Archived NIST Technical Series Publication

The attached publication has been archived (withdrawn), and is provided solely for historical purposes. It may have been superseded by another publication (indicated below).

Archived Publication

Series/Number:	NIST Special Publication 800-17
Title:	Modes of Operation Validation System (MOVS): Requirements and Procedures
Publication Date(s):	February 1998
Withdrawal Date:	August 1, 2018
Withdrawal Note:	This validation system is for algorithms that have been deprecated (e.g., DES, Skipjack). For information on current algorithm validation systems, see information on the Cryptographic Algorithm Validation Program (CAVP).

Superseding Publication(s)

The attached publication has been **superseded by** the following publication(s):

Series/Number:	
Title:	
Author(s):	
Publication Date(s):	
URL/DOI:	

Additional Information (if applicable)

Contact:	Computer Security Division (Information Technology Laboratory)
Latest revision of the attached publication:	
Related information:	https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program https://csrc.nist.gov/publications
Withdrawal announcement (link):	https://csrc.nist.gov/news/2018/nist-to-withdraw-eleven-outdated-sp-800-pubs

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Date updated: August 1, 2018

NAT'L INST. OF STAND & TECH R.I.C.
A11105 354593
NIST PUBLICATIONS

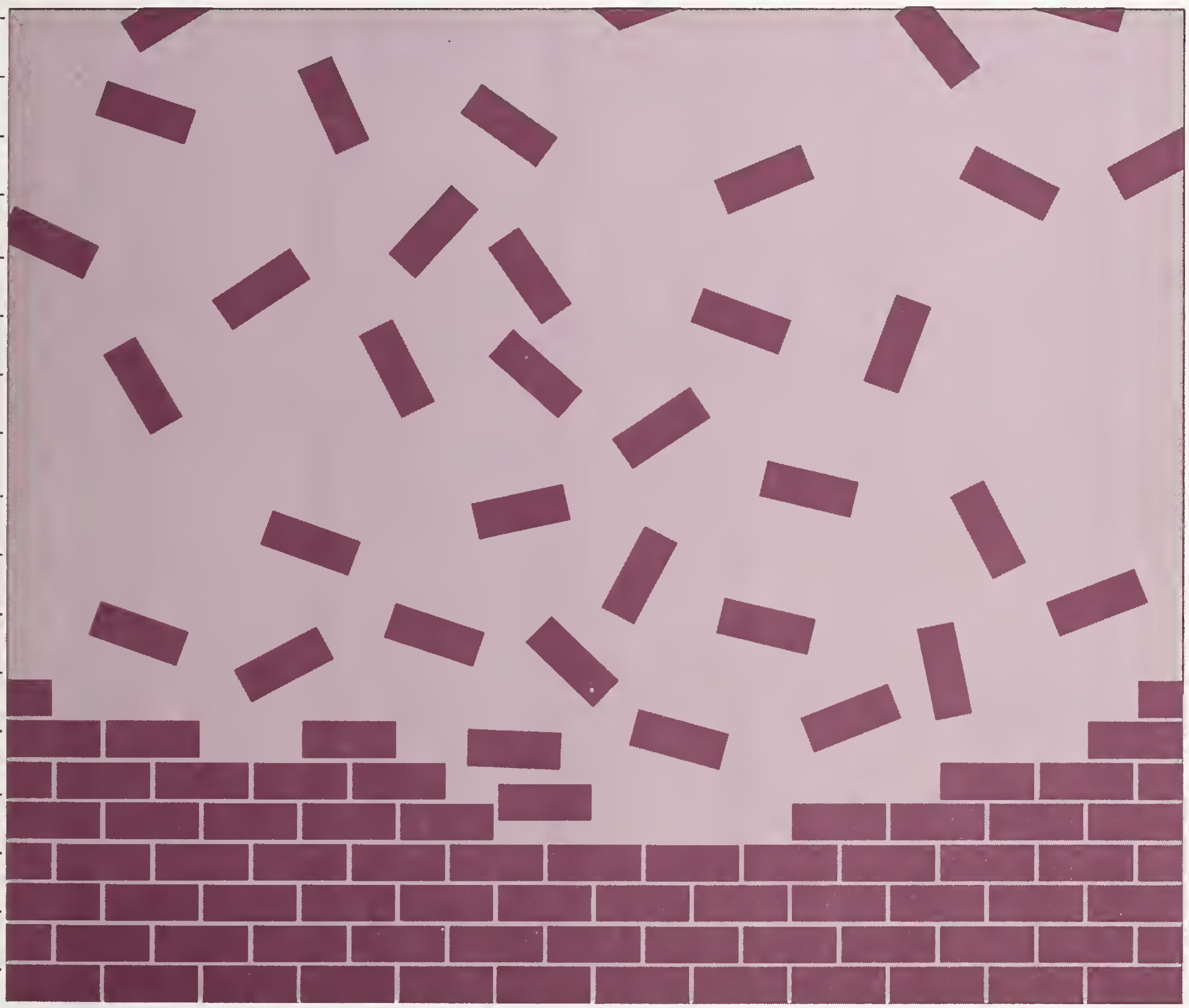
NIST Special Publication 800-17

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology

Modes of Operation Validation System (MOVS): Requirements and Procedures

Sharon Keller and Miles Smid

COMPUTER SECURITY



NIST

QC
100
U57
0.800-17
998

The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering, and develops measurement techniques, test methods, standards, and related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Publications and Program Inquiries Desk, 301-975-3058.

Office of the Director
- National Quality Program
- International and Academic Affairs

Technology Services
- Standards Services
- Technology Partnerships
- Measurement Services
- Technology Innovation
- Information Services

Advanced Technology Program
- Economic Assessment
- Information Technology and Applications
- Chemical and Biomedical Technology
- Materials and Manufacturing Technology
- Electronics and Photonics Technology

Manufacturing Extension Partnership Program
- Regional Programs
- National Programs
- Program Development

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]
- Optoelectronics[1]

Chemical Science and Technology Laboratory
- Biotechnology
- Physical and Chemical Properties[2]
- Analytical Chemistry
- Process Measurements
- Surface and Microanalysis Science

