptographic boundary. | Suggest the following text: "The entropy source shall have a well-defined (conceptual) security boundary. If the entropy source lies outside of the computing platform, "for software cryptographic modules", the entropy shall use a trusted path when entered into the cryptographic module. If the entropy source lies within the computing platform, then no protection is required. If the entropy source lies outside of a hardware cryptographic boundary, the entropy shall use a trusted path when entered into the cryptographic module. |
| 3 | Oracle | Scott Ellett |  | 15 | 650 | 3.2.1 | Section 3.2.1 Item 3 says "Analysis of the entropy sources' behavior at the edges of these conditions (temperature range, voltages, etc. ) shall be documented...". | Will restart testing and entropy estimation need to be performed at various operating conditions? What kind of documentation will NIST be expecting to receive that shows the entropy source has acceptable behavior under various environmental conditions? |
| 4 | Oracle | Dr. Paul Dale |  | 18 | 774 | 4.2 | In 4.2 (line 777) start up tests are mandated. | It isn’t clear what start up tests should be used. The only reference to specific start up test is in line 847 which specifies the continuous test be run on a sample at start up. Some further clarification as to if this is sufficient or if more tests are mandated would be beneficial. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | Oracle | Dr. Paul Dale | | 20 | 847 | 4.3 | In section 4.3 (line 847) a mandated minimum of 4096 start-up samples is specified. | For very slow sources this could mean no entropy for hours.  The IO source on HP-UX would take about four hours to generate this many samples. |

From: John Liberty
**Sent:** Monday, May 09, 2016 5:18 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | IBM | J. Liberty | | 21 | 871 | 4.4 | Specifically allow a entropy source to claim a lower entropy then the design supports to reduce false negatives. For high performance/ high reliability systems, false negatives can lead to the unnecessary and costly replacement of hardware. | At the end of the line add" with respect to the claimed (Hsubmitter) entropy." |
| 2 | IBM | J. Liberty | | 22 | | 4.4.2 | For a noise source where the output width is wide (>> 16 bits), the description is not clear on how that should be tested. For such a large value, the adaptive proportion test as defined would be of little value. And only testing a fixed subset of the output could lead to missed failures. | Allow/require that for large widths that the data can be subsetted. And each subset be tested sequentially. For instance, a 16 bit value where each bit is IID, could be broken into 4 4bit subset. Each of these 4bit subsets data of the output stream could be tested sequentially, in a rotating fashion. That is test first 4 bits for a 512 samples run of the adaptive proportion test, then test the next 4 bits for 512 samples, and so forth. So after every 2048 samples all bits would get tested for this example. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | IBM | J. Liberty | | 64 | 1864 | App. E | Using Pilling Lemma, some bias for IID data can be reduced by applying an XOR between biased data. It can be eliminated when a bias value is XOR with a unbiased value. The ouput entropy cannot exceed the input entropy. | Explicitly allow XOR functions as type of Linear Filtering. |
| 4 | IBM | J Liberty | | 10 | 499 | 3.1.4 | For a complex system such as a processor that contains the entropy source, time from power on to the ability to access the power source can be measured in seconds or minutes. For the restart test, there will likely be a variation in time from power application to first read of data. | Make sure the standard explicately allows this variation in time from power on to first read. |
| 5 | IBM | J. Liberty | | 52 | 1756 | 6.4 | Taken literally, the maximum number of bits of entropy for a sample is restricted to the width of the narrowest test. Given a design that produces say 64 bits wide samples, where each bit is IID, this would throw away a significant amount of entropy. | For cases where the sample width is greater than the testable width, allow the samples to be broken up into subsets, where each subset is tested separately. That is, for a 64 sample where the test max is 16, break up the 64 bits into 4 16 bit samples and test each sample separately. The final entropy is the sum of entropy for each of the the 4 subsets combined. |

From: Colunga, Gerardo
**Sent:** Monday, May 09, 2016 9:02 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| 1 | HP Inc. | Gerardo Colunga | Q | 19 | 782 | 4.2 | Clarification.<br>Continuous tests are to be run indefinitely while the entropy source is running. Digitized output is requested from a noise source only when needed. Is it acceptable to only run continuous tests on digitized output when it is requested from the noise source? In other words it is not necessary to request digitized output from a noise source just to have continuous tests continuously running. | |
| 2 | HP Inc. | Gerardo Colunga | Q | 5 | 371 | 2.2.3 | Clarification.<br>This line implies that start-up tests are to be run "on all components". It is not clear what all of these components are. | Explicitly call out the components on which start-up tests are to be run. |
| 3 | HP Inc. | Gerardo Colunga | Q | 12 | 559 - 560 | 3.1.5 | Clarification<br>Submitter would have estimated hin by entropy assessment using 1 MB data set. Lines 559 - 560 seem to indicate hin needs to be calculated every time which is not possible during normal operation. Need clarification on determining hin for input to the conditioning component. | |
| 4 | HP Inc. | Gerardo Colunga | G | 15 | 646-647 | 3.2.1 | Need to expand on the documentation the developer is required to provide to justify why the entropy source can be relied upon to produce bits with entropy. | |
| 5 | HP Inc. | Gerardo Colunga | Q | 23 | 947 | 4.4.2 | Specification mentions that window size for startup tests is 4096 samples. Table 3 for non-binary data specifies cutoff value for window size 512 which is for continuous tests. What is the cutoff value for startup tests where window size is 4096? Need clarity. Is it that the startup tests needs to be run 8 times over a window size of 512 ? | |

From: Ronan Wallace
**Sent:** Friday, May 13, 2016 10:09 AM

I double-checked the numbers of Table 2 related to the calculation of the cutoff values C. My cross-checks are consistent with the published values for window sizes of 4096 and 65536, however I am calculating different numbers for window sizes of 256 and 64. For instance, with H=1 I calculate cutoff values of 177 and 55 instead of 168 and 51. Might be worth investigating.

