Variable Keysize:

The CAST-128 encryption algorithm [**C.Adams, "Constructing Symmetric Ciphers using the CAST Design Procedure",** ***Designs, Codes, and Cryptography*** **(to appear)**] has been designed to allow a key size which can vary from 40 bits to 128 bits, in 8-bit increments (that is, the allowable key sizes are 40, 48, 56, 64, ..., 112, 120, and 128 bits.

The specification is as follows:

1) For key sizes up to and including 80 bits (i.e., 40, 48, 56, 64, 72, and 80 bits), the algorithm is exactly as specified but uses 12 rounds instead of 16;

2) For key sizes greater than 80 bits, the algorithm uses the full 16 rounds;

3) For key sizes less than 128 bits, the key is padded with zero bytes (in the rightmost, or least significant, positions) out to 128 bits (since the CAST-128 key schedule assumes an input key of 128 bits).

In order to ensure that the algorithm is implemented correctly, the following test vectors can be used for verification.

```
128-bit    key         = 01 23 45 67 12 34 56 78 23 45 67 89 34 56 78 9A
           plaintext   = 01 23 45 67 89 AB CD EF
           ciphertext  = 23 8B 4F E5 84 7E 44 B2

80-bit     key         = 01 23 45 67 12 34 56 78 23 45
                       = 01 23 45 67 12 34 56 78 23 45 00 00 00 00 00 00
           plaintext   = 01 23 45 67 89 AB CD EF
           ciphertext  = EB 6A 71 1A 2C 02 27 1B

40-bit     key         = 01 23 45 67 12
                       = 01 23 45 67 12 00 00 00 00 00 00 00 00 00 00 00
           plaintext   = 01 23 45 67 89 AB CD EF
           ciphertext  = 7A C8 16 D1 6E 9B 30 2E
```

CAST OIDs:

For those who may be using CAST in algorithm negotiation within a protocol, or in any other context which may require the use of OBJECT IDENTIFIERs, the following OIDs have been defined. Additional OIDs may be defined if and when necessary.

cast5CBC:

```
    iso(1) memberBody(2) usa(840) nt(113533) secureNetworks(7)
        algorithms(66) cast5CBC(10)

    Parameters ::= SEQUENCE {
        iv         OCTET STRING DEFAULT 0,  -- Initialization vector
        keyLength  INTEGER                  -- Key length, in bits
    }

    Note: The iv is optional and defaults to all-zero. On the encoding
          end, if an all-zero iv is used, then it should absent from
          the Parameters. On the decoding end, an absent iv should be
```

```
        interpreted as meaning all-zeros.

    This is encryption and decryption in CBC mode using the CAST-128
    symmetric block cipher algorithm (referenced above).


cast5MAC:

    iso(1) memberBody(2) usa(840) nt(113533) secureNetworks(7)
        algorithms(66) cast5MAC(11)

    Parameters ::= SEQUENCE {
        macLength  INTEGER,        -- MAC length, in bits
        keyLength  INTEGER         -- Key length, in bits
    }

    This is message authentication using the CAST-128 symmetric block
    cipher algorithm (referenced above).


pbeWithMD5AndCast5CBC:

    iso(1) memberBody(2) usa(840) nt(113533) secureNetworks(7)
        algorithms(66) pbeWithMD5AndCAST5-CBC(12)

    Parameters ::= SEQUENCE {
        salt            OCTET STRING,
        iterationCount  INTEGER,      -- Total number of hash iterations
        keyLength       INTEGER       -- Key length, in bits
    }

    Note: The IV is derived from the hashing procedure and therefore
          need not be included in Parameters.

    This is password-based encryption and decryption in CBC mode
    using MD5 and the CAST-128 symmetric block cipher algorithm
    (referenced above).
```