1. Seed File Use

Entropy input $\searrow$

Personalization string $\nearrow$ → Seed → **DRBG**

Seed File ← Output 1

Output x

Seed File ← Last output

Next instantiation:

Entropy input $\searrow$

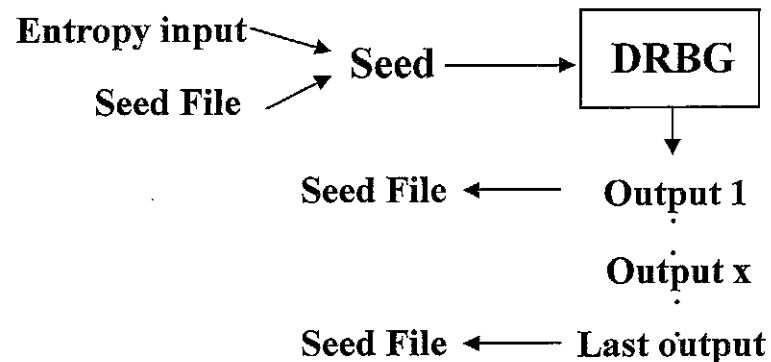Seed File $\nearrow$ → Seed → **DRBG**

Seed File ← Output 1

Output x

Seed File ← Last output
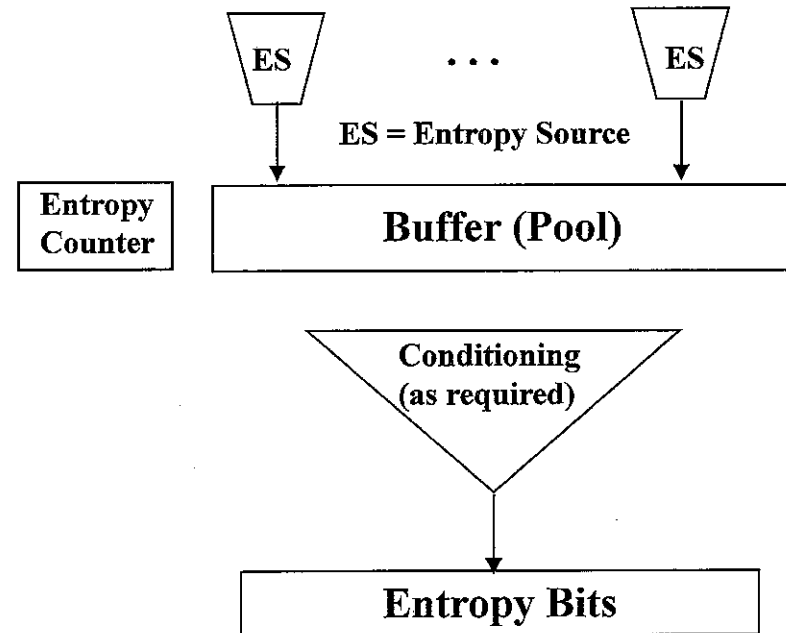
1. Initial instantiation: Use entropy input and (opt.) personalization string to form the seed.
2. Save 1st output in a seed file; replace seed file with last output.
3. New instantiation: Use entropy input and current seed file to form a seed.
4. Go to step 2.

2. Using a Pool of Bits

ES · · · ES

ES = Entropy Source

Entropy Counter

Buffer (Pool)

Conditioning (as required)

Entropy Bits

1. Collect entropy in a pool.
2. Perform conditioning when sufficient entropy has been obtained (as determined by the output buffer).
3. Output has full entropy.

