10.1 Deterministic RBGs Based on Hash Functions

10.1.1 Discussion

The hash-based DRBGs specified in this Standard have been designed to use any Approved hash function and may be used by applications requiring various levels of security, providing that the appropriate hash function is used and sufficient entropy is obtained for the seed. The following are provided as DRBGs based on hash functions:

1. The **Hash_DRBG (...)** specified in Section 10.1.2.
2. The **KHF_DRBG (...)** specified in Section 10.1.3.
3. The **HMAC_DRBG (...)** specified in Section 10.1.4.

The maximum security level that could be supported by each hash function when used in a DRBG is equal to the number of bits in the hash function output block. However, this Standard supports only five security levels for DRBGs: 80, 112, 128, 192, and 256. Table 1 specifies the security strengths and entropy and seed length requirements that **shall** be used for each Approved hash function.

Table1: Security Strength, Entropy and Seed Length requirement for Each Hash Function

Hash Function	Security Strength	Required Minimum Entropy	Seed Length for Hash_DRBG	Seed Length for KHF_DRBG & HMAC_DRBG
SHA-1	80, 112	128	160-512	160 - 2^{32}
	128	128	192-512	160 - 2^{32}
SHA-224	80, 112, 128	128	224-512	224 - 2^{32}
	192	192	256-512	224 - 2^{32}
SHA-256	80, 112, 128	128	256-512	256 - 2^{32}
	192	192	256-512	256 - 2^{32}
	256	256	320-512	256 - 2^{32}
SHA-384	80, 112, 128	128	384-1024	384 - 2^{32}
	192	192	384-1024	384 - 2^{32}
	256	256	384-1024	384 - 2^{32}
SHA-512	80, 112, 128	128	512-1024	512 - 2^{32}
	192	192	512-1024	512 - 2^{32}
	256	256	512-1024	512 - 2^{32}

Questions re the DRBG Self Testing Process (Using Hash_DRBG as an example)

1. Should we test all processes implemented (i.e., instantiation, reseeding and generating bits)?
2. If instantiation is tested:
   - Should we use a special test usage class, or test each usage class implemented? Using a special test usage class would be simpler, but using the real ones would test that the proper value of *t* is used and that the initial state is set properly.
   - Should we test each of the security strengths implemented?
   - Should a special test flag be added to the **Get_entropy (...)** call? This would allow the insertion of a fixed value. Getting sufficient entropy to handle all multiple variations of the test would be tough. Does the DRBG need to test the entropy source?
3. If the reseeding process is tested, many of the same issues mentioned above apply.
4. For the Hash_DRBG (...) process:
   - Should different usage classes be tested?
   - Should different numbers of requested bits be used?
   - Should different strengths be tested (as appropriate for the implementation)?
   - What should we do about testing the prediction resistance aspect?
   - Should we test the reseeding process when the counter reaches the maximum number of updates?
5. Should subsets of these tests be performed, depending on whether the testing is done at power up or on demand?
6. Other issues?