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Optical Technology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- NIST Center for Neutron Research

Manufacturing Engineering Laboratory
- Precision Engineering
- Automated Production Technology
- Intelligent Systems
- Fabrication Technology
- Manufacturing Systems Integration

Building and Fire Research Laboratory
- Structures
- Building Materials
- Building Environment
- Fire Safety Engineering
- Fire Science

Information Technology Laboratory
- Mathematical and Computational Sciences[2]
- Advanced Network Technologies
- Computer Security
- Information Access and User Interfaces
- High Performance Systems and Services
- Distributed Computing and Information Services
- Software Diagnostics and Conformance Testing

[1]At Boulder, CO 80303.
[2]Some elements at Boulder, CO.

NIST Special Publication 800-17

Modes of Operation Validation System (MOVS): Requirements and Procedures

Sharon Keller and Miles Smid

COMPUTER SECURITY

Information Technology Laboratory

National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

February 1998

U.S. Department of Commerce
William M. Daley, Secretary

Technology Administration
Gary R. Bachula, Acting Under Secretary for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure for information technology. ITL develops tests, test methods, reference data, proof of concept implementations and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in federal computer systems. This Special Publication 800 series reports on ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-17
Natl. Inst. Stand. Technol. Spec. Publ. 800-17, 152 pages (Feb. 1998)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1998

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

CBC . . . . Cipher Block Chaining Mode

CMT . . . . Cryptographic Module Testing Laboratory

CMV . . . . NIST Cryptographic Module Validation Program

CFB . . . . Cipher Feed Back Mode

DES . . . . Data Encryption Standard

ECB . . . . Electronic Code Book Mode

EES . . . . Escrowed Encryption Standard

FIPS PUB . . . . Federal Information Processing Standard Publication

IUT . . . . Implementation Under Test

MOVS. . . . . Modes of Operation Validation System

NSA . . . . National Security Agency

NVLAP . . . . .NIST National Voluntary Laboratory Accreditation Program

NBS. . . . . National Bureau of Standards

NIST . . . . National Institute of Standards and Technology

OFB. . . . . Output Feed Back Mode

ACKNOWLEDGMENTS

The authors would like to thank Donna Dodson (NIST), Lisa Carnahan (NIST), Elaine Barker (NIST), and Jim Foti (NIST) for their significant assistance in the development of this Special Publication.

