Comments on Dual_EC_DRBG (March 2004 draft of X9.82, Part 3)

From Elaine Barker

1. The personalization string hasn't been added as a parameter to the instantiation routine. Will suggest fixes below. The string is to be combined with the entropy bits to derive the seed.
2. I suggest adding a table that provides guidance about using the DRBG (see below). Correct whatever is in error.

Curve	Maximum Security Strengths	Minimum Entropy Requirement	Seed length = *m*	Entropy Input Length (*m'*)	Block Size	Appropriate Hash Functions
B-163	80	128	163	168	144	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
K-163	80	128	163	168	144	
P-192	80	128	192	192	176	
P-224	112	128	224	224	208	SHA-224, SHA-256, SHA-384, SHA-512
B-233	112	128	233	240	216	
K-233	112	128	233	240	216	
P-256	128	128	256	256	240	SHA-256, SHA-384, SHA-512
B-283	128	128	283	288	264	
K-283	128	128	283	288	264	
P-384	192	192	384	384	368	SHA-384, SHA-512
B-409	192	192	409	416	392	
K-409	192	192	409	416	392	
P-521	256	256	521	528	504	SHA-512
B-571	256	256	571	576	552	
K-571	256	256	571	576	552	

3. Section 10.3.2.2.1, last two sentences: These could be omitted or changed to something like the following: "*personalization_string* is a string that will be concatenated to the entropy input returned from **Get_entropy (...)** to produce the seeding material that is used to derive the seed. A null string may...".
4. Section 10.3.2.2.3: Remove the comment at the end.

   I think that we decided to change the *additional_input_flag* to *additional_input* so that the additional input is passed to the routine.

5. Section 10.3.2.2.4: We agreed to remove the "insert additional input routine" section from the Hash_DRBG (which was done). However, if you want to retain the text, it could be used in Section 10.3.2.1 or (since the philosophy is appropriate to all the DRBGs) we could add, with appropriate generalizations, to Section 8 or 9.
6. If we decide that we need a method to remove an instantiation, a section in 10.3.2.2 will be needed for that. I'll be sending out the text for the KHF_DRBG and HMAC_DRBG soon, so you could get an idea what it might look like (assuming that I've done it reasonably.
7. As shown in the Hash_DRBG section, we'll need a self testing section in 10.3.2.2. However, I wouldn't bother to flesh a routine for that out in 10.3.2.3 until we decide how we really want to do it.
8. Section 10.3.2.3.1, item 6: Do we want to refer to FIPS 186-2 (which is real) or to the coming FIPS 186-3 (which I hope to publish for comment by this summer)?
9. Section 10.3.2.3.1, item 9: Change "seed" to "entropy input" (twice).
10. Section 10.3.2.3.1, *additional_input_flag*: I think that we decided to remove this flag.

    Get_entropy (...): Does the *min_length* have to equal the *max_length*, since the entropy input is concatenated with the personalization string and forced to the proper length by the Hash_df function?

    old_transformed_seed: Change to *old_transformed_entropy_input,* and change *seed_material* to *entropy_input*. The actual *seed_material* includes the *personalization_string*, and it's the *entropy_input* that we need to record.

    personalization_string: Change "byte array" to "string"?

    prediction_resistance_flag, next to last line: Place a comma before "it", and change "it" to "the flag".

    seed_material: Change to something like "The material used to derive the seed. The seed is then used to derive..."

    state: See the definition of state in the KHF_DRBG and HMAC_DRBG write ups. If this seems better, we should use it.

    transformed_seed: change to *transformed_entropy_input*. See *old_transformed_seed* above.
11. Section 10.3.2.2.2, instantiation routine, input string: copy the parameter list from Section 10.3.2.2.1, which contains the personalization string.

    Step 8: The table could be removed and replaced with a reference to the table provided in comment 2.

    Step 10: Could refer to the table (see comment 2) for the value of m'.

    Steps 10-13: There is a difference in naming convention with the hash-based DRBGs that we should probably rectify at some point.

    Step 12, response to RWK,s comment: As stated in the second paragraph of 8.5.1, we don't require full entropy from the entropy input source. This implies that we need to

Comment [ebb1]: This is just a difference in naming conventions that we need to rectify at some point.

