Simplified Data Encryption Standard
Known Answer Tests

The Known Answer tests are designed to verify the components of the S-DES algorithm in the implementation under test (IUT). These components include the initial permutation IP, the inverse permutation IP^{-1}, the expansion matrix E, the data permutation P, the key permutation PC1 and PC2, and the substitution tables S_1 and S_2. The tests exercise each bit of every component of the algorithm by processing all possible basis vectors through the IUT.

A generic overview of the Known Answer tests required for the validation of the IUTs implementing the encryption and/or decryption processes of S-DEA is discussed below.

The Encryption Process

An IUT which allows encryption requires the successful completion of five Known Answer tests: the Variable Plaintext Known Answer Test, the Inverse Permutation Known Answer Test, the Variable Key Known Answer Test for the Encryption Process, the Permutation Operation Known Answer Test for the Encryption Process, and the Substitution Table Known Answer Test for the Encryption Process.

These Known Answer tests are also used in the testing of IUTs implementing the decryption process.

The Variable Plaintext Known Answer Test

To perform the Variable Plaintext Known Answer Test, the test supplies the IUT with initial values for the key, the plaintext(s) and, if applicable, the initialization vector(s). The test initializes all keys to zero. Each block of data input into the S-DEA is represented as a 8-bit basis vector.

This test is repeated 8 times, using the 8 input basis vectors, allowing for every possible basis vector to be tested. At the completion of the 8th cycle, all results are verified for correctness.

If correct results are obtained from an IUT, the Variable Plaintext Known Answer Test has verified the initial permutation IP and the expansion matrix E via the encrypt operation by presenting a full set of basis vectors to IP and to E. The test also verifies the inverse permutation IP^{-1} via the decrypt operation. It does this by presenting the recovered basis vectors to IP^{-1}.

The Inverse Permutation Known Answer Test

To perform the Inverse Permutation Known Answer Test, the test supplies the IUT with initial values for the key, the plaintext(s) and, if applicable, the initialization vector(s).

This test performs the same processing as the Variable Plaintext Known Answer Test. The difference is that the plaintext value(s) for this test are set to the ciphertext result(s) obtained from the Variable Plaintext Known Answer Test for the corresponding modes of operation.

The key is initialized to zero. This key is a self-dual key. A self-dual key is a key with the property that when you encrypt twice with this key, the result is the initial input. Therefore, the result is the same as encrypting and decrypting with the same key. Using a self-dual key allows basis vectors to be presented to components of the S-DEA to validate the IUT's performance. This is discussed further in the last paragraph of this section.

This test, when applied to an IUT, verifies the inverse permutation (IP^{-1}) via the encrypt operation, because as the basis vectors are recovered, each basis vector is presented to the inverse permutation IP^{-1}. By performing the decrypt operation, the initial permutation IP and the expansion matrix E are verified by presenting the full set of basis vectors to them as well.

The Variable Key Known Answer Test for the Encryption Process

To perform the Variable Key Known Answer Test for the Encryption Process, the test supplies the IUT with initial values for the key, the plaintext(s), and, if applicable, the initialization vector(s). For IUTs supporting the CBC mode of operation, an initial value is supplied to three plaintext variables. These three plaintext variables are initialized to the same value. The other modes of operation only require one plaintext variable.

During the initialization process, the plaintext value(s) and the initialization vector value(s) are set to zero.

This test is repeated 10 times, using the 10 key basis vectors to allow for every possible vector to be tested. At the completion of the 10th cycle, all results are verified for correctness.

When this test is performed for an IUT, the 10 possible key basis vectors which yield unique keys are presented to PC1, verifying the key permutation PC1 via the encrypt operation. Also, during the encrypt operation, a complete set of key basis vectors is presented to PC2 as well, so PC2 is verified.

This test also verifies the right shifts in the key schedule via the S-DES decrypt operation as the basis vectors are recovered.

The Permutation Operation Known Answer Test for the Encryption Process

To implement the Permutation Operation Known Answer Test for the Encryption Process, the test supplies the IUT with 4 key values. The test also supplies initial values for the plaintext(s) and, if applicable, the initialization vector(s). During the initialization of a test, the plaintext value(s) and the first (or only) initialization vector value are set to 0, while the key values are assigned to one of the 4 key values supplied by the test.

Each of the 4 key values supplied by the test is tested. At the completion of the 4th cycle, all results are verified for correctness.

The 4 key values used in this test present a complete set of basis vectors to the permutation operator P. By doing so, P is verified. This occurs when both the encrypt and decrypt operations are performed.

