Appendix E : DRBG Selection

E.3 DRBGs Based on Block Ciphers

E.3.1 The Two Constructions: CTR and OFB

This standard describes two classes of DRBG based on block ciphers: One class uses the block cipher in OFB-mode, the other class uses the CTR-mode. There are almost no security differences between these two DRBGs; CTR mode guarantees that short cycles cannot occur in a single output request, while OFB-mode simply guarantees that short cycles will have an extremely low probability. OFB-mode makes slightly less demanding assumptions on the block cipher, but the security of both DRBGs relates in a very simple and clean way to the security of the block cipher in its intended applications. This is a fundamental difference between these DRBGs and the DRBGs based on hash functions, where the DRBG's security was ultimately based on pseudorandomness properties that don't form a normal part of the requirements for hash functions. An attack on any of the hash-based DRBGs would not necessarily represent a weakness in the hash function; for these block cipher-based constructions, a weakness in the DRBG is directly related to a weakness in the block cipher.

To be a little more concrete, each request for pseudorandom bits made without any additional input produces up to 2^{32} bytes (2^{35} bits) under AES, or 2^{16} bytes (2^{19} bits) under TDEA. Each request leads to the generation of the requested number of pseudorandom bits, followed by a rekeying that is performed using some additional output bits.

Comment [ebb1]: See the note below.

For the CTR mode, suppose there is an attack that allows the attacker to distinguish the outputs from random. This can be used to distinguish the block cipher from random in a chosen plaintext attack with the same text requirements and resources. For the OFB mode, this is also true, unless a short cycle is encountered. With the limits on output sizes per request imposed for AES and TDEA, this happens with negligible probability. (2^{16} outputs are allowed with TDEA; this leaves approximately a 2^{-48} probability of a short cycle. With AES, 2^{32} outputs are allowed; this leaves approximately a 2^{-96} probability of a short cycle.)

Comment [ebb2]: Page: 17
In the previous paragraph, it's 2^{16} and 2^{32} bytes, whereas this paragraph discusses outputs. In the original wri

At the end of each request, the block cipher key and initial block (V) are regenerated by the DRBG. Suppose that the selection of this key leads to a bias in the next output. Then, an attacker would again have a straightforward way to convert that attack into one that demonstrates a weakness in the CTR or OFB modes, and thus in the underlying block cipher.

Assuming that the outputs are indistinguishable from random, the maximum of 2^{64} rekeyings lead to the following rough probabilities of a short cycle:

Cipher	Total States	P(cycle) in 2^{64} Tries
AES-128	256	2^{-128}
AES-192	384	2^{-256}
AES-256	512	2^{-384}
2 key TDEA	176	2^{-48}
3 key TDEA	232	2^{-104}

For all of these, the cycling probabilities in 2^{64}} requests are negligible.

E.3.2 Implementation Issues

The only thing required to implement the raw DRBGs is access to the key schedule and the encryption function of the block cipher algorithm. For these DRBGs, the decryption function can be ignored, which helpful for ciphers like AES, whose decryption function is a bit different from the encryption function.

In order to implement the full DRBG, with the ability to be instantiated and reseeded with free-form input strings, rather than only with full-entropy strings, we must also implement a block cipher derivation function (**Block_Cipher_df**). This has yet to be defined.

Comment [ebb3]: What does this mean?

E.3.3 Performance Characteristics

The block cipher-based DRBGs provide excellent performance. For a single request, the overhead is between three and four block encryptions and one rekeying via the **Update** function. Each *blocklen*-bit piece of the output (where *blocklen* is the block size of the block cipher algorithm) is generated with a single encryption operation.

These DRBGs can be used for any Approved block cipher algorithm with *blocklen* ≥ 64 and *keylen* ≥ 112. ~~(A block cipher such as Skipjack, with an 80 bit key and 64 bit block, would not work here; the probability of a short cycle in 2^{64} requests would be about 2^{-16}, far too high to be acceptable!) However, we note that block ciphers with extremely slow key schedules, such as Blowfish and Khufu, are not very practical with these DRBGs, because the per-request overhead will be very high.~~

Comment [ebb4]: Except for Skipjack, these algorithms are not approved anyway.