Part 1 of this Standard provides general functional objectives and requirements for random bit generators. These objectives and requirements are specified below in italics, followed by a discussion about how they are satisfied by DRBGs in this part of the Standard.

The following testable functional requirements apply to all random bit generators:

1. *The implementation* ***shall*** *be able to be validated, including specific design assertions about what the RBG is not intended to do. Security-relevant branches that govern behavior in exceptional conditions (e.g. initialization, failed health tests, etc.)* ***shall*** *be verified by forcing all error conditions to occur during validation testing.*

   Implementation validation is discussed in Section 11. Test points have been included in the functional model (see Figure 1).

2. *The RBG* ***shall*** *satisfy all the appropriate top-level requirements, particularly the requirements for RBG output.*

   This has been addressed in Section 7.1.

Comment [ebb1]: Page: 32
This part of the requirement needs to be reworded or removed, since it appears to be a requirement of the validator.

The objectives for the functions in an RBG are as follows:

1. *There* ***must*** *be design evidence (theoretical, empirical, or both) to support all security requirements for the RBG, including protection from misbehavior.*

   This is provided by the design of the DRBG, as well as by the operational testing described in Section 11. When an implementation is validated, testing will provide further assurance that the DRBG will not misbehave.

2. *Depending on application requirements, the RBG* ***must*** *be capable of supporting forward and backward secrecy. Given anything that is meant to be observable about a RBG at a particular point in time, it* ***must*** *be infeasible to compute or predict any future or prior output bit.*

   Forward and backward secrecy has been designed into the DRBGs specified herein when observed from outside the DRBG boundary. When an implementation is validated, design evidence **shall** be provided and testing will be conducted to provide assurance that information that is internal to the DRBG is not observable from outside the DRBG boundary.

8.2 DRBG Components of the Model and Functional Objectives and Requirements for Each Component

8.2.1 Entropy Source

The entropy source is the source of digitized bits for the DRBG. This source **shall** be either an Approved NRBG as specified in Part 2 of this Standard, or an Approved DRBG or chain of DRBGs in which the first DRBG in the chain has an Approved NRBG as an entropy source.

The DRBGs specified in this Standard allow for some bias in the entropy source. Whenever a source of entropy is required by the DRBG, the minimum entropy that **shall** be provided by the entropy source is indicated. Depending on the entropy source, additional redundant bits may also be provided.

The primary use of the entropy source for DRBGs, is the acquisition of initializing inputs called seeds. These seeds **shall** be obtained prior to requesting pseudorandom bits from the DRBG. Additional entropy **may** also be introduced during the operation of the DRBG. An example of this might be information that is entered by a user. Further discussions on seeds and user input are provided in Section 9.

Part 1 of this Standard provides functional objectives and requirements for the entropy sources for random bit generators. These objectives and requirements are specified below. They are met when an entropy source that conforms to Part 2 of this Standard is used, and the interface between the entropy source and the DRBG is protected against influence, manipulation and observation.

The requirements for the entropy source of an RBG are:

1. *The entropy source* ***shall*** *be based upon well-established physical principles, or extensively characterized behavior.*
2. *The entropy rate* ***shall*** *be assessable, or the collection self-regulating, so that the amount of entropy per collection unit or event will reliably obtain or exceed a designed lower bound.*

The objectives for the entropy source of an RBG are:

1. *The RBG* ***must*** *be free from predictable and controllable influence, manipulation, and observation.*
2. *Loss or severe degradation of an entropy source* ***must*** *be detectable.*
3. *The entropy source* ***may*** *be formed from multiple sources of random bits.*

8.2.2 Other Input Information

Other information is required by a DRBG as input during the generation process. This information includes the input parameters when the DRBG is called by the consuming application. Input information **shall** be checked for validity when possible.

Depending on the DRBG, time variant information may also be required, e.g., a counter or a date/time value.

A counter used by a deterministic RBG **shall not** repeat during the "instance" of the deterministic RBG. When the deterministic RBG is initialized with a new seed, the counter **may** be set to a fixed value (e.g., set to 1), but **shall** be updated for each state of the DRBG and **shall not** repeat.

A date/time value used by a deterministic RBG **shall not** repeat; a different date/time value **shall** be used whenever a date/time value is requested by the DRBG or another technique **shall** supplement the date/time value to provide uniqueness.

