10. Random Bit Generation Functional Requirements

For the purposes of this standard, there are (A) requirements, (B) features that are highly desirable but not required and (C) optional features. A requirement will always be stated using '**shall**' terminology. A feature that is highly desirable but not required will always be stated using '**should**' terminology. An optional feature is something that may be implemented to meet application requirements, but is not required by this Standard. An optional feature will be stated using '**may**' terminology.

An RBG conforming to the functional model specified above in Section 9 **shall** satisfy the following functional requirements in order to achieve the security properties specified above in Section 8.

10.1 General Discussion

The following functional requirements apply to all random bit generators:

1. The implementation **shall** be designed to allow validation testing; including specific design assertions about what the RBG is prohibited from doing. This **shall** include mechanisms for testing all possible error conditions.
2. The RBG **shall** be designed to meet the security requirements in Part 1, Section 8.

   Documentation requirement: There **shall** be design documentation that describes how the RBG is intended to meet all security requirements, including protection from misbehavior.
3. The RBG **shall** support backtracking resistance.

Comment [ebb1]: Page: 1
This requirement can be removed if Section 8.1.2 requires backtracking resistance (which it should). Then this requirement is subsumed by requirement 2 above.

Comment [ebb2]: Page: 1

Optional attributes for the functions in an RBG are as follows:

1. The RBG **may** be capable of supporting prediction resistance.

10.2 Entropy Input

The requirements for the entropy source of an RBG are:

1. The entropy input **shall** be based upon well-established physical principles or extensively characterized behavior.

   Documentation requirement: These principles **shall** be documented.
2. The entropy rate **shall** be assessable, or the collection **shall** be self-regulating, so that the amount of entropy per collection unit or event will reliably obtain or exceed a designed lower bound.

   Documentation requirement: This aspect of the design **shall** be documented.
3. The entropy input **shall** be designed so that direct manipulation (such as the ability to control the entropy input), predictable and controllable influence (such as the ability to bias entropy input), and direct observation of the entropy input can be prevented or, at least, detected.

   Documentation requirement: This aspect of the design **shall** be documented.

4. Loss or severe degradation of an entropy input **shall** be detectable.

   Documentation requirement: This aspect of the design **shall** be documented.

The optional feature for the entropy input is as follows:

1. The entropy input **may** be formed from multiple sources of entropy to improve resiliency to possible degradation or misbehavior. This can help meeting the requirement that the possibility of misbehavior is sufficiently small. It may be the case that an entropy source is already composed from multiple sources of entropy; this case is certainly allowed, although the entropy assessment of such an entropy source may be more complex.

10.3 Non-secret Inputs

There are no general requirements on Non-secret Inputs. A particular design may choose to incorporate personalization information such as a sequence number, a timestamp, or other non-secret information~~. However,~~ Part 2 (NRBGs) and part 3 (DRBGs) ~~**may**~~ levy requirements on Non-secret inputs that are specific to their use of these inputs. Non-secret inputs can be used as a way to address certain attack concerns.

10.4 Internal State

The requirements for the internal state of a RBG are:

1. The internal state **shall** be protected in a manner that is consistent with the use and sensitivity of the output.
2. The internal state **shall** be functionally maintained properly across power failures, reboots, etc. or regain a secure condition before any output is generated (i.e., either the integrity of the internal state **shall** be assured, or the internal state **shall** be re-initialized).
3. The RBG **shall** specify how it satisfies a particular security level in the internal state components.
4. ~~A DRBG instance **shall** have a specified finite cryptoperiod, after which the RBG **shall** either cease operation or have sufficient additional entropy added, as specified by the security level desired. The maximum cryptoperiod associated with a specific design is specified in Part 3. Operations **shall** cease after the specified cryptoperiod elapses. Note that it is good practice to combine the value of the old seed with the value of the new seed, so this specific activity is not prohibited, rather it is encouraged. The maximum cryptoperiod to be used for a particular instantiation **shall** be specified and documented; the actual cryptoperiod in use may be smaller than the maximum. A specific algorithm **shall** have a maximum cryptoperiod specified for it.~~

Comment [ebb3]: Page: 2
This is a Part 3 only requirement and doesn't belong in Part 1. In the case of the enhanced NRBG, this is a requirement on the Part 3 DRBG.