3. Using a Pool of Bits
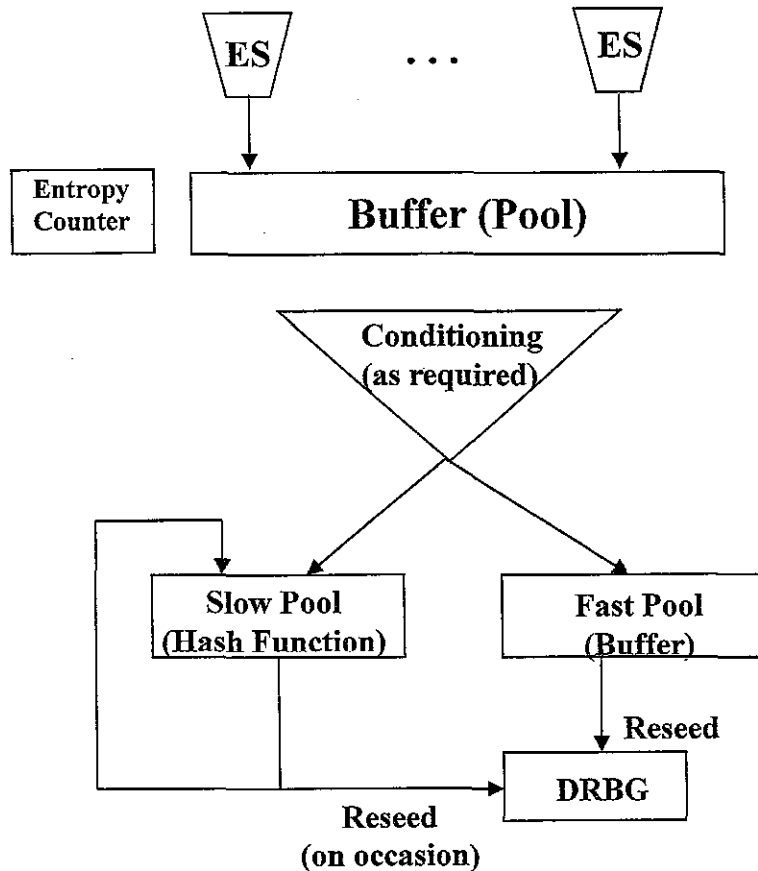
1. Collect entropy in a pool.
2. When sufficient entropy has collected, perform conditioning.
3. Place conditioned bits in the fast pool (a buffer); entropy estimate is best/expected amount.
4. Also place conditioned bits in the fast pool, but hash with previous contents of the fast pool; entropy estimate is conservative (e.g., a fraction of the expected amount).
5. Reseed the DRBG from each fast pool replacement.
6. Reseed the DRBG from the slow pool when the conservative estimate is sufficient.

4. Using a CRC
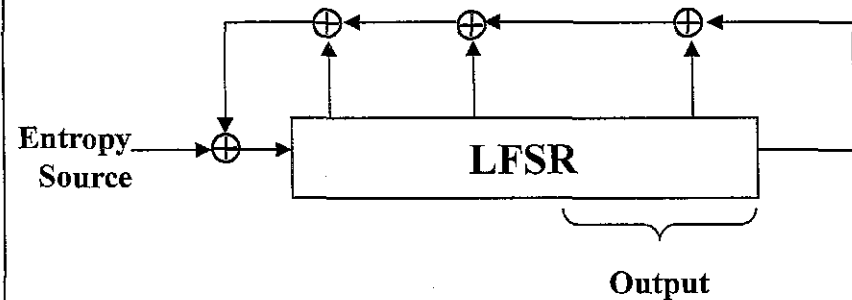
5. Non-persistent State
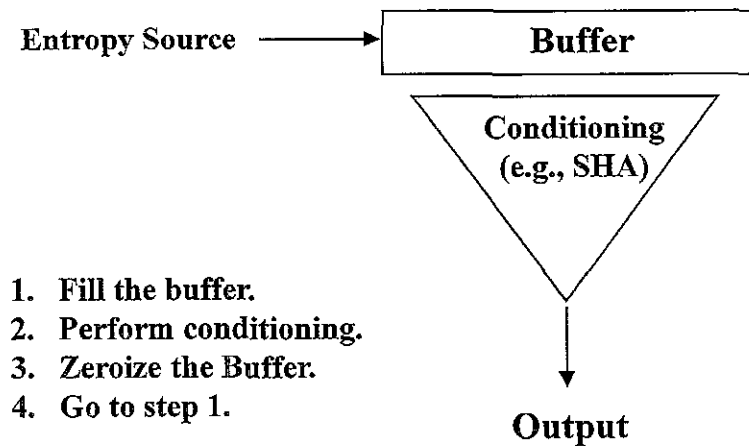
1. Fill the buffer.
2. Perform conditioning.
3. Zeroize the Buffer.
4. Go to step 1.