Algorithm krand

CSPRNG algorithm with $\kappa = 128$ bits of security, implemented following the CTR_DRBG specification [BK12] alongside a RandWrap initialized with device-bounded inputs (sk, t_1) and a κ-bit counter t_2. Realizes $\mathcal{F}^{PRNG}$:

- Seed() initializes a CSPRNG instance with fresh entropy. Automatically called after a certain number of samples to reseed the CSPRNG instance. Internally it performs several steps:
  1. Sample $seed \in \{0,1\}^\kappa$ and $salt \in \{0,1\}^\kappa$ from the OS randomness API.
  2. Compute $seed' \leftarrow \mathsf{RandWrap}(seed)$ and $salt' \leftarrow \mathsf{RandWrap}(salt)$.
  3. Seed the CTR_DRBG construction with $seed'$ and $salt'$.
- Sample$(n) \rightarrow u \in \{0,1\}^n$ generates n uniformly random bits from the CTR_DRBG construction.

References

[BK12] Elaine B Barker and John M Kelsey. Sp 800-90a. recommendation for random number generation using deterministic random bit generators, 2012.