The optional feature for the internal state of an RBG is:

1. ~~The internal states used to produce public data such as nonces and initialization vectors **should** be fully independent from the states used to produce secret data such as cryptographic keys. There **may** be more separation of internal states to meet application requirements, such as separating the internal states for symmetric~~

~~keys from asymmetric keys, signature keys from key establishment keys, MAC keys from data encryption keys, etc. Examples of this separation are found in part 3.~~

Comment [ebb4]: Page: 3
This seems to be related to Part 3 only, so doesn't belong in Part 1.

10.5 Internal State Transition Functions

The requirements for the internal state transition functions of an RBG are:

1. The deterministic elements of internal state transition functions **shall** be verifiable via known-answer testing during the startup and periodic health tests.
2. The internal state transition function **shall**, over time, depend on all the entropy carried by the internal state. That is, added entropy **shall** affect the internal state.
3. The Internal State Transition Function **shall** resist observation and analysis via power consumption, timing, radiation emissions, or other side channels as appropriate, depending on the access by an observer who could be an adversary.

   Documentation requirement: This aspect of the design **shall** be documented.

   For example, if an adversary would have access to the power consumption, then ways to address this concern needs to be considered in the design and documented, but if an adversary would not have access to the power consumption, then this assumption **shall** be stated in the design documentation. Note that timing information may leak across communication networks, while power usage and radiation fluctuation almost certainly will not.

Comment [ebb5]: Page: 3
Do we need this?

4. It **shall not** be feasible (either intentionally or unintentionally) to cause the Internal State Transition Function to return to a prior state in normal operation (this excludes testing and authorized verification of the RBG output), except possibly by chance (depending on the specific design).

~~The optional feature of the internal state transition function is:~~

~~1. The Internal State Transition Function **may** enable the RBG to recover from the compromise of the internal state at a particular time through periodic incorporation of entropy appropriate for the desired security level. Note that an NRBG will always have this property.~~

Comment [ebb6]: Page: 3
This seems to be a DRBG only feature, since NRBGs already do this. Remove from Part 1.

10.6 Output Generation Function

The requirements for the output generation function are:

1. The output generation function **shall** be deterministic (given all inputs) and **shall** allow known-answer testing when requested.
2. The output **shall** be inhibited until the internal state exhibits/obtains sufficient assessed entropy.
3. Once a particular internal state has been used for output, the internal state **shall** be changed before more output is produced.

   ~~Documentation requirement: This aspect of the design **shall** be documented.~~

Comment [ebb7]: Page: 3
This is an algorithm requirement, so a documentation requirements doesn't make sense for a DRBG.

4. Test output from a known answer test **shall** be separated from operational output (e.g., random output that is used for a cryptographic purpose).

5. The output generation function **shall** protect the internal state, so that analysis of RBG outputs does not reveal useful information (from the point of view of compromise) about the internal state that could be used to reveal information about other outputs.

6. The output generation function **shall** use information from the internal state that contains sufficient entropy to support the required security level. ~~This aspect of the design shall be documented.~~

Comment [ebb8]: Page: 4
Doesn't make sense for DRBGs.

7. The output generation function **shall** resist observation and analysis via power consumption, timing, radiation emissions, or other side channels as appropriate.

   Documentation requirement: This aspect of the design **shall** be documented.

10.7 Support Functions

The requirements for support functions are:

1. An RBG **shall** be designed to permit testing that will ensure that the generator continues to operate correctly. These tests **shall** be performed at start-up, upon request and **may** also be performed periodically or continuously.

2. Output **shall** be inhibited during power-up, on-request and periodic testing until testing is complete and the result is acceptable. If the result is not acceptable, the RBG **shall** enter an error state.

3. Output need not be inhibited during continuous testing unless an unacceptable result is encountered. When an unacceptable result is thus determined, output **shall** be inhibited, and the RBG **shall** enter an error state.

4. When an RBG fails a test, the RBG **shall** enter an error state and output an error indicator. The RBG **shall not** perform any operations while in the error state. Parts 2 and 3 will address error recovery.

5. Any other support functions **shall** be documented.