Comment [ebb2]: This is just a difference in naming conventions that we need to rectify at some point.

Comment [ebb3]: This is just a difference in naming conventions that we need to rectify at some point.

ask for more bits than the entropy requirement, and that the returned bits may not have the entropy distributed evenly (maybe as bad as having all the real entropy at one end of the string and zeros to "pad" it out to the requested length). We just don't know what the entropy input source will provide, at this point. Therefore, the hash function is used to distribute the entropy properly. Does this help?

12. Section 10.3.2.2.3: steps 1 and 2: These are OK, but look at the KHF_DRBG and HMAC_DRBG write ups for a possible change.

    Many of the same comments as for Section 10.3.2.2.2.

13. Section 10.3.2.2.4, input: I thought that we had decided to replace the *additional_input_flag* with *additional_input*".

    Steps 1 and 2: These are OK, but look at the KHF_DRBG and HMAC_DRBG write ups for a possible change.

    Insert a step to check that if prediction resistance is requested and the *prediction_resistance_flag* in the state hasn't been set during instantiation, return an error. Step 5 could be modified accordingly (i.e., to remove the check for *state.prediction_resistance_flag*).

    Step 4: Needs to be changed to use the *additional_input* provided during the function call.

14. We may need to add sections on removing an instantiation and self testing (see comments 6 and 7.

15. Section 10.3.2.3.5: Need to consider whether we like the implementation "advice" at the end of each hash-based DRBG that indicates what could be omitted under certain conditions.

16. Need to decide what we want in Sections 10.3.2.3 and 10.3.2.4 sections so that the hash-based DRBGs can handle it the same way. Is this what we want?

Comments on MS_DRBG (March 2004 draft of X9.82, Part 3)

1. Section 10.3.3.1, line 1: Remove the comment and "so-called".
2. Suggest moving the table in Section 10.3.3.3.2, step 1 to Section 10.3.3.1 and add columns indicating the appropriate choices for the hash functions at each security level and the amount of required entopy, etc. See the following table as a suggestion. Does it make sense to add columns for *e*, *k* and *r*?

   Additional concern: How does John Stasak's strategy paper affect the use of this DRBG? I think he wants to cut RSA off at $n = 2048$?

Security Strength	**Minimum Entropy Requirement**	**Size of *n***	**No. of hard bits**	**Appropriate hash functions**
80	128	1024	10	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
112	128	2048	11	SHA-224, SHA-256, SHA-384, SHA-512
128	128	3072	11	SHA-256, SHA-384, SHA-512
192	192	7680	12	SHA-384, SHA-512
256	256	15360	13	SHA-512

   Page 138, para. beginning "Backtracking": Spell "resistance" correctly. The last sentence refers to Section 10.4.2.2.4. The text in that section could be placed here, or that text could be generalized and placed in Section 8 or 9.
3. Section 10.3.3.2.1: Remove the comment. Are there restrictions on the reseed_interval? If so, checks should be made on that.

   Last two sentences of the next to last para.: Change to something like : "*personalization_string* is a string that will be concatenated to the entropy input returned from **Get_entropy (...)** to produce the seeding material that is used to derive the seed. A null string may...".

   Last paragraph: Still need the material for C.4.
4. Section 10.3.3.2.2, 1st line: Remove "explicit".
5. Section 10.3.3.2.3: Remove the comment at the end.

   I think that we decided to change the *additional_input_flag* to *additional_input* so that the additional input is passed to the routine.
6. Section 10.3.3.2.4: We agreed to remove the "insert additional input routine" section from the Hash_DRBG (which was done). However, if you want to retain the text, it could be used in Section 10.3.3.1 or (since the philosophy is appropriate to all the DRBGs) we could add, with appropriate generalizations, to Section 8 or 9.

7. If we decide that we need a method to remove an instantiation, a section in 10.3.3.2 will be needed for that. I'll be sending out the text for the KHF_DRBG and HMAC_DRBG soon, so you could get an idea what it might look like (assuming that I've done it reasonably.
8. As shown in the Hash_DRBG section, we'll need a self testing section in 10.3.3.2. However, I wouldn't bother to flesh a routine for that out in 10.3.2.3 until we decide how we really want to do it.
9. Section 10.3.3.3.1: The list needs to be renumbered.