From: Richard Moulds
**Sent:** Friday, May 13, 2016 2:50 PM

| # | Organization | Commentor | Type | Page # | Line # | Section | Comment(Include rationale for comment) | Suggested change |
|---|---|---|---|---|---|---|---|---|
| | Whitewood | Richard Moulds | T | | | 5 | Because the procedure for testing the IID assumption is formulated in terms of a dataset composed of a specific number of samples, the draft does not take into account the possibility of non-IID behaviors on different time scales in different noise sources. For example, 60Hz line voltage frequency may be evident in a 1-Mbit sample from a 1kbps noise source, but would likely not be apparent in a 1-Mbit sample from a 100Mbps source, owing to the different acquisition times. The IID testing method should also take into account the relevant physical time scales within and exterior to the source. | The procedure for testing the IID assumption should be modified to include a pre-test for correlations and periodicity with standard statistical tests, such as the serial autocorrelation test, and the spectral (DFT) test from SP800-22. To be relevant these would need to be performed on a larger sequential data set - 100 x 10Msamples, say - than for the existing IID tests. Only if these new tests are deemed to be passed would the process proceed to the existing IID testing method, using the 1 Msample data set.If the statistical testing failed, validation would proceed on the non-IID path. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Whitewood | Richard Moulds | T | | | | It would be useful for the end customer if the draft (and therefore future certifications) attempted to classify the nature of  the noise source being used and the presence of safety oriented conditioning functions. For example, FIPS 140 has defined 4 general levels of security for cryptographic modules and these have become widely recognized by end-users and have proven to be a useful guide in comparing products. If the 90B standard  makes no distinction between the various types of noise sources and safety features then the only quantitative measure available to end-users will be the measured entropy score, which is potentially misleading and only tells a part of the story. Imagine if car manufacturers competed purely on the basis of miles per gallon! Classifying every possible type of noise source source is not feasible but it should be possible to establish basic categories of operation. For example, one could draw the distinction between noise sources for which only a phenomenological randomness model may be possible (such as user mouse clicks) and those which have a rigorous physical model, such as thermal or quantum noise sources. Similarly it would be useful to capture in the | The 90b draft should recognize 3 levels of rigor and robustness in entropy sources. Level 1 (lowest level): noise sources, with or without conditioning, for which the submitter's entropy assessment is derived from a phenomenlogical model developed from observed source behavior; Level 2: noise sources for which the submitter's entropy assessment is anchored to a physical model of the source randomness (dynamical, chaotic, thermodynamic, optical, electrical or quantum physical); Level 3 (highest level): Level 2 with an approved conditioning function. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | certifcation if the prodiuct offers an approved cryptographic conditioning for fail-safe security. Without this form of classification, there is a disincentive for vendors to focus on providing an output of true uniformity and independence, and to incorporate conditioning for fail-safety which typically comes at the cost of a reduction in throughput of at least a factor of two. Without getting credit for making sound design decisions, vendors may be temped to focus exclusively on entorpy score and throughput - ultimately to the dis-service of the end-user. | |
| | Whitewood | Richard Moulds | E | 11 | 545 | | The symbol "Z" is not defined | |
| | Whitewood | Richard Moulds | G | 4 | | 2.2 | The terminology in Figure 1 is inconsistent with the terminology used elsewhere in the draft. "Noise source" is used in Figure 1 to refer to the analogue noise source, but in the body of the draft is used to mean the digitized and optionally-post-processed noise. | In Figure 1, replace "Noise source" with "Analogue noise source", and add the name "Digital noise source" to label the dotted line box. |
| | Whitewood | Richard Moulds | E | 10 | 495 | 3.1.3 | There is no "Requirement 8 in Section 3.2.2)" | Either correct text on line 495 or add Requirement 8 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Whitewood | Richard Moulds | T | 4 | 338 | 2.2.1 | The term "post-processing" should be restricted to mean digital signal processing, and not entropy extraction, which entails a reduction in the amount of data, often by post-selection. (See later comment for more on this point.) Modern signal processing techniques are performed in the digital, rather than the analogue, domain and enable higher performance at lower cost and lower power consumption. A particular digital signal processing method that should be approved is decorrelation by shift-and-XOR. This is widely used in TRNGs to remove correlation that may be introduced in the digitization process, and has a rigorous theoretical basis. | The "shift-and-XOR" decorrelator should be included as an approved post-processing function. In this method, each digitized noise bit, b[i], of a binary sequence {... b[2], b[1], b[0]} is XORed with a delayed noise bit, b[i+N], with an offset of a number of bit positions, N, selected to ensure independence between the direct and delayed bit sequences, to produce a post-processed bit, y[i] = b[i] XOR b[i+N]. (Instead of individual bits, blocks of bits may also be XORed.) It is the digital domain equivalent of combining two analogue signals from the same noise source. This procedure was analyzed theoretically by Vazirani, and shown to alleviate a variable next-bit-bias, which may even be under adversarial control, and has been evaluated experimentally. Variable next-bit-bias is a common feature in digitized analogue noise, owing to the finite slew-rate of electronic amplifiers. This post-processing function is |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | already in use in a number of vendors' True Random Number Generator (TRNG) products. It is described and analyzed in the German AIS31 standard. This effective decorrelation method allows vendors to provide a high performance/quality raw bit stream at lower cost and with lower power consumption than by increasing the performance (higher bandwidth and faster slew rates) for analogue and digital electronic components. |
| | Whitewood | Richard Moulds | G | | | | The terms “samples”, “symbols”, and “alphabet” are used without clear definition and inconsistently between sections 3, 5 and 6. | Clearly define "sample", "symbol" and "alphabet" and make usage of these terms consistent throughout the document. |
| | Whitewood | Richard Moulds | T | 20 | 844 | 4.3 | The proposal for a 2^-50 minimum Type 1 error probability would result in faster devices (higher output bit rates) requiring resets at shorter intervals than slower devices. This could be unacceptable in environments such as data centers or cloud providers. | We suggest scaling the minimum Type 1 error probability by the device's output bit rate, with a minimum probability of 2^-50 for a 1 Mbps device, and corresponding smaller probabilities for faster devices. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Whitewood | Richard Moulds | T | 13 | 602 | 3.1.5 | The proposed 0.85 safety factor leads to a discontinuity in the output entropy from the conditioning component as a function of the input entropy. For example, if the approved conditioning function SHA512 is used with an input entropy of 1,023 bits, the output entropy will be 0.85x512 ~ 435 bits per 512 output block. Whereas, if the input entropy is only one bit larger (1,024 bits), the output will have full entropy (512 bits). This discontinuity was discussed at the workshop and we understand the logic for proposing it. However, as written it will have an unwelcome side effect. The current draft creates a penalty for providers of premium entropy sources and may inadvertently motivate designers to make design choices that may be unnecessarily restrictive and against the interests of the customer and industry in general. The goal of most TRNG vendors is to create an entropy source that produces entropy at as high quality as possible, require as little conditioning as possible while always maintaining the highest possible throughput. Vendors that are able to create raw IID entropy with high entropy scores (i.e. >0.9 entropy per bit) will view conditioning functions such as | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SHA512 as a safety feature rather than an entropy enhancement tool. In which case they would design systems to incur the minimum throughput degradation caused by conditioning and rely on the raw quality of their source. In many cases the minimum realistic throughput degradation in the conditioning block will be a factor of two. As defined in the current draft, a vendor with a near perfect IID raw entropy source requiring minimal conditioning would be faced with a choice of declaring a lower entropy rate (reduced to 85%) than is really available in the system or to increase the degradation in throughput by reducing the output speed of the conditioning function relative to it's input (i.e. worse than a factor of two). Neither of these outcomes is in line with the customer's best interests. For example, consider a high quality noise source with 0.9 bits of entropy per bit, and the designer wishes to incorporate the safety benefits of SHA512 conditioning. It would be common design practice to use a 1,024 input block for the SHA512, resulting in a factor of 2 reduction in throughput. But with the current draft this would also result in a lower output entropy per bit (of | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0.85 bits) than the underlying noise source. The reduction in both throughput and claimed entropy would create a strong disincentive to build the best possible raw entropy source and incorporate the safety benefits of an approved conditioning stage and would prevent the vendor offering the customer the highest possible throughput performance. While the wording in the current draft is fine for non-IID sources and for systems that employ extensive entropy extraction and conditioning, it is ill-suited to products that place greater emphasis on the quality of the source and less emphasis on back-end conditioning. We strongly suggest that the standard should at least not penalize those vendors that have invested in developing high quality sources and should ideally provide them with an advantage. We would suggest adding a caveat to the standard that enables a system where it can be shown that if n_in is >= to 2 x n_out AND h_in (the entropy per bit of input) is >0.9 (or some other suitably high figure) then the output can be claimed to be full entropy. | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Whitewood | Richard Moulds | T | 12 | 408 | 3 | Entropy source validation is designed to evaluate the entropy coming out of a digitized noise source. As such it is necessary and most informative to evaluate the entropy before any entropy extractors are applied to the digitized data. Since the test point for raw data is defined as sitting between the post-processing function and the Conditioning function it is critical to clearly define these two types of functions. In our view, the post-processing function should be focused on removing unwanted artifacts that arise during the process of collecting and digitizing the analogue noise. In other words, post-processing is attmepting to expose the true nature of the noise source, free from artifacts that did not originate in the noise source itself. In general, post-processing would operate at line rate and not result in a reduction of data. The Conditioning function, on the other hand, should focus on improving the effective quality of the noise source itself. Anything that seeks to improve entropy should be peformed after the raw bits test. Both the von Neumann algorithm and the Linear Filtering algorithm are entropy extractors and change the entropy content of the data and | Change the allowed post processing algorithms to Shift-and-XOR and Length of runs. Designate the use of the von Neumann algorithm as Conditioning and only available to IID data. The Linear filtering or other algorithms that change the length of the data string should also be considered as Conditioning functions |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | should be correctly classified as examples of conditioning functions. Because the von Neumann algorithm should never be applied to non-IID data, it should not be applied until after the raw data test point and therefore can't be classified as a post-processing algorithm. Linear filtering is a post selection algorithm that also drastically changes the entropy content of data and again should be used only as a conditioning function. Shift-and-XOR is a decorrelation algorithm and thus fulfills the role of a post-processing function and can be safely applied before the IID-status of the digitized noise source has been determined. | |