8.2.3 Internal State

The internal state is the memory of the DRBG and consists of all of the parameters, variables and other stored values that the DRBG uses or acts upon. The internal state includes values that are acted upon by the internal state transition function between requests, keys used during each call, user input that has been obtained while a request is serviced, and time-variant parameters used by the DRBG. The internal state is dependent on the specific DRBG and includes all information that is required to produce the pseudorandom bits from one request to the next. Some portion of the internal state **shall** be changed by the internal state transition function at each iteration of the DRBG.

Part 1 of this Standard provides objectives and requirements on the internal state of a random bit generator. These objectives and requirements are specified below in italics, followed by a discussion about how they are satisfied by this part of the Standard.

The requirements for the internal state of a RBG are:

1. *The internal state **shall** be protected in a manner consistent with the use and sensitivity of the output.*

   This requirement is fulfilled by an implementation that conforms with this part of the Standard, particularly if implemented within an appropriate level of FIPS 140-2 cryptographic module.

2. *The internal state **shall** be functionally maintained properly across power failures, reboots, etc. or regain a secure condition quickly (i.e., either the integrity of the internal state **shall** be assured, or the internal state **shall** be re-initialized).*

   The fulfillment of this requirement is dependent on the physical embodiment of the DRBG and how it has been designed. When a DRBG is validated, design evidence **shall** be provided and testing will be conducted to establish that this requirement is fulfilled.

3. *The state elements that accumulate or carry entropy for the RBG **shall** be at least 2x, where x is the desired cryptographic strength expressed in bits of security. (x bits of security means that it takes about 2^x operations to attack the cryptographic system.*

   This requirement is inherent in the specification of the DRBGs herein and by the use of an appropriate entropy source that conforms to Part 2 of this Standard.

4. *A seed **shall** have a specified finite cryptoperiod, after which the seed **shall** be updated with sufficient additional entropy or operations using that seed **shall** cease operation.*

   This still needs to be addressed.

5. *A specific internal state **shall not** be deliberated reused, although this might happen by chance.*

   This requirement is fulfilled if a DRBG conforms to this part of the Standard and uses an entropy source that conforms to Part 2 of this Standard. If an implementation is validated, design evidence **shall** be provided to establish that this requirement will be fulfilled.

The objective for the internal state of an RBG is:

1. *The internal states used to produce public data such as nonces and initialization vectors **must** be fully independent from the states used to produce secret data such as cryptographic keys.*

This Standard recommends, but does not require, seed separation for different types of random data, including using different seeds for the generation of public data than are used to generate secret data (see Section 9.4).

8.2.4 Internal State Transition Function

The internal state transition function uses the internal state and one or more Approved algorithms to produce pseudorandom bits. During this process, the internal state of the DRBG is altered. The algorithms used and the method of altering the internal state depends on the specific DRBG.

The DRBGs in this Standard have three separate state transition functions:

1. Prior to the initial use of the DRBG, seed material is obtained, and all initial input is determined. The initial input is used as all or part of the initial state of the DRBG.
2. Each request for pseudorandom bits produces the requested bits using the current internal state and determines a new internal state that is used for the next request of bits.
3. When an application determines that reseeding of the DRBG is required, a reseeding function obtains new seed material, combines it with the current internal state values, and determines a new internal state for the next request for pseudorandom bits. By combining the new seed material with the current internal state, the entropy available from the current state is not lost, but is enhanced by the entropy of the new seed material.

Part 1 of this Standard provides objectives and requirements on the internal state transition function of a random bit generator. These objectives and requirements are specified below in italics, followed by a discussion about how they are satisfied by the DRBGs in this part of the Standard.

The requirement for the internal state transition functions of an RBG is:

1. *The deterministic elements of the transition function* ***shall*** *be verifiable via known-answer testing.*

   This requirement is fulfilled by the operational testing specified in Section 11 and, optionally, by implementation validation.

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

1. *The internal state transition function* ***must*** *depend on all the entropy carried by the internal state.*

   This objective is fulfilled by the design of the DRBGs in this Standard (see Section 10).

2. *It* ***must not*** *be feasible (either intentionally and 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).*

   This objective is fulfilled by the design of each DRBG in this Standard during one instance of the DRBG. There is no such assurance between different instances of

Comment [ebb2]: Page: 36
Need to bring the necessity of reseeding into this.

the same DRBG. However, when an entropy source that complies with Part 2 of this Standard is used so that the entropy source is statistically unique, there is a very low probability of a recurrence.

3. *The internal state transition functions* ***must*** *resist observation and analysis via power consumption, timing, radiation emissions, or other side channels as appropriate.*

   Fulfillment of this objective will depend on the embodiment of the DRBG. When an implementation is validated, design evidence **shall** be provided that indicates how this objective is satisfied; testing will be conducted to provide assurance that this objective has been met .

