Random Bit Generation Requirements

The cryptographic techniques in many of the X9 standards require the generation of random values. These random values shall be obtained from an approved X9 random bit generator. The following are approved generators:

- Type I: A non-deterministic random bit generator satisfying the criteria in Clause 8.

- Type II: One of the deterministic random bit generators in Clause 7, seeded by a second approved random bit generator.

The "second random bit generator" in (b) may itself be either Type I or Type II. Accordingly, a "chain" of deterministic random bit generators is allowed, where each is seeded by the next deterministic random bit generator, except for the last, which is seeded by a non-deterministic random bit generator.

The random values required in X9 standards are generally of two types: either a bit string of a specified length, or an integer in a specified interval. In the first case, the bit string shall be obtained by generating a sequence of bits of the specified length. In the second, the integer shall be obtained by one of the techniques in Clause [[ref]]. [[Note: This is the proposed new clause with techniques for obtaining an integer in a certain interval from a sequence of random bits.]]

The deterministic random bit generators in Clause 7 are designed to generate random bits from a secret, random seed. However, they may also be used in certain applications to generate random bits from a public seed. For instance, they may be used to generate domain parameters in a discrete logarithm or elliptic curve cryptosystem (see ANSI X9.42 or ANSI X9.63). In this case, since the domain parameters are public, the seed may also be published to give other parties assurance that the domain parameters have not been selected arbitrarily. [[Note: Do all of the generators in Clause 7 have the appropriate property for this purpose, i.e., that it is difficult to find a seed corresponding to a given output?]]

Primitives and Generation Methods

- Deterministic random bit generators may be viewed as primitives, i.e., mathematical transformations (different than encryption/decryption primitives, since they have an internal state)

- Generation methods vary:
    - a secret, random bit string
    - a secret, random integer in an interval
    - a public, random bit string
    - a public, random integer in an interval

- As an example, the method for generating a secret, random bit string might be as follows:

Options:

- RBGS, an approved random bit generation scheme for generating a seed
- DRBG, a deterministic random bit generator

Initialization:

- Generate a random seed with RBGS.
- Initialize the DRBG with the random seed.

Generation:

- Obtain an output of the appropriate length from DRBG.