From: Marco.Bucci
**Sent:** Wednesday, May 18, 2016 10:17 AM

Please find enclosed a short note in which I summarize the discussion we had at the workshop regarding the open points you mentioned.

As I already discussed with John, the main problem that our design faces with this standard draft is that the possible issues to be monitored by the health test have been already eliminated by design. Namely the noise source is, at least, as robust as any other digital device and **there is no way to produce any relevant entropy reduction except than causing a hard failure** (e.g. shortcutting or cutting a wire). As a result, a statistical health test cannot add any robustness, but can just produce fake faults due to the type I error. Beside the facts that the health test has a cost comparable with several noise sources, it is clear that using multiple sources would provide much more robustness than any statistical health test. This is the direction in which we would like to go in the next generation of RBG's. If you will look to the documents I sent you, I'm sure you will agree that this is a really big step with respect of the today's scenario.

I understand that the fact that a full-entropy source can be as robust as a pseudo-random generator, but more efficient, can be "shocking", but this is a result that come at the end of 20 years of work on this topic. I hope that there will be a way to leave the standard open to these "unusual" results.

I attach the document that I already gave John regarding the theoretical background of our design. This paper is already published by Springer. Lately we also got the production samples which have been tested under all the operation conditions. As expected the entropy rate is almost constant, but eventually, stressing temperature, the whole chip stop to work while the entropy source still operates correctly. This is not surprising considering that the noise source is a digital asynchronous circuit, which, intrinsically, is more robust than a digital synchronous circuit.

I also link a CHES paper where it is shown how, in the previous designs, we addressed the problem of on-line and off-line testability <https://www.iacr.org/archive/ches2005/011.pdf> .

Thanks in advance for your attention

Marco Bucci

# Some notes on the Open Questions at NIST Random Bit Generation Workshop May 2-3, 2016

**Marco Bucci**

## 1 Post-processing functions

### 1.1 Are they necessary?

A suitable post-processing allows the implementation of provable full entropy sources. The standard should not prevent this possibility.

### 1.2 What else should be included?

Any kind of post-processing should be allowed as far as it is provided the evidence that entropy evaluation can be performed on the post-processed data. Basically we need to prevent that postprocessing results in a pseudo-random generator that makes looking “good” a source which is not.