ABSTRACT

The National Institute of Standards and Technology (NIST) Modes of Operation Validation System (MOVS) specifies the procedures involved in validating implementations of the DES algorithm in FIPS PUB 46-2 *The Data Encryption Standard (DES)* and the Skipjack algorithm in FIPS PUB 185, *Escrowed Encryption Standard (ESS)*. The MOVS is designed to perform automated testing on Implementations Under Test (IUTs). This publication provides brief overviews of the DES and Skipjack algorithms and introduces the basic design and configuration of the MOVS. Included in this overview are the specifications for the two categories of tests which make up the MOVS, i.e., the Known Answer tests and the Modes tests. The requirements and administrative procedures to be followed by those seeking formal NIST validation of an implementation of the DES or Skipjack algorithm are presented. The requirements described include the specific protocols for communication between the IUT and the MOVS, the types of tests which the IUT must pass for formal NIST validation, and general instructions for accessing and interfacing with the MOVS. An appendix with tables of values and results for the DES and Skipjack Known Answer tests is also provided.

Key words: automated testing, computer security, cryptographic algorithms, cryptography, Data Encryption Standard (DES), Federal Information Processing Standard (FIPS), NVLAP, Skipjack algorithm, secret key cryptography, validation.

1. INTRODUCTION

1.1 Background

This publication specifies the various tests required to validate implementations under test (IUTs) for conformance to the DES and Skipjack algorithms. When applied to IUTs of the DES algorithm, the Modes of Operation Validation System (MOVS) provides conformance testing for the various components of the algorithm, as well as testing for apparent operational errors. The MOVS is also used to test for apparent operational errors in IUTs of the Skipjack algorithm.

The MOVS is composed of two types of validation tests, the Known Answer tests and the Modes tests. Both of these are based on validation tests described in SP500-20, *Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard.* As SP500-20's title implies, the validation tests were written to validate hardware implementations of the DES algorithm. SP800-17 expands on this by specifying how to validate implementations of the DES algorithm in software, firmware, hardware, or any combination thereof. The document also addresses implementations of the Skipjack algorithm, which must be implemented in electronic devices (e.g., very large scale integration chips). The Known Answer tests and Modes tests are based on the standard DES test set and the Monte-Carlo tests respectively, as specified in SP500-20.

To perform the Known Answer tests, the MOVS supplies known values to the IUT. The IUT then processes the input through the implemented algorithm, and the results are compared to expected values. When applied to IUTs of the DES algorithm, the Known Answer tests verify that the IUT correctly implements the components of the algorithm (e.g., S boxes, ...). When applied to IUTs of the Skipjack algorithm, these same tests verify that the implemented algorithm produces the correct results, i.e., given known input, the correct results are produced.

Since the test set used for the Known Answer tests is public knowledge, another type of validation test has been designed to use pseudo-random data. This test is the Modes test. The Modes test verifies that the IUT has not been designed just to pass the Known Answer tests. A successful series of Modes tests gives some assurance that an anomalous combination of inputs does not exist that would cause the test to end abnormally for reasons not directly related to the implementation of the algorithm. An additional purpose of the Modes test is to verify that no undesirable condition within the IUT will cause the key or plaintext to be exposed due to an implementation or operational error. The Modes test is not a reliability test, but merely checks for the presence of an apparent operational error.

1.2 Organization

Section 2 gives a brief overview of the DES and Skipjack algorithms and the four modes of operation allowed by both of these algorithms. Section 3 provides an overview of the tests which make up the Modes of Operation Validation System (MOVS) for the DES and Skipjack algorithms. Section 4 describes the basic protocol used by the MOVS. Section 5 provides a detailed explanation of each test required by the MOVS to validate an IUT of the DES and Skipjack algorithms. Section 6 outlines the design of the MOVS. Appendix A provides an example of round outputs for the DES, and Appendix B provides tables of values for the Known Answer tests for both the DES and Skipjack algorithms. These tables include Table 1 - Resulting Ciphertext from the Variable Plaintext Known Answer Test for DES, Table 2 - Resulting Ciphertext from the Variable Key Known Answer Test for DES, Table 3 - Values to be Used for the Permutation Operation Known Answer Test, Table 4 - Values to be Used for the Substitution Tables Known Answer Test, Table 5 - Resulting Ciphertext from the Variable Plaintext Known Answer Test for Skipjack, and Table 6 - Resulting Ciphertext from the Variable Key Known Answer Test for Skipjack.