   Step 12: Capitalize "s". This is only initialized to the seed; it's not the seed after being updated, so suggest Changing to something like "The value S..."

   Step 14: This should be the strength instantiated for the state. See item 7 of 10.3.2.3.1.

   Step 18: Change *seed* to *entropy_input* (twice)?

   additional_input_flag: Remove this. See comment 5.

   Get_entropy (...), line 1: Remove "Approved". Does the value of *r* allow enough difference between the entropy requirement and the seed length requirement to allow for input that doesn't have full entropy. Also, does the *min_length* have to equal the *max_length*, since the entropy input is concatenated with the personalization string and forced to the proper length by the Hash_df function?

   old_transformed_seed: Change to *old_transformed_entropy_input*, and change *seed_material* to *entropy_input*. The actual *seed_material* includes the *personalization_string*, and it's the *entropy_input* that we need to record.

   personalization_string: Change "byte array" to "string"?

   prediction_resistance_flag, next to last line: Place a comma before "it", and change "it" to "the flag".

   seed_material: Change to something like "The material used to derive the seed. The seed is then used to derive..."

   state: See the definition of state in the KHF_DRBG and HMAC_DRBG write ups. If this seems better, we should use it.

   strength: This should be the record of the instantiated strength in the state.

   transformed_seed: change to *transformed_entropy_input*. See *old_transformed_seed* above.

10. Section 10.3.3.3.2, para. 2: Change to "...*n*, *r*, *e* and *k* **shall** be selected..."

    The table could be removed in favor of the table suggested in comment 2.

    Item 3, 4th line: The table specifies a particular number of bits, not a range of bits. Does think this sentence need to be changed? Fifth line: Place commas after "Thus" and "cases".

    Item 5: Should we openly allow moduli other than those specified in the table? For validation, we want to have an Approved moduli present (though others may also be

Comment [ebb4]: This is just a difference in naming conventions that we need to rectify at some point.

Comment [ebb5]: This is just a difference in naming conventions that we need to rectify at some point.

Comment [ebb6]: This is just a difference in naming conventions that we need to rectify at some point.

present). Also, for DSS, we are restricting the primes to being odd (talk to Rich Davis). I think this means that we shouldn't discuss private primes.

11. Section 10.3.3.3.3, input parameters: Add a *personalization_string* and remove the comment.

    Step 6: If a reseed_interval is provided, are there any restrictions that need to be checked?

    Step 7: I don't think that *min_entropy* has been defined yet.

    Steps 7-10: There is a difference in naming convention with the hash-based DRBGs that we should probably rectify at some point.

12. Section 10.3.3.2.3: steps 1 and 2: These are OK, but look at the KHF_DRBG and HMAC_DRBG write ups for a possible change.

    Many of the same comments as for Section 10.3.3.2.2.

    Remove the comment from step 3.3.

13. Section 10.3.3.2.4, input: I thought that we had decided to replace the *additional_input_flag* with *additional_input*".

    Steps 1 and 2: These are OK, but look at the KHF_DRBG and HMAC_DRBG write ups for a possible change.

    Renumber the steps.

    Insert a step to check that if prediction resistance is requested and the *prediction_resistance_flag* in the state hasn't been set during instantiation, return an error. Step 5 (the second one) could be modified accordingly (i.e., to remove the check for *state.prediction_resistance_flag*).

    Step 7: Needs to be changed to use the *additional_input* provided during the function call.

    Step 9: We need to define the >> operation (say, in Section 4 or 10.3.3.3.1). Then the comment can be removed.

14. We may need to add sections on removing an instantiation and self testing (see comments 7 and 8.

15. Need to consider whether we like the implementation "advice" at the end of each hash-based DRBG that indicates what could be omitted under certain conditions.

16. Need to decide what we want in Sections 10.3.2.3 and 10.3.2.4 sections so that the hash-based DRBGs can handle it the same way. Is this what we want?