May 1, 2007

Comment on X9.82-3

Lily.Chen@nist.gov

1. Page 1, Section 1, Scope, item 3, replace "that use" with "that are based on". (It can hardly say a DRBG Mechanism use a number theoretic problem.)
2. Section 4. Terms and Definitions. Some of the terms are also defined in part 1. Some of the definitions are not consistent with the definitions in part 1.
    a. Term 4.1, replace "that combines" with "that operates on".
    b. Term 4.6, entropy. This definition is not consistent with the definition in part 1. I think the definition in part 1 is more precise since in this standard, the entropy should be defined on a random variable.
    c. Term 4.18. This term is for RNG. But the definition is for RBG. Shall add something on relationship of RBG and RNG? It may not be obvious.
    d. Term 4.20. "replay protection" may not be provided by either encryption and integrity protection. It is why it is needed for replay protection and how to provide it. (In general, replay protection is only provided when re-using the previous authenticated data will help to get an entity authenticated. In order to provide replay protection, it will need at least one party who accepts the authentication data to provide fresh random data or they need to have a counter. It does not seem the case for the security channel between different components for the DRBG.)
    e. Term 4.21, in part 1, it was stated that the amount of work needed is $2^{security\_strength-1}$. Here it is $2^{security\_strength}$. Make them consistent.
3. Page 9, Section 4, Figure 1, shall error state has an arrow connected to Uninstantiate Function?
4. Page 10, section 7.2.4, first line, in the sentence, "The DRBG mechanism functions handle the DRBG's internal state", "handle" may be replaced by "transform" or "operate on".
5. Page 16, item 7, replace "an "extra strong" entropy input" with "an additional entropy input".
6. Page 18, section 8.6, In X9.82, prediction resistance is a requirement for reseeding. That is, prediction resistance implies that if one state is known, then it can only predict the output of a DRBG until it is reseeded. It cannot predict beyond the time of next reseeding. It is defined in part 1. But it seems misleading. It is actually very easy to be understood as a general concept for unpredictability not the specific one for this standard. (I cannot think anything better than this. Why not directly call it reseed required.)
7. Page 19, Section 9. The statement that "The DRBG mechanism functions in this Standard are specified as an algorithm (see Section 10) and an "envelope" of pseudocode around that algorithm (defined in this section)" is not helpful to explain DRBG mechanism functions. Maybe the first paragraph can be replaced by "The DRBG mechanism functions in this Standard can be specified by a series of pseudocode. For each function, a specific algorithm (see Section 10) is called to execute the transitions. A function need not be implemented using the same series of pseudocode. But the function shall have equivalent functionality."
8. Page 20, inside **Instantiate_function ()**, "*requested_instantiation, security_strength*" should be "*requested_instantiation_security_strength*" (No comma. All connected as one parameter).

9. Page 20, under **Instantiate_function**, the three items explain the three input parameters of the **Instantiate_function.** Therefore, “**Instantiate_function ()**:” should be replaced with “**Instantiate_function ()**, where”
10. Page 20, under **Instantiate_function**, item 2, second line, delete “the” in “a the”.
11. Page 20, the last paragraph. “2. nonce:” this indicates that nonce is not provided by the consuming applications. However, if nonce is included in the personalization string, then it may be provided by the consuming applications. (see page 17, section 8.5.2).
12. Page 22, Section 9.3, the first and the second bullet are equivalent. Please check and delete one of them.
13. Page 22, bottom line, replace “**Reseed_function ()**:” with “**Reseed_function ()**, where”
14. Page 24, Section 9.4.1, replace “**Generate_function ()**:” with “**Generate_function ()**, where”
15. Page 24, bottom line, replace “that are implemented to always support prediction resistance or that support prediction resistance do not require this parameter.” with “that are implemented to always support prediction resistance or that **never** support prediction resistance do not require this parameter.”
16. Page 29, Section 10.1, it has never been clear what is a DRBG mechanism so far in this standard. Maybe it will be helpful to add some sentences in the front like, “A DRBG mechanism is implemented through functions defined in Section 9, while each DRBG function employs an algorithm to execute the transitions. In this standard, a DRBG mechanism is named depends on a specific cryptographic primitive, which is used as a basic operations for the algorithm.” After this paragraph, probably it is ready to start the first paragraph in section 10.1 as “Several DRBG mechanisms are specified in this Standard. ….”
17. Page 29, Section 10.2.1 delete the first sentence. Or replace it with “A hash function DRBG mechanism is based on a hash function that is non-invertible or one-way”.
18. Page 29, Section 10.2, the second paragraph, it is suggested to eliminate the term “envelope” unless it is well defined somewhere. The last sentence of the second paragraph can be replaced by “Table 2 specifies values that shall be used for the DRBG mechanisms for each Approved hash function.”
19. Page 30, section 10.2.2.1, here, **HMAC_DRBG_Update** and **HMAC** are both called functions. Actually, they are functions from mathematics point of view. But in this standard, we use term function for Instantiate function, reseed function, generate function, etc. Can we use another term?
20. Page 30, section 10.2.2.1, the 1st paragraph, the last sentence can be replaced by “The strength of hash function used shall meet or exceed the security requirements of the consuming applications.”
21. Page 30, Figure 7, this figure caption is “HMAC_DRBG Generate Function”. But it actually contains instantiate function, generate function, and reseed function. Probably, it can be captioned as “HMAC_DRBG Mechanism” or simply “HMAC_DRBG” like Figure 10 on page 37.
22. Page 31, section 10.2.2.2.2, Replace “**HMAC_DRBG_Update()**:” with “**HMAC_DRBG_Update()**, where”.
23. Page 32, section 10.2.2.2.3, Replace “**HMAC_DRBG_Instantiate_Algorithm()**:” with “**HMAC_DRBG_Instantiate_Algorithm()**, where”.