2. PRIVATE KEY ALGORITHMS

2.1 Data Encryption Standard (DES) (FIPS PUB 46-2)

FIPS PUB 46-2, *The Data Encryption Standard (DES)*, published on December 30, 1993, is a cryptographic algorithm which has been standardized for use within the Federal Government for protecting the transmission and storage of unclassified computer data. DES is a FIPS approved cryptographic algorithm as required by FIPS 140-1, *Security Requirements for Cryptographic Modules, January 11, 1994.*

The DES algorithm is a recirculating, 64-bit, block product cipher whose security is based on a secret key. The DES keys are 64-bit binary vectors consisting of 56 information bits and 8 parity bits. The parity bits are reserved for error detection purposes and are not used by the encryption algorithm. The 56 information bits are used by the enciphering and deciphering operations and are referred to as the active key.

In the enciphering computation, a block to be enciphered is subjected to an initial permutation (IP), then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation (IP^{-1}). The key-dependent computation can be defined in terms of a function f, called the cipher function, and a function KS, called the key schedule. The function f involves E operators, substitution tables (S-boxes), and permutations (P). The 64 bit input block is divided into two halves, each consisting of 32 bits. One half is used as input to the function *f*, and the result is exclusive ORed to the other half. After one iteration, or round, the two halves of data are swapped, and the operation is performed again. The DES algorithm uses 16 rounds to produce a recirculating block product cipher. The cipher produced by the algorithm displays no correlation to the input. Every bit of the output depends on every bit of the input and on every bit of the active key. An example of round-by-round encryption for a given key and plaintext is shown in Appendix A.

For a thorough discussion of the DES algorithm and its components, consult FIPS PUB 46-2. Guidelines on the proper usage of the DES are published in FIPS PUB 74, *Guidelines for Implementing and Using the NBS Data Encryption Standard.* A brief description of the components of the DES algorithm follows.

2.1.1 The S-boxes

The non-linear substitution tables, or S-boxes, constitute an important part of the algorithm. The purpose of the S-boxes is to ensure that the algorithm is not linear. There are eight different S-boxes. Figure 2.1 displays one of these. Each S-box contains 64 entries, organized as a 4×16 matrix. Each entry is a four bit binary number, represented as 0-15. A particular entry in a single S-box is selected by six bits, two of which select a row and four select a column. The entry in the corresponding row and column is the output for that input. Each row in each S-box is a permutation of the numbers 0-15, so no entry is repeated in any one row. The output of the parallel connection of eight S-boxes is 32 bits.

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figure 2.1 ***One of the Eight S-Boxes in the DES***

2.1.2 The Key Schedule

The key schedule provides a linear means of thoroughly intermixing the bits of the 56-bit key specified for use in the DES operation to form a different 48-bit key for each of the 16 rounds of the DES algorithm. This is done in the following manner: The key is subjected to a permuted choice 1 (PC1) where the bits of the key are reorganized. The permuted key is then divided into two parts denoted C_i and D_i. These parts are shifted left a predetermined number of times producing C_{i+1} and D_{i+1}. The resulting values are subjected to a permuted choice 2 (PC2) which reorganizes the bits again, producing the round key K_{i+1}. To compute the next round key K_{i+2}, C_{i+1} and D_{i+1} are shifted left a predetermined number of times. The resulting value is then subjected to PC2. This procedure is repeated to calculate the 16 round keys.

Both the permutations in the key-schedule, PC1 and PC2, intermix the key bits among the round keys in such a way as to equalize key-bit utilization. It does this by forcing each key bit to be used no more than 15 times and no less than 12 times.

Figure 2.2 shows how the key schedule determines the sixteen 48-bit round keys from the 56-bit encryption key.