As an example, there are at least two of LFSR hashing post-processing methods which satisfies this requirement and allow the implementation of testable full entropy sources:

1. *Design of Testable Random Bit Generator*s (CHES 2005):
   Data are generated in bytes. Each byte is extracted from 8bit of a 32 bit LFSR which is reset at the beginning of each byte generation and then is fed with string of bits which is independent from the string used for the previous generation. As a result of the independence of the input strings and the LFSR reset, each generated byte is also independent and the delivered entropy can easily measured.
   NOTICE: the LFSR reset is needed just for entropy measurement purpose. Obviously the reset operation wastes entropy (which is therefore underestimated) and, since the LFSR operation is linear, it can be proven that the entropy without reset is, at least, the same than with reset (but actually pretty larger).
2. *A Fully-Digital Chaos-Based Random Bit Generator* (*The New Codebreakers*, Springer 2016):
   Data are generated in bit. Each bit is extracted from the decimation (let say a factor 4, 8 or 16, depending on the source) of a 32 bit LFSR featuring a primitive polynomial which is fed by the source. Entropy is estimated after a further post-processing performed by a Linear SR which is the corresponding self-synchronising “descrambler” of the previous one (see https://en.wikipedia.org/wiki/Scrambler). It can be seen that while the first LFSR, because of the decimation, performs an efficient hashing, the second Linear SR removes almost all the memory (i.e. the “pseudo-randomness”) introduced by the first

one (basically **the second Linear SR is a kind of “predictor”** of the first LFSR). As a result, the entropy of the output sequence can be measured using a conditional entropy test having a reasonable short memory (see Fig. 3).

**NOTE**
**This “non-pseudo-randomness” property of the post-processing can be easily assesses by verifying that, reducing the input entropy (e.g. in simulation) below a certain value, the postprocessed data do not look any more having maximal entropy (see Fig. 3).**

NOTE
It should be allowed to execute the **health test before post-processing** (see Fig. 1). The reason is that, typically, after post-processing the entropy assessment requires a too large amount of data and therefore it can be executed just off-line. However, the submitter can provide the evidence that, given a certain worst case for the entropy before post-processing, after post processing a certain amount of entropy is expected. Notice that this is also the approach adopted by the AIS 31 standard as *alternative criteria.*

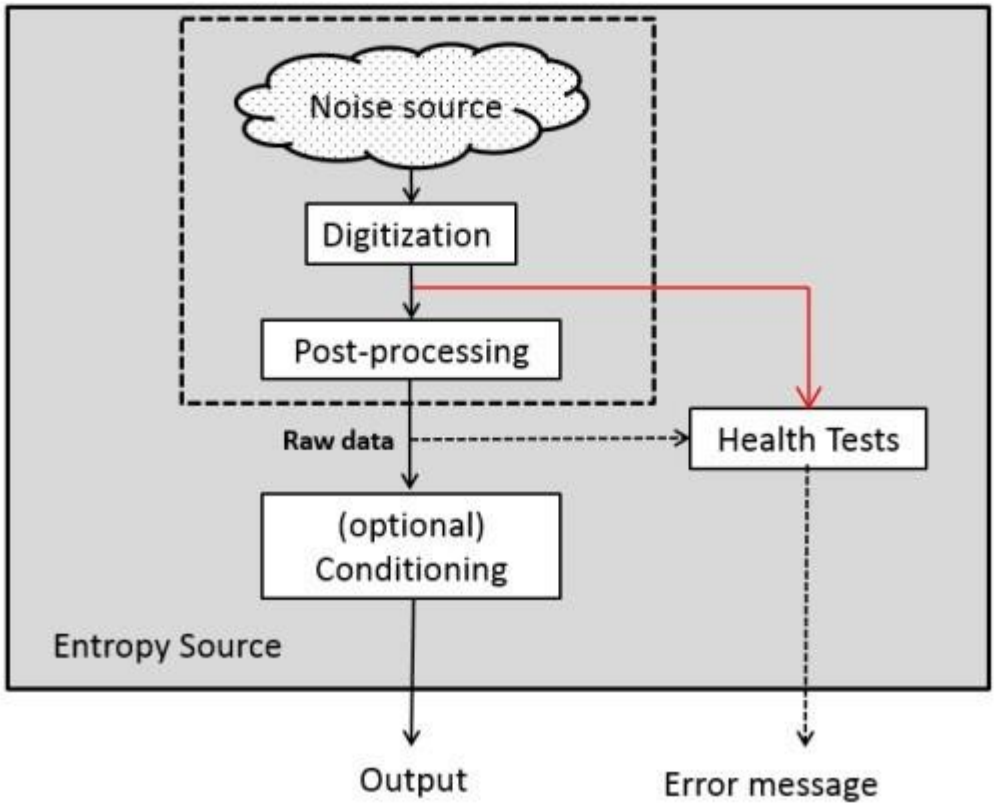

*Illustration 1: Health test should be allowed to be executed before of postprocessing provided that there are evidences regarding the entropy expected after post-processing when a certain amount of entropy is given on the input.*

### 1.3 What should be removed? -

## 2 Testing strategy

### 2.1 How we can improve this? Given constraints on cost of evaluation and lab resources?

As mentioned in Section 5.2.1, in case the symbol sequence has a limited memory, the straightforward and “standard” method for estimating entropy consists in using a **conditional entropy estimation**. The sample size should be derived by the memory length,

symbol size and the required estimation variance. Namely, **everything can be parametrised on the particular source and the particular entropy target** (e.g. sparse entropy or full entropy).
Regarding entropy estimation after post-processing, the required sample size is typically much larger (e.g. because of longer memory or because the target is full entropy and therefore a much smaller estimation variance is required). However, once the model of the source is validated, the effect of post-processing can be evaluated using worst-case simulation model of the source. In this way also several Gbit samples can be generated easily (see Fig. 3).

## 3 Health test

### 3.1 What actions should be taken when health tests raise an alarm?

This is a very crucial point which, unfortunately, shows how, besides low effectiveness and implementation difficulties, statistical health tests could have a doubtful utility.

Indeed, regarding the possible actions, we could consider the following options:

- **report the alarm, do not get the current data and rise a new data request:**
  here the assumption is that the health test is able to generate the alarm during the generation of each single data (otherwise discarding the current data does not make sense). **However this condition it not easy to be satisfied**.
  In practice, this solution is equivalent to have a variable generation time and, in facts, a variable compression ratio. The fault probability depends on how many successive alarms are allowed. Of course **the question regarding which action should be taken when all the attempts fail remains**.

- **reset the system:**
  here the hypothesis is that the consumer application can accept a random fault and recover from it. This could be a heavy requirement in some applications.
  However, in case the device would be under attack, as far as the alarm is not able to detect the anomalies before any data is delivered, the attack can be repeated after reset.

- **permanently block the system:**
  this is not acceptable in several applications as far as the probability of fault is not comparable with the probability of fault of any other part of the system.
  Notice that we could have a permanent block just because the system was temporary operating under wrong conditions.

- **permanently block the system after a certain number of alarms is occurred:**
  this is obviously equivalent to change the threshold of the test in order to get a negligible probability of false alarm.

- **replace the device:** this would imply some redundancy and an active action, eventually automatically executed by the system.

