Comments on X9.82, Part 4: Constructions

From Elaine Barker

General Comments:

1. Don’t use the word “we” or other such personal pronouns in a standard.
2. Number the pages.
3. A number of concepts need to be both defined and discussed in layman’s terms early in the document:

   Computationally secure

   Information-theoretically secure

   Basic and enhanced NRBGs

   Composite mechanism

   DRBG mechanism

   Conditioning techniques

   Conditioned entropy sources

   External conditioning

   Uniformly distributed

   Composite RBG

   Persistent state

   Internal, cryptographic component

   Composite mechanism
4. Keep in mind that we want to post a version of Part 3 relatively soon. Too many changes will make this impossible.

Specific comments:

1. Lead in comment: Cryptanalysis discussions will just lead to too much controversy; limit this kind of discussion and put it in an annex.
2. Section 0.e: Why assign a target security level to a basic NRBG?
3. Section 1, paragraph 1, line 4: “random and pseudorandom”?
4. Section 1.4, para. 3: Are we going to allow software entropy sources?
5. Section 2.2.1, para.1: Put in Part 3.
6. Section 2.2.1.1: Only discuss for entropy sources; put in Part 2? Definition is in Part 3 currently.
7. Section 2.2.1.2: The specification will be in Part 3, so doesn’t need to be here.
8. Section 2.2.3: Define CRC.
9. Section 2.2.5, comment: Why buffer outside rather than inside? Does this imply a discussion in Part 2?
10. Section 2.2.5.1, para. 2: The term “accumulated across outputs” is not clear. Remove “incorrectly from line 2.

    Para. 4, line 3: Why must a queue expand? This seems to be an implementation rather than general guidance. How big is a queue entry?

11. Section 2.2.5.2, para. 4, lines 1 and 2: Is the new output always added to the oldest entry?

    Para. 5: Rewrite the first couple of sentences for clarity.

12. Section 2.2.5.3, para. 1: Need guidance on the size of R.

    Para. 2: Change "add" to "insert"? It might be useful to use consistent names throughout the standard; for example, the block output length could be *outlen.*

    Step c: Should this be "min" rather than "max"; If the sum of the *ee*'s <R, then this doesn't seem to work; is a period of initializing required?

    Steps a-e: Is this assuming full entropy?

13. Section 2.2.5.3.1, Title: Do you mean using the hash buffer using a hash function as a conditioning function for an entropy source?

    Item c: What are normal entropy values?

14. Section 2.3, para. 2: Make this a requirement rather than an assumption?

    Para. 2: We haven't decided to approve/accept software entropy sources; should go in Part 2, anyway.

15. Section 2.3.2: Is the use of an entropy pool a requirement or an example?

16. Section 2.4: Much of this seems to belong in Part 2.

17. Section 3.1, para. 1, line 3: The use of counter mode is an example.

    Para. 2, last 3 lines: This can only be done if prediction resistance is possible, and prediction resistance is not a requirement.

    Items a,b,c and the paragraphs before and after: What is the difference between a DRBG mechanism, a DRBG algorithm and a DRBG procedure (per Part 3)? Should a consuming application access the DRBG algorithm directly, or via the DRBG procedure where checks are made, entropy is acquired, etc? Note that Part 3 requires access via the procedure; when used as part of an enhanced NRBG, direct access may make sense. Also, these aren't the input parameters from either Part 1 or Part 3 (to the procedure).

18. Section 3.2, para. 2: Since prediction resistance is accomplished by reseeding, are there two or three places where entropy is obtained?

19. Section 3.2.1: Instantiate using the procedure and its parameters rather than directly calling the algorithm.

20. Section 3.2.1.1: How does this affect Part 3?

21. Section 3.2.1.2: Should this go in Part 3? Should it be pointed to from Part 3 for further guidance?

22. Section 3.2.1.3, para. 1: Do we really want to use should for the use of a seedfile? Note that the use of a seedfile is not currently addressed in Part 3.

    Item b: Part 3 doesn't address the oversampling issue. Part 3 allows the minimum size to be = *min_entropy*.

23. Section 3.2.2.1: A full entropy request would also require a reseed.

    Last para.: We need to be consistent in the use of terms: entropy input vs. entropy source vs. entropy input source.

24. Section 3.2.2.2: The seedfile concept is not currently in Part 3. Should it be called from the reseed procedure, the reseed algorithm, or from where?
25. Section 3.2.2.3, para. 2: "The DRBG mechanism collects and buffers entropy...". This seems to indicate that the mechanism contains the DRBG procedures and the entropy source. Is this correct?
26. Section 3.2.2.3.1: Is this the instantiate process or something in addition to the instantiate process?
27. Section 3.2.2.3.2: The use of the fast and slow pools should probably be a design example, not a requirement.
28. Section 3.2.3: Again, shouldn't the access to the DRBG algorithm be via the DRBG procedure?
29. Section 4: Need to put this in layman's English. Most readers will not be fluent in these terms.
30. Section 4.2.1.1: Need to be consistent with the definition/construction of a seed. In Part 3, a seed is constructed of entropy input and (possibly) a personalization string during instantiation. During reseeding, the seed is constructed from entropy input, a current state value and (possibley) additional input (though the text itself doesn't cover this as explicitly as it probably should).
31. Section 4.2.1.1.1, line 3: The term "internal state" is not used elsewhere in this way. The intended concept is not addressed properly anywhere yet. Part 3 has a "working state" which includes the *reseed_counter* and, for some DRBGs, other values critical to the DRBG; but this is more than what's intended here. If we are going to discuss this concept of the state, we need a unique, well-defined term.
32. Section 4.2.1.2: This seems to be an implementation rather than a general "concept".