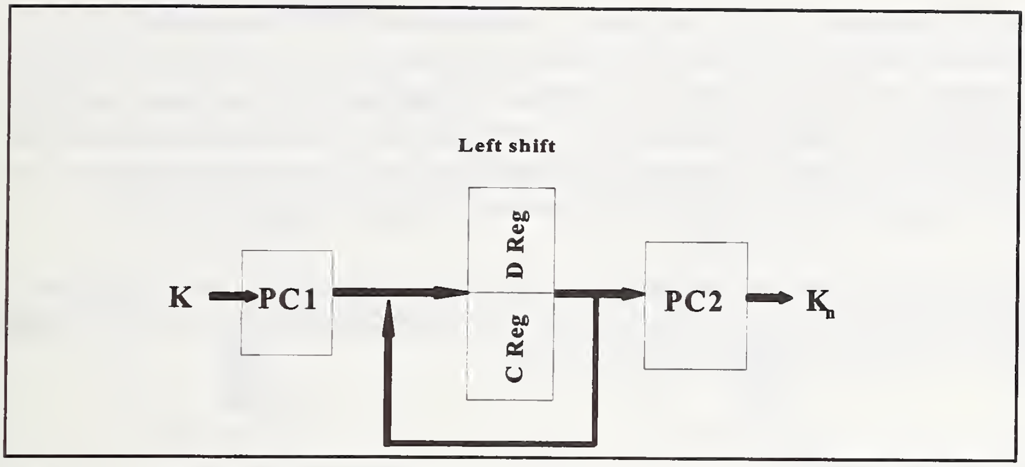


Figure 2.2 *The Key Schedule for the DES*

2.1.3 The Permutations and E Operator

The role of the permutation P is to thoroughly mix the data bits so they cannot be traced back through the S-boxes. The initial and final permutations are byte oriented, and the data is output eight bits at a time. The operator E expands a 32 bit input to a 48 bit output that is added mod two to the round key. The permutations in the key-schedule, PC1 and PC2, intermix the bits that result from the S-box substitution in a complex way to prevent bit tracing.

Each permutation is a linear operator, and so can be thought of as an $n \times m$ matrix and can be validated completely if it operates correctly on an appropriate maximal linearly independent set of input vectors, i.e., a suitable basis.

2.2 Skipjack Encryption Algorithm

The Skipjack algorithm is a classified symmetric-key cryptographic algorithm designed by the National Security Agency (NSA). The specifications for the Skipjack algorithm are contained in the R21 Informal Technical Report entitled "SKIPJACK" (S), R21-TECH-044-91, May 21, 1991. Organizations holding an appropriate security clearance and entering into a Memorandum of Agreement with the National Security Agency regarding implementations of the standard will be provided access to the classified specifications.

As discussed in FIPS PUB 185, *Escrowed Encryption Standard (ESS)*, the Skipjack algorithm has been approved for government applications requiring the encryption of sensitive but unclassified data telecommunications. The Skipjack algorithm is a 64-bit code book transformation that utilizes the same four DES modes of operation as specified in FIPS PUB 81, *DES Modes of Operation* and FIPS PUB 74, *Guidelines for Implementing and Using the NBS Data Encryption Standard*. Skipjack uses an 80-bit encryption/decryption key (compared with a 56-bit key used by DES) and has 32 rounds of processing per single encrypt/decrypt operation (compared with 16 rounds for the DES). Skipjack outputs 64 bits of output per round.

The Skipjack algorithm may only be implemented in electronic devices (e.g., very large scale integration chips). The devices may be incorporated in security equipment used to encrypt (and decrypt) sensitive unclassified telecommunications data.

2.3 The Four Modes of Operation

The DES and Skipjack algorithms both utilize the same four modes of operation specified in FIPS PUB 81, *DES Modes of Operation*. These modes are the Electronic Codebook (ECB) Mode, the Cipher Block Chaining (CBC) Mode, the Cipher Feedback (CFB) Mode, and the Output Feedback (OFB) Mode.