However, in most cases, no one of the previous solutions seems to be practical or effective. This leads to the conclusion that **the probability of false alarm must be made negligible** (see Section 3.2).

### 3. 2 The minimum allowed value of a type I error for health testing is selected as 2-50. Is this selection reasonable?

The **type I error of the health testing must be considered as a system fault** since, as already mentioned in 3.1, there is no practical action to recover from this condition. Although the effectiveness of the test must be ensured, **the standard cannot impose the implementation of a fault**. The designer must be free to reduce the type I error probability by "over-designing" the entropy source in order to deliver redundant entropy (see, as an example, presentation Section VI1). In facts, from the application point of view, **the type I error should not occur during the whole life of the system**. The redundant design needed to achieve this target is just a matter of availability/cost ratio as it is usual in the design of any high availability device.

### 3. 3. Issues with Health test

The standard should be open to alternative health test criteria suitable for the specific implementation of the noise source. The submitter should provide a convincing **fault analysis** and show how the proposed test covers the different possible faults that could occur.

This concept is already well enunciated in 4.6 (Pag. 24):

*The submitter can avoid the need to use the two approved continuous health tests by providing convincing evidence that the failure being considered will be reliably detected by the vendordefined continuous tests. This evidence may be a proof or the results of statistical simulations.*

However the criteria a) and b) should be removed since **statistical constraints are not necessarily the most efficient way to monitor the correct operation of the device** (and, in general, they are not).

Regarding the approved health tests (Repetition Count Test and Adaptive Proportion Test), although, in principle, these tests could be implemented, there are general issues regarding the effectiveness and efficiency of statistical tests, especially when they should be executed in real time and especially when the source is generic and not specifically designed to be testable.

Namely the main issues could be resumed in the following points:

- **ineffectiveness with statistically dependent sequences:**

in case of a sequence of non independent symbols, statistical tests became practically not feasible (too complex) or totally unreliable (entropy is largely over estimated).
As an example, sources based on ring oscillators can deliver almost periodical data, but, a simple test as a health test cannot distinguish such a quasi-deterministic behaviour from real entropy (i.e. the distribution looks perfectly “flat” as far as the test does not consider the dependency among a sufficient long sequence of symbols, see Section 5.3.1).

- **ineffectiveness regarding fake entropy:**
e.g. in sources based on ring oscillator, power supply noise results in a “deterministic jitter” which cannot be distinguished from random jitter (e.g. jitter due to thermal noise).

- **ineffectiveness regarding observation/manipulation attacks:**
no statistical test, regardless its complexity, can detect observation (probing) or manipulation (i.e. forcing a pseudo-random sequence) of the source. **Notice that this issues can be solved using multiple sources** (see Table 1).

- **too large complexity:**
the implementation of the health test can be much more complex than the implementation multiple sources (e.g. an asynchronous-digital chaotic source has a cost equivalent to about 10 DFF's). However, **the use of multiple sources provides a much better protection against faults and attacks** (see Table 1).

- **unacceptable delay before detecting anomalies (no data should be delivered in case of fault):** typically (except in some cases in which the source is designed on purpose to be testable) the health test can generate the alarm just after several output data are delivered. In facts, **this makes the health test practically useless**.

Of course, a FIFO could be used to store data till to the conclusion of the related health test. However, beyond the additional cost, this FIFO would result in a point of vulnerability. Notice that, **also in this case, multiple sources, offer better protection**: no low entropy data at all is delivered, except in case that all the source fail (see Table 1).

It is worth to mention that there are solutions that can mitigate the problems related to the statistical health tests or even be an alternative solution:

- **use a predictive model of the source:**
a model of the source is given in order to provide the best prediction of the next symbols depending on the previous ones. Then entropy is evaluated on the discrepancy between prediction and what is actually produced. As a result of the prediction model, entropy estimation can became much easier. **Basically, this is a kind of pre-processing before the health test**.

- **monitoring of system parameters:**

entropy is estimated according to a model (possibly a stochastic model) based on parameters which are monitored in real time (e.g. see Section VI-2 presentation). Monitoring system parameters can be more affective and prompter than estimating statistics. **False alarms cannot be generated**, moreover, if properly designed, the system does not deliver any low entropy data (i.e. **the alarm is generated before delivering low entropy data**).

- **use a robust source design and, possibly, redundancy (multiple sources):**
  the noise source is designed to be, at least, as robust as any other digital device (e.g. asynchronous-digital chaotic source). The **statistical health test is removed** (or replaced by a simpler “total-failure test”) since it cannot provide any additional robustness, but, on the contrary, due to type I errors, just spurious failures. As well as for any other digital device, **multiple sources can be used in order to increase availability** (see Section 4.1).

### 4. Multiple noise sources:

### 4.1 The Recommendation only allows using multiple noise sources if the noise sources are independent. Should the use of dependent noise sources also be allowed, and if so, how can we calculate an entropy assessment in this case?

Independence does not means that the sources must be based on different operation principle. Moreover, depending which is the purpose of having multiple sources, a different kind of independence is required.

We could distinguish two different cases:

- **increasing throughput:** multiple sources operate in parallel in order to deliver the required amount of entropy.
- **increasing availability:**
  multiple sources operate in parallel, in order to deliver the required amount of entropy even if just one (or a subset) of them is still operating correctly.

In the first case the problem is the assessment of the total entropy and therefore the matter is the statistical dependency among the delivered data.

Notice that, assuming two identical sources, evaluating dependency and total entropy is **just the same problem as evaluating IID or non IID hypothesis and entropy rate as in case of a single source**.

The second case is totally different because what is required is the **fault independence**, which, of course, implies a preliminary **fault analysis of the system**. Namely, the fault of one source (possibly due to attack or manipulation) should not result in the fault of another source (except in case of a total failure which prevent the system to deliver any data at all).

It is worthwhile to notice that, whenever there is evidence that the source implementation is as robust as the implementation of the health test (i.e. there is a negligible entropy variation over the whole operation conditions), **using multiple sources provides a better availability with respect of statistical health tests**.