The Substitution Table Known Answer Test for the Encryption Process

To implement the Substitution Table Known Answer Test for the Encryption Process, the test supplies the IUT with 7 key-data sets. Depending on the mode of operation implemented, the data value will be assigned to the plaintext or to the initialization vector variables. During initialization, the plaintext values (or the initialization vector values, depending on the mode of operation supported), and the key values are initialized to one of the 19 key-data sets supplied by the test.

This test is repeated for each of the 7 key-data sets, allowing every value in the set of 7 key-data sets to be tested. At the completion of the 7th set, all results are verified for correctness.

The set of 7 key-data sets used in this test result in every entry of both S-box substitution tables being used at least once during both the encrypt and decrypt operations. Thus, this test verifies the 64 entries in each of the eight substitution tables.

The Decryption Process

The five Known Answer tests required for validation of IUTs implementing the decryption process of the S-DEA consist of the Variable Ciphertext Known Answer Test, the Initial Permutation Known Answer Test, the Variable Key Known Answer Test for the Decryption Process, the Permutation Operation Known Answer Test for the Decryption Process and the Substitution Table Known Answer Test for the Decryption Process. These tests are only performed by IUTs that support the ECB and CBC modes of operation, since only these modes of operation utilize the three S-DEA stages in reverse order during the decryption process.

The Variable Ciphertext Known Answer Test

To perform the Variable Ciphertext Known Answer Test, the test supplies the IUT with 8 ciphertext values. These values are obtained from the results of the Variable Plaintext Known Answer Test if the IUT performs both encryption and decryption. Otherwise, the test will supply the IUT with the ciphertext values. If applicable, the test also supplies initial values for the initialization vector(s). The keys and initialization vectors are initialized to zero for each test.

This test is repeated once for each of the 8 ciphertext values. If the 8 resulting plaintext values form the set of basis vectors, it can be assumed that all of the operations were performed successfully.

As the basis vectors are recovered via the decrypt operation, they are presented to the inverse permutation IP^{-1}, thus verifying it. This test also verifies the initial permutation IP and the expansion matrix E via the encrypt operation by presenting a full set of basis vectors to these components.

The Initial Permutation Known Answer Test

To perform the Initial Permutation Known Answer Test, the TMOVS supplies the IUT with initial values for the ciphertext, the keys, and, if applicable, the initialization vector(s). The ciphertext value(s) are set to the plaintext result(s) obtained from the Variable Ciphertext Known Answer Test.

The key is initialized to zero (with odd parity set). This key is a self-dual key. A self-dual key is a key with the property that when you decrypt (or encrypt) twice with this key, the result is the initial input. Therefore, the result is the same as encrypting and decrypting with the same key. Using a self-dual key allows basis vectors to be presented to components of the S-DEA to validate the IUT's performance. This is discussed further in the last paragraph of this section.

This test is run for each of the 64 ciphertext values. At the completion of the 64th cycle, all results are verified for correctness.

This test, when applied to an IUT, verifies that the initial permutation IP and the expansion matrix E via the decrypt operation, by presenting the full set of basisi vectors to the components. Via the encrypt operation, this test also verifies the inverse permutation (IP^{-1}) as the basis vectors are recovered by presenting each basis vector to the inverse permutation IP^{-1}.

The Variable Key Known Answer Test for the Decryption Process

To implement the Variable Key Known Answer Test for the Decryption Process, the test supplies the IUT with 10 keys. The test also supplies initial values for the initialization vector values, if applicable.

During the initialization process, the ciphertext value(s) are initialized in one of two ways. If the IUT supports both encryption and decryption, the values resulting from the encryption performed in the Variable Key Known Answer Test for the Encryption Process will be used to initialize the ciphertext values. Otherwise, the TMOVS will supply the ciphertext values along with the information discussed in the previous paragraph. The initialization vector value(s) are set to zero for each test. The key for each round is initialized to a 10-bit key basis vector which contains a "1" in the i^{th} significant position and "0"s in all remaining significant positions of the key.

This test is repeated for each of the 10 key basis vectors, allowing for every possible key basis vector to be tested. At the completion of the 10th cycle, all results are verified for correctness.

This test verifies the right shifts in the key schedule via the S-DEA decrypt operation as the basis vectors are recovered.

During the encrypt operation, a complete set of basis vectors is presented to the key permutation, PC1, thus verifying PC1. Since the key schedule consists of left shifts, a complete set of basis vectors is also presented to PC2 verifying PC2 as well.