2.3.1 Electronic Codebook (ECB) Mode

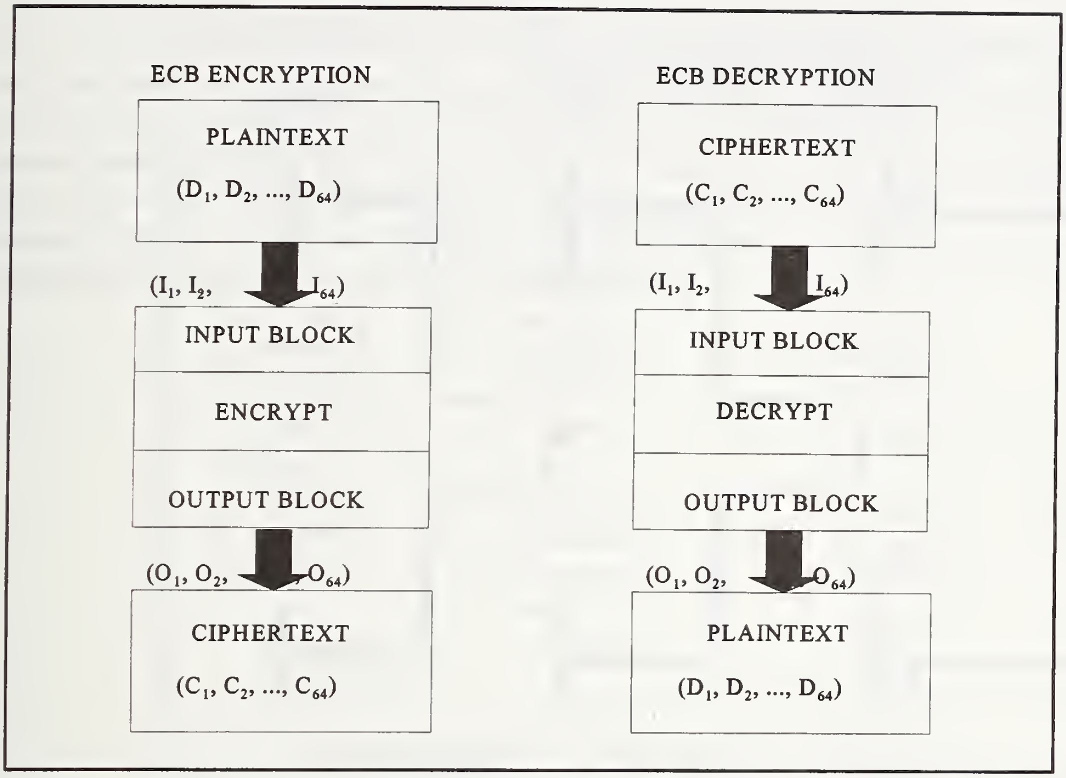


Figure 2.3 ***Electronic Codebook (ECB) Mode***

The Electronic Codebook (ECB) mode is shown in Figure 2.3. In ECB encryption, a plaintext data block (D_1, D_2, ..., D_{64}) is used directly as the input block (I_1, I_2, ..., I_{64}). The input block is processed through the DES or Skipjack algorithm in the encrypt state. The resultant output block (O_1, O_2,...,O_{64}) is used directly as ciphertext (C_1, C_2,..., C_{64}).

In ECB decryption, a ciphertext block (C_1, C_2, ..., C_{64}) is used directly as the input block (I_1, I_2,...,I_{64}). The input block is then processed through the DES or Skipjack algorithm in the decrypt state. The resultant output block (O_1, O_2,...,O_{64}) produces the plaintext (D_1,D_2,...,D_{64}). The ECB decryption process is the same as the ECB encryption process except that the decrypt state of the DES or Skipjack algorithm is used rather than the encrypt state.