In Table 1 shows a comparison between the availability provided by an entropy source using a single noise source plus a health test versus an entropy source using a double noise source without any health test. It can be seen that the second solution offers always a better fault coverage.

| Implementation | Availability Issue | | | |
|---|---|---|---|---|
| | single source fault | double fault | single source forcing/observing | false alarm |
| single source + health test | ✓ | ✗<br>(fault on both source and test) | ✗ | ✗ |
| | ✗<br>(low entropy data delivered till alarm assertion) | | | |
| 2 sources | ✓ | ✗<br>(fault on both sources) | ✓ | ✓ |

*Table 1: Comparison between the availability provided using a single noise source plus a health test versus a double noise source without any health test.*

## 5. Issues with our entropy estimator

### 5.1 IID vs non-IID Path

We must be aware that this distinction between IID and non-IID is not really significant regarding entropy assessment. First of all, while IID has a precise meaning, non-IID is a just too much generic definition and, in facts, the general case of non-IDD processes is just not approachable. If it was be addressable, this would obviously means that cryptography could not exist.

By the way, IID processes with a too large symbol space (e.g. 32 bit symbols) are not approachable as well.

However, in most of the cases, the symbols are, at least, **Identically Distributed** (since they derive from the same ergodic process), but depending on the model of the noise source, there are some kind of statistical dependencies which, typically, are **memory** (i.e. the current symbol depends on previous symbols with a dependency that, in most of the cases, decreases exponentially) and **periodicity**.

Periodicity, which is typical, for instance, of generator based on free running oscillators, is actually not easy to be addressed since, even in case of very low entropy, the frequency spectrum could be very complex (especially in case more than two oscillators are used).

On the other side, sequence memory can be addressed in a completely generic manner as long as it is sufficiently short. Indeed, **whichever is the statistical dependency with respect of the previous symbols**, the conditional entropy estimation $H(x_i/x_{i-1}, x_{i-2}, \ldots x_{i-L})$ returns the correct entropy rate of the sequence X as long as the memory of the sequence is shorter than L.

It is worthwhile to notice that since it holds

$$H(x_i/x_{i-1}, x_{i-2}, \ldots x_{i-L}) = H(x_i, x_{i-1}, x_{i-2}, \ldots x_{i-L}) - H(x_{i-1}, x_{i-2}, \ldots x_{i-L})$$

then, considering a non-I-ID (**non-Independent Identically Distributed**) sequence having an L long memory is practically equivalent to consider an IID sequence where the symbols are defined as the composition of L + 1 consecutive symbols of the original sequence.

In other words, more than on IID or on non-I-ID, the practical possibility to evaluate entropy depends on the quantity

$$N_{bit} = \log_2(SymbolSpaceSize) * (MemoryLength + 1) \ .$$

which, together with the required variance of the estimator, fixed the size of the sample needed for the evaluation.

We could even say that the IDD case is just a particular case of the non-I-ID one where the memory length L is zero.

**Definitely the limitation of the symbol space and the memory length is what makes the entropy possible to evaluate or not.**

### 5.1.1 Issues with IID test

**-**

## 5.2 Issues with non-IID test

### 5.2.1 Markov test has maximum space of 6 bits

The space of 6 bit for the Markov test is not sufficient for most of the applications (validation of a noise source model could need more than 20 bits). Moreover, the sequences to be evaluated are typically not a first-order Markov chain. Even if a higher-order Markov chain can be reduced to a first order one, this approach is not efficient. This is fact is obvious if, as an example, we consider a 6-order binary sequence. **The system is completely described by a** $2\cdot2^6$=128 **transition matrix, while, with the proposed method a** $2^6\cdot2^6$=4096 **matrix is needed**.

It should be clear that the **Markov test is completely equivalent to the evaluation of the conditional entropy** $H\,(x_i/x_{i-1}\,,\,x_{i-2},\ldots\,x_{i-L})$ (see Section 5.1). Indeed, the matrix describing the conditional probability of the next symbol $x_i$ with respect of the previous L symbols $x_{i-1},\,x_{i-2}\,,\ldots\,x_{i-L}$ , is nothing else than the transition matrix of the Markov system whose state is defined by the previous L symbols.

However using the conditional entropy approach is conceptually straightforward as well as more efficient and flexible.

It can be seen that conditional entropy estimation allows a quite precise evaluation of sparse entropy sources even **considering a memory length of more than 20 bits** (see Fig. 2). Indeed, when the entropy rate is far from the maximum, the estimator variance can be relaxed and there is no need of very large samples. In facts, the estimator is correct, results fit very well with the expected theoretical values (see Fig. 2) and **there is obviously no need to use min-entropy instead of the actual entropy**.

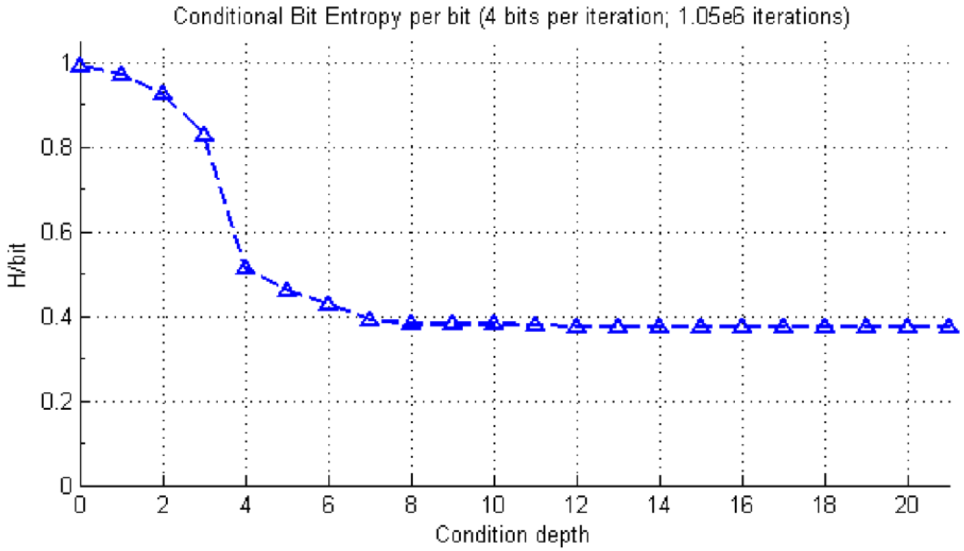

*Illustration 2: Conditional entropy estimation of a chaotic noise source. A sample of 4 million bit is more than sufficient to show that the entropy rate converges exactly to the expected value (in a chaotic source the entropy rate is equal to the Lyapunov exponent of the system).*