The Permutation Operation Known Answer Test for the Decryption Process

To implement the Permutation Operation Known Answer Test for the Decryption Process, the test supplies the IUT with 4 key-data sets, consisting of an initial value for the key and values for the ciphertext. The test also supplies initial values for the initialization vector(s), if applicable. The values for the key and ciphertext are supplied in one of two ways. If the IUT performs both encryption and decryption, values for the key and ciphertext resulting from the encryption performed in the Permutation Operation Known Answer Test for the Encryption Process will be used. Otherwise, the key and ciphertext values will be supplied by the test. If applicable, the initialization vector will be set to zero for each test.

This test is repeated for each of the 4 key-data sets. At the completion of the 4th set, the results of each of the 4 tests are verified to be zero.

The 4 key sets used in this test present a complete set of basis vectors to the permutation operator P. By doing so, P is verified. This occurs when both the encrypt and decrypt operations are performed.

The Substitution Table Known Answer Test for the Decryption Process

To implement the Substitution Table Known Answer Test for the Decryption Process, the test supplies the IUT with 7 key-data sets consisting of an initial value for the key and values for the ciphertext. The test also supplies initial values for the initialization vector, if applicable. The values for the keys and the ciphertext value(s) are supplied in one of two ways. If the IUT performs both encryption and decryption, the values for the key and ciphertext resulting from the encryption performed in the Substitution Table Known Answer Test for the Encryption Process will be used. Otherwise, the key and ciphertext values will be supplied by the test. If applicable, the initialization vector will be set to zero for each test.

This test is repeated for each of the 7 key-data sets allowing for the set of 7 key-data sets to be processed. At the completion of the 7th set, all results are verified for correctness.

The set of 7 key-data sets used in this test result in every entry of both S-box substitution tables being used at least once during both the encrypt and decrypt operations. Thus, this test verifies the 64 entries in each of the eight substitution tables.

S-DES Electronic Codebook (ECB) Mode

The IUTs which implement the S-DES Electronic Codebook (ECB) mode should be validated by the successful completion of a series of Known Answer tests corresponding to the cryptographic processes allowed by the IUT.

Encryption Process

The process of validating an IUT which implements the ECB mode of operation for the encryption process should involve the successful completion of the following five tests:

1. The Variable Plaintext Known Answer Test - ECB mode
2. The Inverse Permutation Known Answer Test - ECB mode
3. The Variable Key Known Answer Test for the Encryption Process - ECB mode
4. The Permutation Operation Known Answer Test for the Encryption Process - ECB mode
5. The Substitution Table Known Answer Test for the Encryption Process - ECB mode

An explanation of the tests follows.

The Variable Plaintext Known Answer Test – ECB Mode

```
Test:   Initialize:     KEY = $0000000000_b$

                        $P_1 = 10000000_b$

        Send:           KEY, $P_1$

IUT:    FOR i = 1 to 8
          {
                $I_i = P_i$

                $I_i$ is read into S-DEA and encrypted using KEY

                Send i, KEY, $P_i$, $C_i$

                $P_{i+1}$ = basis vector where single "1" bit is in position i + 1
          }
Test:   Compare results from each loop with known answers.
        See Table A.1.
```

The Inverse Permutation Known Answer Test - ECB Mode

```
Test:   Initialize:     KEY = 0000000000_b

                        P_i (where i = 1..8) = 8 C values from the Variable Plaintext Known
                        Answer Test

        Send:           KEY, P_1,…,P_8

IUT:    FOR i = 1 to 8
          {
                I_i = P_i

                I_i is read into S-DEA and encrypted using KEY

                Send i, KEY, P_i, C_i

                P_i+1 = corresponding P_i+1 from Test
          }

Test:   Compare results form each loop with known answers.
        Should be the set of basis vectors.
```

The Variable Key Known Answer Test for the Encryption Process - ECB Mode

```
Test:   Initialize:     KEY_1 = 1000000000_b

                        P = 00000000_b

        Send:           KEY_1, P

IUT:    FOR i = 1 to 10
          {
                I = P

                I is read into S-DEA and encrypted using KEY_i

                Send i, KEY_i, P, C_i

                KEY_i+1 = vector consisting of "0" in every bit position except for a single
                "1" bit in position i+1.
          }

Test:   Compare results to the 10 encryptions with known answers.
        Use Table A.2.
```