2.3.2 Cipher Block Chaining (CBC) Mode

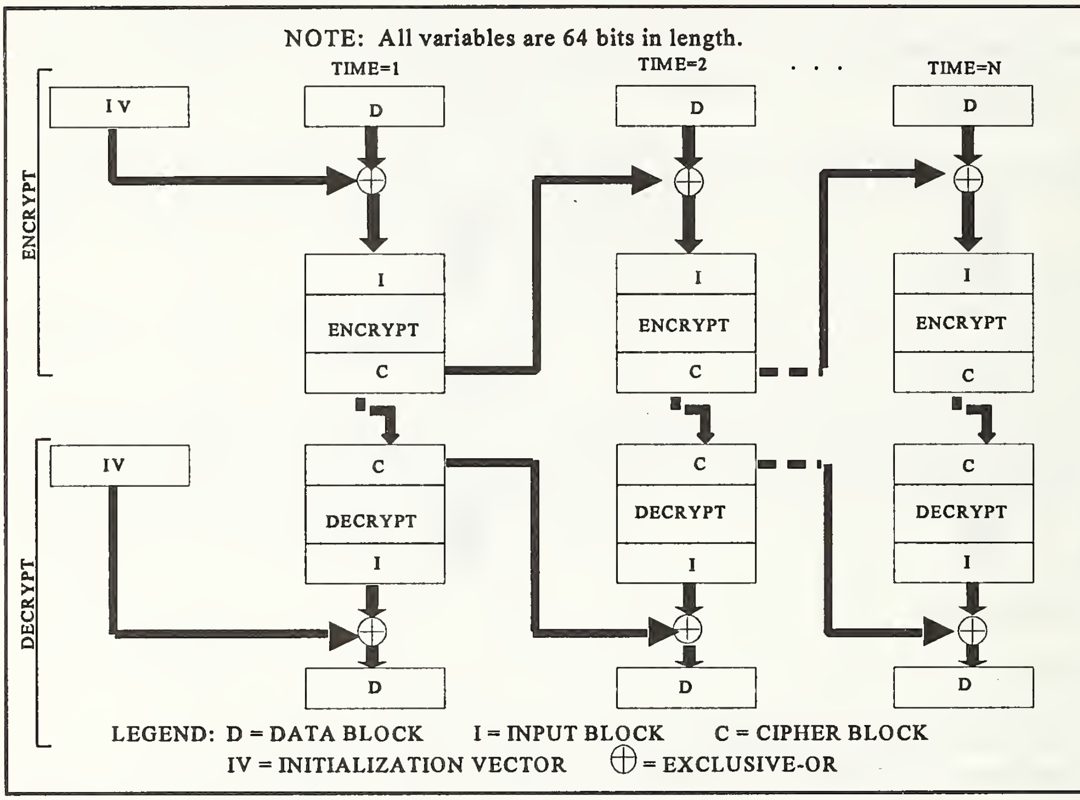


Figure 2.4 *Cipher Block Chaining (CBC) Mode*

As shown in the upper half of Figure 2.4, the Cipher Block Chaining (CBC) mode begins processing by dividing a plaintext message into 64 bit data blocks. In CBC encryption, the first input block ($I_1, I_2, \ldots, I_{64}$) is formed by exclusive-ORing the first plaintext data block ($D_1, D_2, \ldots, D_{64}$) with a 64-bit initialization vector IV, i.e., $(I_1,I_2,\ldots,I_{64}) = (IV_1 \oplus D_1, IV_2 \oplus D_2, \ldots IV_{64} \oplus D_{64})$. The input block is processed through the DES or Skipjack algorithm in the encrypt state, and the resulting output block is used as the ciphertext, i.e., $(C_1,C_2,\ldots,C_{64}) = (O_1,O_2,\ldots,O_{64})$. This first ciphertext block is then exclusive-ORed with the second plaintext data block to produce the second input block, i.e., $(I_1,I_2,\ldots,I_{64}) = (C_1 \oplus D_1, C_2 \oplus D_2, \ldots, C_{64} \oplus D_{64})$. Note that I and D now refer to the second block. The second input block is processed through the DES or Skipjack algorithm in the encrypt state to produce the second ciphertext block. This encryption process continues to "chain" successive cipher and plaintext blocks together until the last plaintext block in the message is encrypted. If the message does not consist of an integral number of data blocks, then the final partial data block should be encrypted in a manner specified for the application. One such method is described in Appendix C of FIPS PUB 81.

message is used as the input block and is processed through the DES or Skipjack algorithm in the decrypt state, i.e., $(I_1,I_2,...,I_{64}) = (C_1,C_2,...,C_{64})$. The resulting output block, which equals the original input block to the algorithm during encryption, is exclusive-ORed with the IV (which must be the same as that used during encryption) to produce the first plaintext block, i.e., $(D_1,D_2,...,D_{64}) = (O_1 \oplus IV_1, O_2 \oplus IV_2,...,O_{64} \oplus IV_{64})$. The second ciphertext block is then used as the next input block and is processed through the DES or Skipjack algorithm in the decrypt state. The resulting output block is exclusive-ORed with the first ciphertext block to produce the second plaintext data block, i.e., $(D_1,D_2,...,D_{64}) = (O_1 \oplus C_1, O_2 \oplus C_2,...,O_{64} \oplus C_{64})$. (Note D and O refer to the second block.) The CBC decryption process continues in this manner until the last complete ciphertext block has been decrypted. Ciphertext representing a partial data block must be decrypted in a manner as specified for the application.