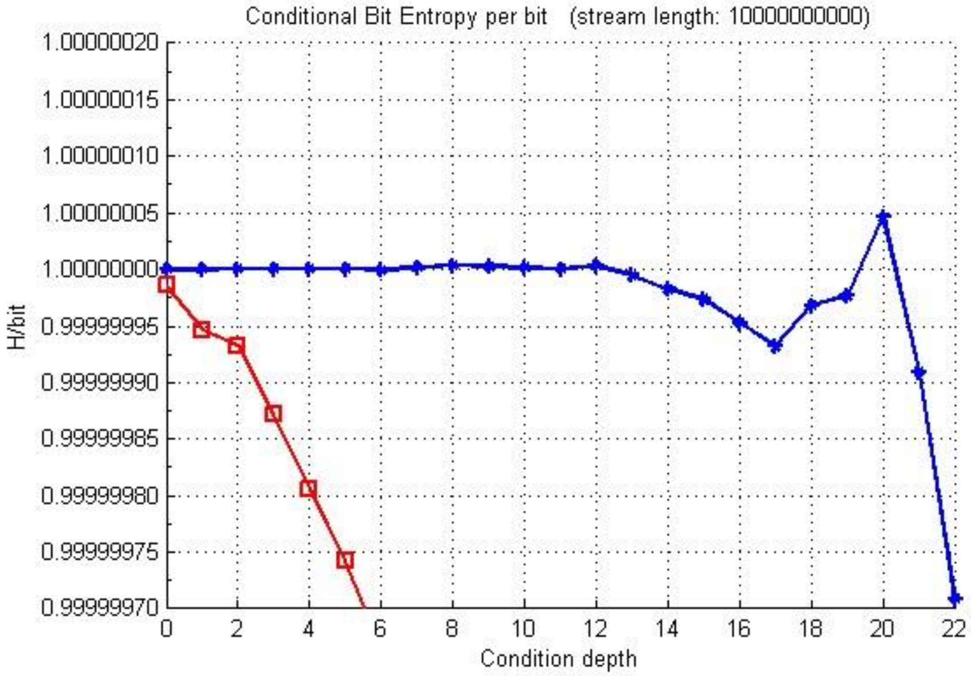

*Illustration 3: Conditional entropy estimation of a chaotic noise source after post processing (see Section 1.1). The star marked line (blue) corresponds to the source worst-case, while the square marked (red) line shows the effect when the source entropy is artificially reduced till to be not sufficient.*
*The very large input streams (10Gbit) are generated by simulation using a model of the noise source. The post-processing is provided of a descrambling transformation in order to feature a memory shorter than the test (condition depth). The entropy variation shown by the star marked (blue) line are due just to the variance of the estimator which increases over the condition depth. On the contrary, the square marked (red) line shows that a lack of input entropy can be actually detected by the test despite of the heavy post processing.*

Notice that this method can be also used for assessing that post-processed data are (practically) fullentropy. Of course, in this case the variance of the estimator must be very low (e.g. $10^{-6}$) and consequently a very large sample is needed (e.g. 10Gbit, see Fig. 3). However, this amount of data can be generated by simulation once the model of the noise source is provided and validated (as mentioned before, having sparse entropy, the evaluation of the noise source does not need very large samples and therefore, the noise model can be statistically validated).

### 5.2.2 Limits on possible entropy/sample (0.1-6 bit? 0.1-8bits)

**-**

## 5.3 General issues

### 5.3.1 We do not have great coverage of periodicity

Very likely periodical behaviours can be better investigated in the frequency domain, e.g. with an FFT or another similar transformation. How these tools could be used to get an entropy estimation is an interesting problem. Maybe the concept of **spectral entropy** should be investigated. Notice that a similar problem is approached in the field of audio compression algorithms where spectrum regularities are exploited in order to reduce the quantity of information needed to record an audio track.

Nevertheless, exactly because of such a difficult entropy evaluation, as far as it is possible, **periodical behaviours should be avoided by design** (design for testability).

Just as an example in Fig 4 and 5 the FFT of the same two oscillator source is shown. In Fig 4 jitter is 0 and therefore there is no entropy while in Fig. 5 jitter is about 10e-3. However a conditional entropy estimation with memory length 6 returns about the same value (about 0.25 bit entropy per bit).

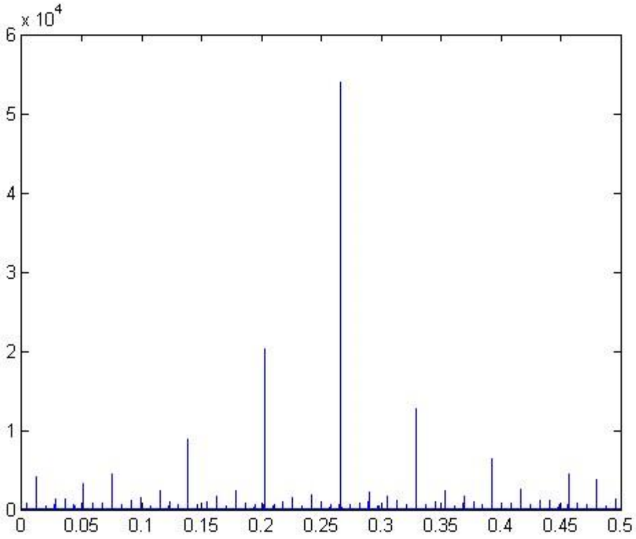

*Illustration 4: FFT of a two oscillator source with* $f_1$=1 , $f_2$≃51.73 *and no jitter.*

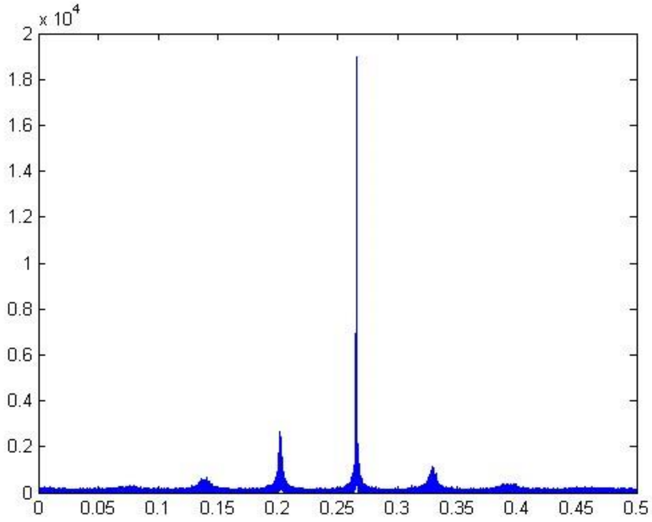

*Illustration 5: FFT of a two oscillator source with* $f_1 = 1, f_2 \simeq 51.73$ *and* $\sigma_{f1} \simeq 10e\text{-}3$ .