The Permutation Operation Known Answer Test for the Encryption Process - ECB Mode

```
Test:   Initialize:     $KEY_i$ (where i = 1..4) = 4 KEY values in Table A.3

                        $P = 00000000_b$

        Send:           P, $KEY_1$,..., $KEY_4$

IUT:    FOR i = 1 to 4
          {
              I = P

              I is read into S-DEA and encrypted using $KEY_i$

              Send i, $KEY_i$, P, $C_i$

              $KEY_{i+1} = KEY_{i+1}$ from Test
          }

Test:   Compare results form each loop with the known answers. Use Table A.3.
```

The Substitution Table Known Answer Test for the Encryption Process - ECB Mode

```
Test:   Initialize:     $KEY_i$ (where i = 1..7) = 7 KEY values in Table A.4

                        $P_i$ (where i = 1..7) = 7 corresponding P values in Table A.4

        Send:            $KEY_1$, $P_1$, $KEY_2$, $P_2$,..., $KEY_7$, $P_7$

IUT:    FOR i = 1 to 7
          {
              $I_i = P_i$

              $I_i$ is read into S-DEA and encrypted using $KEY_i$

              Send i, $KEY_i$, $P_i$, $C_i$

              $KEY_{i+1} = KEY_{i+1}$ from Test

              $P_{i+1} = P_{i+1}$ from Test
          }
```

Test: Compare results form each loop with the known answers. Use Table A.4.

Decryption Process

The process of validating an IUT which implements the ECB mode of operation in the decryption process should involve the successful completion of the following five tests:

1. The Variable Ciphertext Known Answer Test - ECB mode
2. The Initial Permutation Known Answer Test - ECB mode
3. The Variable Key Known Answer Test for the Decryption Process- ECB mode
4. The Permutation Operation Known Answer Test for the Decryption Process- ECB mode
5. The Substitution Table Known Answer Test for the Decryption Process - ECB mode

An explanation of the tests follows.

The Variable Ciphertext Known Answer Test – ECB Mode

Test: Initialize: KEY = 0000000000_b

If encryption is supported by IUT:

Send KEY

If encryption is not supported by IUT:

Initialize C_i (where i=1..8) = 8 C values in Table A.1

Send KEY, C_1, C_2, …, C_8

IUT: If encryption is supported by IUT:

Initialize C_1 = first value from output of Variable Plaintext Known Answer Test.

Otherwise, use the first value received from the Test.

```
FOR i = 1 to 8
  {
      $I_i = C_i$

      $I_i$ is read into S-DEA and decrypted using KEY

      Send i, KEY, $C_i$, $P_i$
```

```
            If encryption is supported:

            $C_{i+1}$ = corresponding $C_{i+1}$ from output of Variable Plaintext Known
            Answer Test

            else

            $C_{i+1}$ = corresponding $C_{i+1}$ from Test
        }
```

Test: Compare results from each loop with known answers. Should be the set of basis vectors.

The Initial Permutation Known Answer Test - ECB Mode

```
Test:   Initialize:     $KEY = 0000000000_b$

                        $C_i$ (where i = 1..8) = 8 P values from the Variable Plaintext Known
                        Answer Test

        Send:           KEY, $C_1,\ldots,C_8$

IUT:    FOR i = 1 to 8
        {
            $I_i = C_i$

            $I_i$ is read into S-DEA and decrypted using KEY

            Send i, KEY, $C_i$, $P_i$

            $C_{i+1}$ = corresponding $C_{i+1}$ from Test
        }
```

Test: Compare results form each loop with known answers.
See Table A.1

The Variable Key Known Answer Tests for the Decryption Process - ECB Mode

```
Test:   Initialize:     $KEY_1 = 1000000000_b$

        If encryption is supported by IUT:

                        Send $KEY_1$

        If encryption is not supported by IUT:
```

Initialize C_i (where i=1..10) = 10 C values in Table A.2

Send KEY, C_1, C_2, …, C_{10}

IUT: If encryption is supported by IUT:

Initialize C_i = first value from output of Variable Key Known Answer Test for the Encryption Process.

Otherwise, use the first value received from the Test.

```
FOR i = 1 to 10
  {
        $I_i = C_i$

        $I_i$ is read into S-DEA and decrypted using KEY

        Send i, $KEY_i$, $C_i$, $P_i$

        $KEY_{i+1}$ = vector consisting of “0” in every bit position except for a single
        “1” bit in position i+1.

        If encryption is supported by the IUT:

        $C_{i+1}$ = corresponding $C_{i+1}$ from output of Variable Key Known Answer
        Test for the Encryption Process

        else

        $C_{i+1}$ = corresponding $C_{i+1}$ from Test
  }
```

Test: Compare results of the 10 decryptions with known answers.
Should be P = 00000000_b for all 10 decryptions.

