8.2 DRBG Boundary

DRBG processes **shall** be encapsulated within DRBG boundaries. A boundary may be either physical or conceptual. Within a DRBG boundary,

1. The DRBG internal state and the operation of the DRBG processes **shall** only be affected according to the DRBG specification.
2. The DRBG internal state **shall** exist solely within the DRBG boundary.
3. Information about secret parts of the DRBG internal state and intermediate values in computations involving these secret parts **shall not** affect any information that leaves the DRBG boundary, except as specified for the DRBG pseudorandom bit outputs. The internal state **shall** be contained within the DRBG boundary and **shall not** be accessible from outside the boundary.

Each DRBG design in Section 10 includes one or more cryptographic primitives (e.g., a hash function). The DRBG boundary **shall** be specified to either prohibit or permit access of the cryptographic primitive (e.g., a hash function) by other cryptographic functions (e.g., a digital signature generation or verification process).

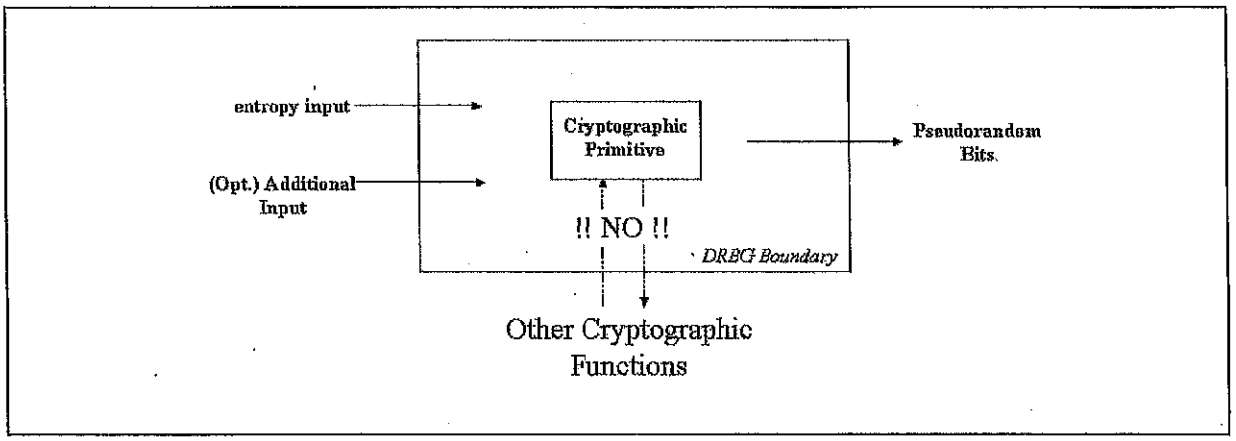

Figure 3: DRBG Boundary Containing No Other Functions

In Figure 3, any other functions reside outside the DRBG boundary. A cryptographic primitive within the DRBG boundary **shall not** be accessible by functions outside the DRBG boundary; observe the !!NO!! across the dotted vertical arrows in the figure. For example, a digital signature function that is outside the DRBG boundary **shall not** use a hash function that is contained within the DRBG boundary. In this case, a separate hash function would be required for the digital signature function. A design in which the DRBG is the only functionality within the DRBG boundary will provide a higher level of assurance than a design in which other functions can access the DRBG's cryptographic primitive or internal state.

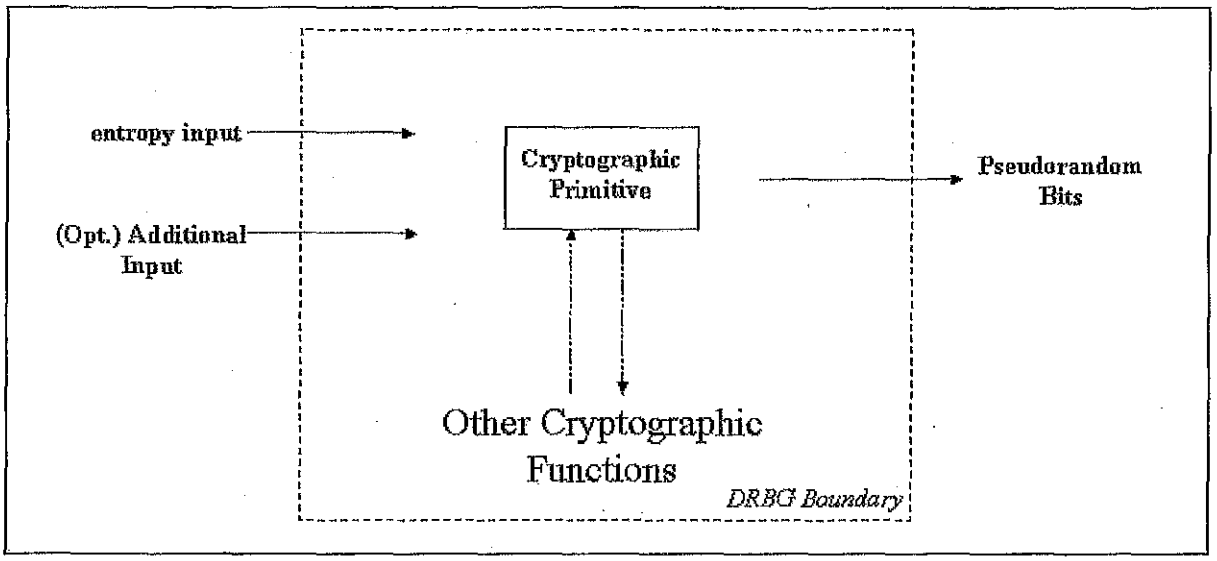

Figure 4: DRBG Boundary Containing Other Functions

In Figure 4, functions in addition to DRBG processes reside within the DRBG boundary. This design provides less assurance than the design shown in Figure 3 because the internal state is potentially accessible by other non-DRBG functions. When a DRBG boundary contains other non-DRBG functions, the internal state of the DRBG **shall not** be used or modified by these other non-DRBG functions. However, using this design, a cryptographic primitive used by the DRBG processes **may** be used by other cryptographic functions within the DRBG boundary (e.g., during the generation of verification of digital signatures); observe the vertical dashed arrows to and from the DRBG's cryptographic primitive.

In both figures, the entropy input is shown as being provided from outside the DRBG boundary. This is depicted as such for convenience only. The entropy input may be provided from either inside or outside the DRBG boundary. In either case, the requirements for protecting and handling the entropy input and the resulting seed are specified in Section 8.5.

DRBG processes used by consuming applications **shall** be implemented within FIPS 140-2 cryptographic module boundaries. A DRBG boundary **shall** be either fully contained within the cryptographic module boundary or **shall** be coincident with the cryptographic module boundary. Annex B discusses implementation considerations associated incorporating the DRBG boundary with the cryptographic module boundary.