From: Marco.Bucci
**Sent:** Thursday, May 19, 2016 12:34 PM

Is it possible to have the formula for the “probability of detecting noise source failure”? I see only the formula for the “Cutoff Value”.

As far as I understand, there is something wrong in the Table 3: for H=1 the same probability 0.7 is given for both 50% and 33% entropy loss.

Anyway I’m trying to implement it on our entropy source, just to understand how much it can be effective. Actually, if I do not put some kind of filter to remove periodicity, the test is always passed regardless the entropy.

Thanks for your attention

Marco Bucci

From: Buller, Darryl M
Sent: Thursday, May 19, 2016 1:41 PM

# Comment on NIST SP 800-90B

**Summary**

This comment pertains to the 0.85 value used in SP 800-90B to account for collisions in the output space of a conditioning function. In short, this value appears to be conservative for realistic entropy sizes (i.e. 128, 256, 512 bits). We provide a method that approximates the min-entropy loss of a random variable when used as input to a conditioning function and also include some examples. We are presenting this method so that you can have the option (if desired) to experiment with various values in order to increase the granularity of the entropy accreditation function when the source is run through a conditioner.

**Method**

Consider the set of functions such that each function maps $m$ bits to $n$ bits. Let $f$ be a randomly chosen function from this set that is applied to an input space with $h$ bits of min-entropy, where $h \leq m$. We use a binomial distribution to determine the probability that $k$ inputs are mapped to a particular output value by supposing that there are effectively $2^h$ equally likely inputs, each having a probability of $2^{-n}$ of mapping to this particular output. Then the probability that this output is mapped to by $k$ inputs is

$$\binom{2^{\mathrm{h}}}{\mathrm{k}}\left(\frac{1}{2^{\mathrm{n}}}\right)^{\mathrm{k}}\left(1-\frac{1}{2^{\mathrm{n}}}\right)^{2^{\mathrm{h}}-\mathrm{k}}.$$

We estimate the min-entropy of the output space by finding the maximum value of $k$ such that there is only one expected output that is mapped to $k$ times. Since there are $2^n$ outputs, we want to solve for $k$ in the following equation:

$$2^n * \binom{2^{\mathrm{h}}}{\mathrm{k}}\left(\frac{1}{2^{\mathrm{n}}}\right)^{\mathrm{k}}\left(1-\frac{1}{2^{\mathrm{n}}}\right)^{2^{\mathrm{h}}-\mathrm{k}} = 1.$$

Note that this equation may have a second solution corresponding to the minimum value of $k$ that satisfies this equation.

In order to solve this equation, we use a Poisson approximation to the binomial distribution with mean $\mu = 2^{h-n}$, which yields

$$2^n * \frac{(2^{h-n})^k \, e^{-2^{h-n}}}{k!} = 1.$$

To estimate $k!$, we use Stirling's approximation $\sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ so that this equation becomes

$$2^n * \frac{(2^{h-n})^k \, e^{-2^{h-n}}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k} = 1 .$$

To simplify computations, we compute the log2 of both sides of this equation and rearrange the terms,

$$k(\log_2 e - \log_2 k + h - n) - \frac{1}{2}\log_2 k = (2^{h-n}) \log_2 e + \frac{1}{2}\log_2(2\pi) - n .$$

We use a binary search to solve this equation. In order to find the decreasing region of the curve

$$k(\log_2 e - \log_2 k + h - n) - \frac{1}{2}\log_2 k ,$$

which contains only the solution corresponding to the maximum value of $k$ that makes the expected count equal to one, we set the left bound of the binary search equal to $\mu$ (in the following paragraph we show that $\mu$ is slightly larger than the value $\hat{k}$ corresponding to the peak of this curve). Note that $\mu$ should be less than the desired value of $k$, and can therefore be used as the left bound. We then continually double this left bound until the curve is below

$$(2^{h-n}) \log_2 e + \frac{1}{2}\log_2(2\pi) - n ,$$

and use this value as the right bound.

We justify that $\mu$ is a valid left bound for the binary search by taking the derivative of this curve with respect to $k$ and solving for $\hat{k}$, the value at which the derivative is zero. This leads to

$$\hat{k}\left(2^{\frac{1}{2\ln(2)\hat{k}}}\right) = 2^{h-n} .$$

Since $2^{1/(2\ln(2)\hat{k})} > 1$, it follows that $2^{h-n} > \hat{k}$. Therefore, the curve is decreasing at $\mu = 2^{h-n}$, so we can use $2^{h-n}$ as the left bound. Note that if $2^{h-n}$ is large, the binary search may not exactly converge due to precision errors. In this case, we take the solution occurring after a certain number of iterations (~100,000).

Once the desired value of $k$ has been found by the binary search, we compute the min-entropy loss, with respect to

$n$, as

$$-\log_2\left(\frac{1}{2^n}\right)+\log_2\left(\frac{k/\mu}{2^n}\right)=n-h+\log_2 k.$$

Therefore, the remaining min-entropy is $n-(n-h+\log_2 k)=h-\log_2 k$ , and the percentage of min-entropy remaining is

$$100*\frac{h-\log_2 k}{\min(h,n)}.$$

Note that this percentage is relative to the maximum possible min-entropy (i.e., min $(h,n)$).

**Results**

We provide a table showing results of this approximation for various values of $h$ and $n$.

| $h$ | $n$ | Expected Max Value of $k$ | Remaining Min-Entropy | Remaining Min-Entropy (%) |
|---|---|---|---|---|
| 22 | 22 | 9.64 | 18.73 | 85.14 |
| 128 | 128 | 33.76 | 122.92 | 96.03 |
| 256 | 256 | 57.01 | 250.17 | 97.72 |
| 384 | 384 | 78.06 | 377.71 | 98.36 |
| 512 | 512 | 97.86 | 505.39 | 98.71 |
| 128 | 200 | 2.75 | 126.54 | 98.86 |
| 200 | 128 | 4.72e+21 | ~128.00 | ~100.00 |
| 128 | 160 | 4.80 | 125.74 | 98.23 |
| 160 | 128 | 4295779087.33 | ~128.00 | ~100.00 |

Note that slightly more entropy is lost when $h < n$ as opposed to $h > n$. This may be due to the *birthday paradox*, as the expected number of collisions when $h < n$ is greater than one would typically assume.