The Permutation Operation Known Answer Test for the Decryption Process - ECB Mode

Test: Initialize: KEY_i (where i = 1..4) = 4 KEY values in Table A.3

If encryption is supported by the IUT:

Send KEY_1, ⋯, KEY_4

If encryption is not supported by the IUT:

Initialize C_i (where i=1..4) = corresponding C values in Table A.3

Send KEY_1, C_1, KEY_2, C_2,⋯, KEY_4, C_4

IUT: If encryption is supported by the IUT:

Initialize C_1 = first value retained from Permutation Operation Known Answer Test for the Encryption Process

Otherwise, use the first value received from the Test.

```
FOR i = 1 to 4
  {
        $I = C_i$

        I is read into S-DEA and decrypted using $KEY_i$

        Send i, $KEY_i$, $C_i$, $P_i$

        $KEY_{i+1} = KEY_{i+1}$ from Test

        If encryption is supported:

                $C_{i+1}$ = corresponding $C_{i+1}$ from output of Permutation Operation
                Known Answer Test for the Encryption Process

        else

                $C_{i+1}$ = corresponding $C_{i+1}$ from Test
  }
```

Test: Compare results form each loop with known answers.
Should be $P = 00000000_b$ for all 4 rounds.

The Substitution Table Known Answer Test for the Decryption Process - ECB Mode

Test: Initialize: KEY_i (where i = 1..7) = 7 KEY values in Table A.4

If encryption is supported by the IUT:

Send KEY_1, ⋯, KEY_7

```
If encryption is not supported by the IUT:

        Initialize $C_i$ (where i=1..7) = corresponding C values in Table A.4

        Send $KEY_1$, $C_{1,}$ $KEY_2$, $C_2$,···, $KEY_7$, $C_7$

IUT:    If encryption is supported by the IUT:

        Initialize $C_1$ = first value retained from Substitution Table Known Answer
        Test for the Encryption Process

    Otherwise, use the first value received from the Test.

    FOR i = 1 to 7
      {
        $I = C_i$

        I is read into S-DEA and decrypted using $KEY_i$

        Send i, $KEY_i$, $C_i$, $P_i$

        $KEY_{i+1} = KEY_{i+1}$ from Test

        If encryption is supported:

            $C_{i+1}$ = corresponding $C_{i+1}$ from output of Substitution Table
            Known Answer Test for the Encryption Process

        else

            $C_{i+1}$ = corresponding $C_{i+1}$ from Test
      }

Test:   Compare results of each loop with the known answers. See Table A.4.
```

Appendix A
Tables of Values for the Known Answer Tests

Table A.1 Resulting Ciphertext from the Variable Plaintext Known Answer Test For the ECB Mode of Operation

(NOTE: KEY = 0000000000_b)

ROUND	PLAINTEXT	CIPHERTEXT
0	10000000	00010111
1	01000000	11010001
2	00100000	00111101
3	00010000	10100100
4	00001000	01111110
5	00000100	10101011
6	00000010	10011101
7	00000001	00101011

Table A.2 Resulting Ciphertext from the Variable Key Known Answer Test For the ECB mode of Operation

(NOTE: Plaintext/text = 00000000_b)

ROUND	Key	CIPHERTEXT
0	1000000000	11101010
1	0100000000	10111011
2	0010000000	00011001
3	0001000000	10101001
4	0000100000	10110110
5	0000010000	01000111
6	0000001000	00000111
7	0000000100	01111000
8	0000000010	01101001
9	0000000001	01111001

Table A.3 Values to be Used for the Permutation Operation Known Answer Test For the ECB mode of Operation

(NOTE: Plaintext/text = 00000000_b for each round)

ROUND	Key	CIPHERTEXT
0	0000100100	11000101
1	0010000100	11001010
2	0000000000	00100100
3	0000000101	00111111

Table A.4 Values to be Used for the Substitution Table Known Answer Test For the ECB mode of Operation

(NOTE: Plaintext/text = 00000000_b for each round)

ROUND	Key	CIPHERTEXT
0	0000000000	00100100
1	0000011001	00000101
2	0001100111	00011110
3	0001111101	01011001
4	0001111110	11101011
5	0010100111	10100101
6	0100001000	00100001