DRBG recommendations from the X9.82 Editing Group

In an effort to move X9.82 Part 3 past the debate over which DRBGs to include in the initial iteration, the editing group is proposing the following list of DRBGs for inclusion. Factors that were considered in creating this list include 1) diversity of technologies (hash function, block cipher, number theoretic), 2) efficiency, and 3) security. We also attempted to narrow the list without unduly limiting diversity (and possibly hamstringing developers).

Recommendations (and brief rationale/discussion):

HMAC_DRBG

We decided to go with the cleaner, more understood HMAC_DRBG, over HASH_DRBG. HASH_DRBG has been redesigned so many times and by so many people that it has become a patchwork. HMAC_DRBG is rather conservative, and down the road there may be room for a more efficient hash-based DRBG to be added to the Standard.

CTR_DRBG & OFB_DRBG

We decided to include both block cipher DRBGs. Developers who have either an Output Feedback or Cipher Block Chaining Mode implementation will be able to reuse it for the OFB_DRBG. Obviously, Counter Mode implementations can be reused for CTR_DRBG. Implementing both (if somebody felt compelled to do so) would not take much additional work, as the core operations are the same.

We recommend support for all key sizes of AES, as well as three-key Triple DES.

DUAL_EC_DRBG (prime curves only)

We recommend keeping DUAL_EC_DRBG. Despite the fact that it is much slower than the other DRBGs, it offers a third distinct technology that can serve as a hedge against breakthroughs in cryptanalysis of hashes and block ciphers. Further work has been done to improve our understanding of both the cycle structure and the distribution of x-coordinates for the DUAL_EC_DRBG, and NSA is working to get these results released. In the interest of cutting back on the workload for validators, we are suggesting that only the four NIST curves (P-224, P-256, P-384, P-521) that are defined over prime fields be included in the Standard. These are the curves that have been analyzed most carefully for use in the DUAL_EC_DRBG, and they are also the curves that have been adopted by early implementers. We also recommend that a method for the verifiable generation of points be included.

We suggest dropping HASH_DRBG and the MS_DRBG, as well as support for the other NIST curves in the DUAL_EC_DRBG.