Comments on ASC X9.82, Part 1 (dated May 14, 2004)

From Elaine Barker, NIST

1. The X9 committee and class of X9 standards should be "ASC X9" rather than "ANSI X9", where "ASC" stands for "American Standards Committee".
2. Section 4, DRBG, line 4, "do": I think this should be "may", since we discussed that providing sufficient prediction resistence could result in full entropy in the output. For example, if an output block yields 160 bits, and 160 bits of entropy are provided whenever 160 bits of output are produced, then the DRBG might be said to provide full entropy (I think).
3. Section 7.2, para. beginning "One allowed...": This seems to imply that using the result of the Peres unbiasing itself isn't allowed; my understanding is that it's OK.
4. Section 8.4, item 4, line 2: The protection boundary and the encapsulation in a FIPS 140-2 module need to be separated. For example, a DRBG that is implemented in a toolkit will not be in a FIPS 140-2 cryptomodule boundary, but a real implementation in s product should be.
5. Section 9.2, paragraphs 1, 2 and 4: There seem to be three requirements here that should be reflected in Section 10.2.
6. Section 9.5, sentence 2: What if the OGF uses a pointer into the state to select bits. Supposedly, the pointer is part of the state, but the OGF updates it. Suggest changing to something like "The ISTF sets and alters the internal state."
7. Section 9.6, 3rd para., sentence 3: This needs to be removed. If the OGF selects a subset of the bits in the pool, it's possible that the OGF can select several independent subsets before action by the ISTF is actually required. Either remove this sentence or change to something like "An ISTF **shall** either update the internal state between successive actions of the OGF, or the OGF **shall** select independent subsets of bits in the internal state without reusing any previously selected bits between updates of the internal state by the ISTF". Also, this would need to be a requirement in 10.5, 10.6 or both.
8. Section 9.6, 3rd para., sentence 4: This should be a requirement in Section 10.6.
9. Section 10.1, requirement 3: Remove this, as it is included in requirement 2.
10. Section 11.2.2: This is not the Get_entropy function used by the DRBGs; the parameters are wrong. If this is intended for DRBGs, then adapt the text in Section 9.5.2 of Part 3. Otherwise, is this to be used for the NRBGs? In this case, (1) is a handle needed? (2) does the NRBG use additional input? (3) why include a prediction resistance flag and a full entropy flag?
11. Section 11.2.3, last sentence: The tested bits may be used after testing is completed, in the case of an NRBG.

12. Section 12.2, last sentence: We decided not to provide statistical tests on the output, but only on the entropy input.

13. Section 12.4, item 4: Is this an enhanced NRBG?

14. Section 12.4, item 5: Change to something like the following: "An Enhanced NRBG with full algorithm independence: DRBG output is combined with the output of a Basic NRBG using a bitwise exclusive-OR operation. This provides full entropy for the output when the Basic NRBG is operating correctly; if the Basic NRBG fails (e.g., because of an entropy input source failure), the NRBG continues to operate as a DRBG at the security level provided by the DRBG, assuming that the DRBG has been successfully seeded for the security level."

15. Section 12.5: Add a Basic NRBG to the table.

16. Section 15.4: We need to be careful that we allow a DRBG implementation without a (current) entropy input source. In this case, the inclusion of the entropy input would be a "system" issue. The current text would seem to dictate that the entropy input source is "bound" with any product.