4. *The internal state transition function* ***may*** *enable the RBG to recover from the compromise of the internal state through periodic incorporation of entropy.*

   When an implementation is validated, design evidence **shall** be provided to provide assurance that the DRBG is able to fulfill this objective if such a capability has been designed into the DRBG.

8.2.5 Output Generation Function

The output generation function of a DRBG produces pseudorandom bits that are a function of the internal state of the DRBG and any input that is introduced while the internal state transition function is operating. These pseudorandom bits are deterministic with respect to the input information. Any formatting of the bits prior to output is determined by a particular implementation.

Part 1 of this Standard provides objectives and requirements for the output generation function of a random bit generator. These objectives and requirements are specified below in italics, followed by a discussion about how they are satisfied by the DRBGs in this part of the Standard.

The requirements for the output generation function are:

1. *The (deterministic) output generation function* ***shall*** *be able to be validated via known-answer testing.*

   Operational testing as specified in Section 11 **shall** be performed and **shall** include known-answer testing. If an implementation is validated, known-answer testing will be performed and the implementation will be examined to ensure that known-answer testing will be performed during normal operations.

2. *The output* ***shall*** *be inhibited until the internal state exhibits/obtains sufficient entropy.*

   A DRBG **shall** be designed to inhibit operation and output until sufficient entropy is obtained for the desired level of security (see Section 9.4). If the DRBG requires one or more cryptographic keys, the DRBG **shall not** operate or produce output until the keys are available as specified in this part of the Standard (see Section 9.5). When an implementation is validated, design evidence **shall** be provided that will provide assurance that this requirement is fulfilled.

3. *Test output and public output* ***shall*** *be separated from secret output*

Comment [ebb3]: Page: 37
This may need rewording ?

The DRBG **shall** be designed to provide test output only during testing, and to otherwise provide output only from the appropriate DRBG instance.

4. *When the internal state is dependent on previous state(s), the output generation function **shall** protect the internal state, so that analysis of randomizer outputs do not reveal useful information (from the point of view of compromise) about the internal state.*

   The DRBGs in this Standard have been designed to fulfill this requirement.

The objectives for the output generation function of an RBG are:

1. *The output generation function **must** depend on all of the entropy carried by the internal state.*

   The DRBGs in this Standard have been designed to fulfill this objective. If an implementation is validated, testing will be conducted to provide assurance that the implementation conforms to the algorithm specification.

2. *Within the constraints of the consuming application, the output generation function **must** be resistant to influences that will produce a chosen, previously ungenerated string.*

   Guidance has been provided about the DRBG boundary and its relation to other functions inside and outside that boundary (see Section 9.2). If an implementation is validated, testing will be conducted to provide assurance that the implementation conforms to the algorithm specification.

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

   This objective depends on the embodiment of the DRBG. If an implementation is validated, design evidence **shall** be provided that will provide assurance that this objective is fulfilled.

4. *When the output is generated in blocks from internal states whose bits are not fully independent (i.e., the RBG is functioning deterministically, or the entropy collection of a non-deterministic RBG produces bits with non-zero correlation), then changing one bit of the input **must** result in changing approximately half of the bits of the output. It **must** be infeasible to predict which output bits will change, without knowing the entire input.*

   This feature has been designed into the DRBGs. If an implementation is validated, testing will be conducted to provide assurance that the implementation conforms to the algorithm specification.

8.2.6 Support Functions

The support functions for a DRBG are concerned with assessing and reacting to the health of the DRBG.

A DRBG **shall** be designed to permit testing that will ensure that the generator is correctly implemented and continues to operate correctly. A test function **shall** be available for this purpose. The test function **shall** also allow the insertion of predetermined values of the input information in order to test for expected results. If any test fails, the DRBG **shall** enter an error state and output an error indicator. The DRBG **shall not** perform any operations while in an error state. All output **shall** be inhibited when an error state exists.

Error states may include "hard" errors that indicate an equipment malfunction that may require maintenance, service, repair or replacement of the DRBG, or may include recoverable "soft" errors that may require initialization or resetting of the DRBG. Recovery from error states **should** be possible except for those caused by hard errors that require maintenance, service, repair or replacement of the DRBG. [Editor's note: We probably need to include more advice on recovering from errors.]

Optional implementation validation is specified in Section 11.2 and in *TG-19, Part X, Implementation Validation of Random Bit Generators*. Operational testing **shall** be implemented in accordance with the tests specified in Section 11.3.