7 Top-Level Security Objectives and Requirements

7.1 Top-Level Objectives and Requirements for Random Bit Generator Output

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

The top-level testable requirements for RBG output streams are as follows;

1. *A series of outputs of the RBG **shall** pass the statistical tests specified in Part 2 of this standard.*

   The DRBGs specified in this part of the Standard have been determined to pass these tests prior to inclusion in this standard. Further statistical testing of these Approved DRBGs is not required.

2. *Considered as a black box, the RBG output **shall** have forward and/or backward secrecy. This means that, given only a sequence of output bits, it **shall not** be feasible (up to the selected security level) to compute or predict any future and/or previous output bit.*

   In this part of the Standard, the boundary of the black box is called a DRBG boundary. The DRBGs specified herein provide forward and backward secrecy when observed from outside the DRBG boundary, providing that the seed material and states of the DRBG are not disclosed. However, when considered from within the DRBG boundary, a given DRBG may not provide one or more of these properties.

The top-level untestable objectives for the RBG output are:

1. *Under reasonable assumptions, it **must not** be feasible to distinguish the output of the RBG from true random numbers that are uniformly distributed. All possible outputs **must** occur with equal probability. A series of outputs **must** appear to conform to the uniform distribution.*

   The DRBGs specified herein have been designed with this property.

2. *The outputs of the RBG **must** appear to be independent and unbiased. Given a sequence of output bits, it **must not** be feasible to compute or predict any output bit.*

   The DRBGs specified herein have been designed with this property.

3. *The outputs **must** be statistically unique.*

   The DRBGs specified herein have been designed with this property.

7.2 Top-Level Objectives and Requirements for Random Bit Generator Properties and Operation

Comment [ebb1]: Page: 29
All these need to be checked that they are actually covered within the document..

Part 1 of this Standard provides top-level objectives and requirements for random bit generator properties and operation. These objectives and requirements are specified below

in italics, followed by a discussion about how they are satisfied for the DRBGs in this part of the Standard.

The top-level testable requirements for the properties and operations of an RBG are as follows:

1. *The RBG **shall not** generate bits unless the generator possesses sufficient entropy. The criteria for sufficiency **shall** be the greater of the requirements of this Standard and the requirements of the consuming application.*

   Guidance is provided in Sections 9 - 10 regarding this requirement. When an implementation is validated, design evidence **shall** be provided and testing will be conducted to establish that the DRBG does not provide output until sufficient entropy has been acquired.

2. *The RBG **shall** either (A) enter a permanent error state, or (B) be able to recover from a loss or compromise of entropy if the permanent error state is deemed unacceptable for the application requirements. These requirements may be satisfied procedurally or innately in the design.*

   Guidance is provided in Section 11 regarding this requirement. If an implementation is validated, testing will be conducted to establish that this requirement has been fulfilled.

The top-level objectives for the properties and operations of an RBG that are generally untestable are as follows:

1. *The probability that the RBG can "misbehave" in some pathological way that violates the output requirements (e.g., constant output or small cycles, that is, looping such that the same output is repeated) **must** be sufficiently small. That means that the probability of error **must** be consistent with the overall confidence required of the RBG, which need not coincide with the required strength of cryptographic security.*

   Comment [ebb2]: Page: 29
   Need to expand on what is meant here.

   The design of the DRBGs specified herein, together with the operational testing specified in Section 11 will mitigate the possibility of misbehavior. If an implementation is validated, testing will provide further assurance that the DRBG will not misbehave and violate this objective.

2. *The RBG **must** be free from predictable influence, manipulation, or side-channel observation. This means that the design and implementation should have a defined protection boundary, for example, a FIPS 140-2 cryptographic module boundary.*

   When implemented within a well-defined protection boundary using the guidance provided in Section 11, there should be a reasonable expectation that the DRBGs specified herein will be free from such influence, manipulation or observation.

3. *The RBG **must** be protected in a manner that is consistent with the use and sensitivity of the output for the consuming application.*

   Guidance is provided herein for protecting the DRBG.

   Comment [ebb3]: Page: 30
   Probably need to expand on this.

4. *Considered as a glass box, an RBG **may** have forward and/or backward secrecy. This is more general than the requirement on the output strings; it means that given*

*all accessible information about the RBG (comprising some subset of inputs, algorithms, and outputs), it **shall** be infeasible (up to the specified security level) to compute or predict any future and/or previous output bit.*

The glass box concept refers to observation of the DRBG from inside the DRBG boundary; i.e., it refers to knowledge of the inner workings of the DRBG. The forward and backward secrecy attributes of a given DRBG are discussed for each DRBG.

5. *A consuming application **may** require that the RBG be auditable when using a seed that is publicly known (capable of reconstructing the bits it generates). A different consuming application **may** require that the RBG not be auditable, for example, when using a secret seed.*

Comment [ebb4]: Page: 30
Do we still want to keep this objective ? I thought that we were now assuming that all DRBG inpus should be secret.