Issues on ANSI X9.82, Part 3

March 5-6, 2003

1. Section 3 (page 1): Do we want to keep the reference to DEA?
2. Section 3 (page 4): How should we reference the standards in the registry?
3. Section 7.1 (page 9): Is the model still OK?
4. Section 7.2.1, item 7 (page 12): At the September meeting, I was instructed to remove this item for reseeding. Do we still want to do this?
5. Section 9.1.1.2.4, step 5.1 (page 18): Should this step go inside or outside the loop. In the FIPS 186-2 change notice, this is inside the loop. However, it makes sense to place it outside the loop.
6. Sections 9.1.1.4, 9.1.2.4, 9.1.3.4, 9.1.4.4 and 9.2.1.4 (pages 19, 27, 32, 41 and 52): For each of the generators, a statement has been made about providing forward and backward secrecy. Are the statements correct?
7. Sections 9.1.1.5, 9.1.2.5, 9.1.3.5, 9.1.4.5 and 9.2.1.5 (pages 19, 27, 33, 41 and 53): What advice do we want to provide on reseeding each generator?
8. Section 9.1.2 (page 22): How do we ascertain that an application will be using the correct application-specific constant *t*?
9. Section 9.1.2.2.2, output (page 25): If we include types for the input and output parameters, what do we specify for the type of the state, which is often composite? Or should we list out all the information in the state?
10. Section 9.1.2.2.2, step 7 (page 25): Should the application-specific constant and the strength be included in the state?
11. Section 9.1.2.2.2, step 7 (page 25): Can we save the state internally to the process for comparison during reseeding?
12. Section 9.1.3 (page 28): The SHAKeylessHashDRBG was retained because of its current use. However, the keyed version is not currently being used. Do we want to retain it in X9.82?
13. Section 9.1.3.3, item 5 (page 32): Would it make sense to include *n*, *r* and *UserInput* in the state?
14. Section 9.1.4.2.2, step 2 (page 38): Do we assume that the fixed keys are already generated when the KeyedHashDRBG is initialized?
15. Section 9.2.1.1, User input text (page 43): How big should the user input be? The block size? The security strength?
16. Sections 9.2.1.2.2, 9.2.1.2.3 and 9.2.1.2.4 (pages 45, 47 and 50): At present, this generator is aimed at just TDEA and AES? Does it have to be more general?

17. Sections 9.2.1.2.5 and 9.2.1.2.6 (pages 51 and 52): TDEAkdf and AESkdf are essentially used as hash functions. Are there other suggestions besides the AES kwy wrapping algorithm?
18. Section 10.2 (page 56): Do we want to specify a FIPS 140-2 level?
19. Sections 10.3.3 and 10.3.6 (pages 57 and 58): FIPS 140-2 specifies the use of error detection codes (EDCs). Is this good enough for DRBGs?
20. Section 10.3.3 (page 57): If an EDC is used, is 16 bits the correct minimum length?
21. Section 10.3.7, para. 2 (page 58): Is 16 the right number?