24. Page 33, section 10.2.2.2.4, Replace "**HMAC_DRBG_Reseed_Algorithm():**" with "**HMAC_DRBG_Reseed_Algorithm()**, where".
25. Page 34, Replace "**HMAC_DRBG_Generate_Algorithm():**" with "**HMAC_DRBG_Generate_Algorithm()**, where".
26. Page 35, section 10.3.2.1. on the top of Table 3, replace the whole paragraph with "Table 3 specifies the values that shall be used for the block cipher DRBG mechanisms."
27. Page 36, second sentence in the first paragraph under the table cannot be understood.
28. Page 39, section 10.3.2.2.3.1, the title of this section is too long and unnecessary. Replace with "**Instantiation without Using a Derivation Function**".
29. Page 39, Section 10.3.2.2.3.1, replace "**CTR_DRBG_Instantiate_algorithm**():" with "**CTR_DRBG_Instantiate_algorithm**(), where"
30. Page 39, Section 10.3.2.2.3.1, under **CTR_DRBG_Instantiate_algorithm**()" make item 1 and 2 align with each other.
31. Page 40, section 10.3.2.2.3.1, the title of this section is too long and unnecessary. Replace with "**Instantiation with a Derivation Function**".
32. Page 40, Section 10.3.2.2.3.2, replace "**CTR_DRBG_Instantiate_algorithm**():" with "**CTR_DRBG_Instantiate_algorithm**(), where"
33. Page 40, Section 10.3.2.2.3.2, under **CTR_DRBG_Instantiate_algorithm**()" make item 3 align with item 1 and 2.
34. Page 41, Section 10.3.2.2.4.1. the title of this section is too long and unnecessary. Replace with **"Reseeding without Using a Derivation Function"**
35. Page 41, Section 10.3.2.2.4.1, replace "**CTR_DRBG_Reseed_algorith** ():" with "**CTR_DRBG_Reseed_algorith** (), where"
36. Page 41, Section 10.3.2.2.4.2, replace the title with "**Reseeding with a Derivation Function**".
37. Page 41, replace "**CTR_DRBG_Reseed_algorith** ():" with "**CTR_DRBG_Reseed_algorith** (), where"
38. Page 42, Section 10.3.2.2.5.1, replace the title with "**Generate Pseudorandom Bits with a Derivation Function**".
39. Page 42, Section 10.3.2.2.5.1, replace "**CTR_DRBG_Generate_algorithm** ():" with "**CTR_DRBG_Generate_algorithm** (), where"
40. Page 43, Section 10.3.2.2.5.1, replace the title with "**Generate Pseudorandom Bits without Using a Derivation Function**".
41. Page 43, Section 10.3.2.2.5.2, replace "**CTR_DRBG_Generate_algorithm** ():" with "**CTR_DRBG_Generate_algorithm** (), where"
42. Page 46, Section 10.4.2.1, the 2nd paragraph should include a sentence to highlight the basic operations for Dual_EC_DRBG. (The 1st paragraph only talks about ECDLP.) Perhaps something like "**Dual_EC_DRBG** uses elliptic curve scalar multiplication operation on two elliptic curve points P and Q to generate pseudorandom strings. Figure 11 depicts **DUAL_EC_DRBG**. The elliptic curve is defined in a finite field with the size approximately 2^m. For all the NIST curves given in this Standard for the DRBG, *m* is at least twice the *security_strength*, and never less than 256." After these sentences, probably it is the time to talk about an initial seed and seedlen.
43. Page 46, Section 10.4.2.1, the second paragraph, m may not be seedlen. Therefore, "m will be referred to as seedlen" may not be correct. Actually, m may not need to be referred as seedlen in this Standard.

44. Page 46-47, Section 10.4.2.1, it has spent large space to discuss “backtracking resistence”, which is neither necessary nor improper. (The other DRBG does not include these discussions. Why this one does?) It is suggested to shorten the backtracking resistance discussion. However, it should explain the two equations on the top of page 47. The Figure 12 should be used to illustrate the equations instead of the backtracking resistance.
45. Page 47, on the top of Table 4, replace “Table 4 specifies the values that **shall** be used for the envelope and algorithm for each curve.” with replace “Table 4 specifies the values that **shall** be used for the **DUAL_EC_DRBG** for each curve.”
46. Page 49, section 10.4.2.2.2, replace “**DUAL_EC_DRBG_Instantiate_algorithm** ():” with“**DUAL_EC_DRBG_Instantiate_algorithm** (), where”
47. Page 50, section 10.4.2.2.3, replace “**DUAL_EC_DRBG_Reseed_algorithm** ():” with“**DUAL_EC_DRBG_Reseed_algorithm** (), where”
48. Page 51, section 10.4.2.2.4, replace “**DUAL_EC_DRBG_Generate_algorithm** ():” with“**DUAL_EC_DRBG_Generate_algorithm** (), where”
49. Page 51, the **Truncate** function is defined only for DUA_EC_DRBG. Actually, it has been used for other DRBGs. Maybe it is not necessary to use this function. Furthermore, it does not seem a case that the input string is shorter than the requested number of bits.
50. Page 53, Section 10.5.2, replace “**Hash_df()**:” with “**Hash_df**(), where”
51. Page 54, Section 10.5.3, replace “**Block_Cipher_df()**:” with “**Block_Cipher_df**(), where”
52. Page 54, Section 10.5.3, the input parameter “no_of_bits_to_return” is misused in the process as “number_of_bits_to_return”. Please do a global search in this section to make it consistent.
53. Page 56, Section 10.5.4 replace “**BCC**():” with “**BCC**(), where”