2.3.3 Cipher Feedback (CFB) Mode

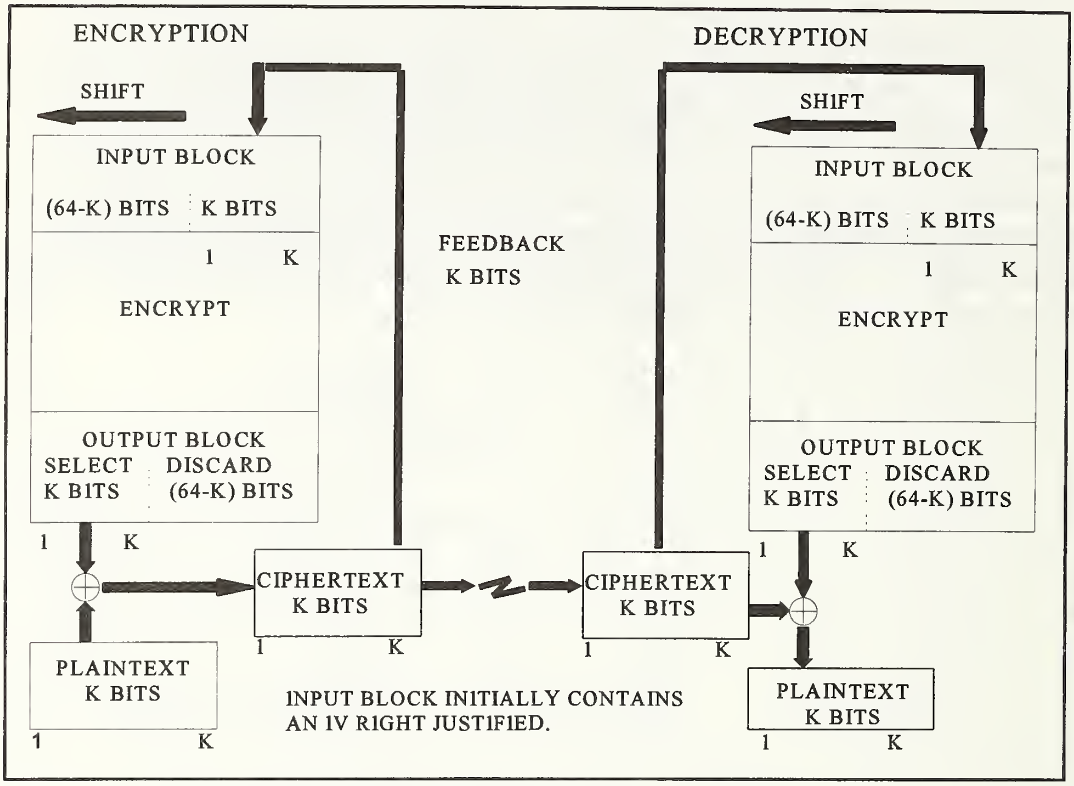


Figure 2.5 *Cipher Feedback (CFB) Mode*

The Cipher Feedback (CFB) mode is shown in Figure 2.5. A message to be encrypted is divided into K-bit data units, where K may equal 1 through 64 inclusively ($K = 1,2,...,64$). In both the CFB encrypt and decrypt operations, an initialization vector (IV) of length L is used, where L may equal 1 through 64 inclusively ($L=1,2,...,64$). The IV is placed in the least significant bits of the input block with the unused bits set to "0", i.e., $(I_1,I_2,...,I_{64}) = (0,0,...,0,IV_1,IV_2,...,IV_L)$. This input block is processed through the DES or Skipjack algorithm in the encrypt state to produce an output block. During encryption, ciphertext is produced by exclusive-ORing a K-bit plaintext data unit with the most significant K bits of the output block, i.e.,$(C_1,C_2,...,C_K) = (D_1 \oplus O_1, D_2 \oplus O_2, ..., D_K \oplus O_K)$. Similarly, during decryption, plaintext is produced by exclusive-ORing a K-bit unit of ciphertext with the most significant K bits of the output block, i.e., $(D_1,D_2,...,D_K) = (C_1 \oplus O_1,C_2 \oplus O_2,...,C_K \oplus O_K)$. In both cases the unused bits of the output block are discarded. For both the encryption and decryption processes, the next input block is created by discarding the most significant K bits of the previous input block, shifting the remaining bits K positions to the left and then inserting the K bits of ciphertext just produced in the encryption operation or just used in the decryption operation into the least significant bit positions, i.e., $(I_1,I_2,...,I_{64}) = (I_{[K+1]}, I_{[K+2]}, ... , I_{64},C_1,C_2, ...C_K)$. This input block is then processed through the DES or Skipjack algorithm in the encrypt state to produce the next output block. This process continues until the entire plaintext message has been encrypted or until the entire ciphertext message has been decrypted. For each operation of the DES or Skipjack algorithm, one K-bit unit of plaintext produces one K-bit unit of ciphertext, and one K-bit unit of ciphertext produces one K-bit unit of plaintext.

2.3.4 Output Feedback (OFB) Mode

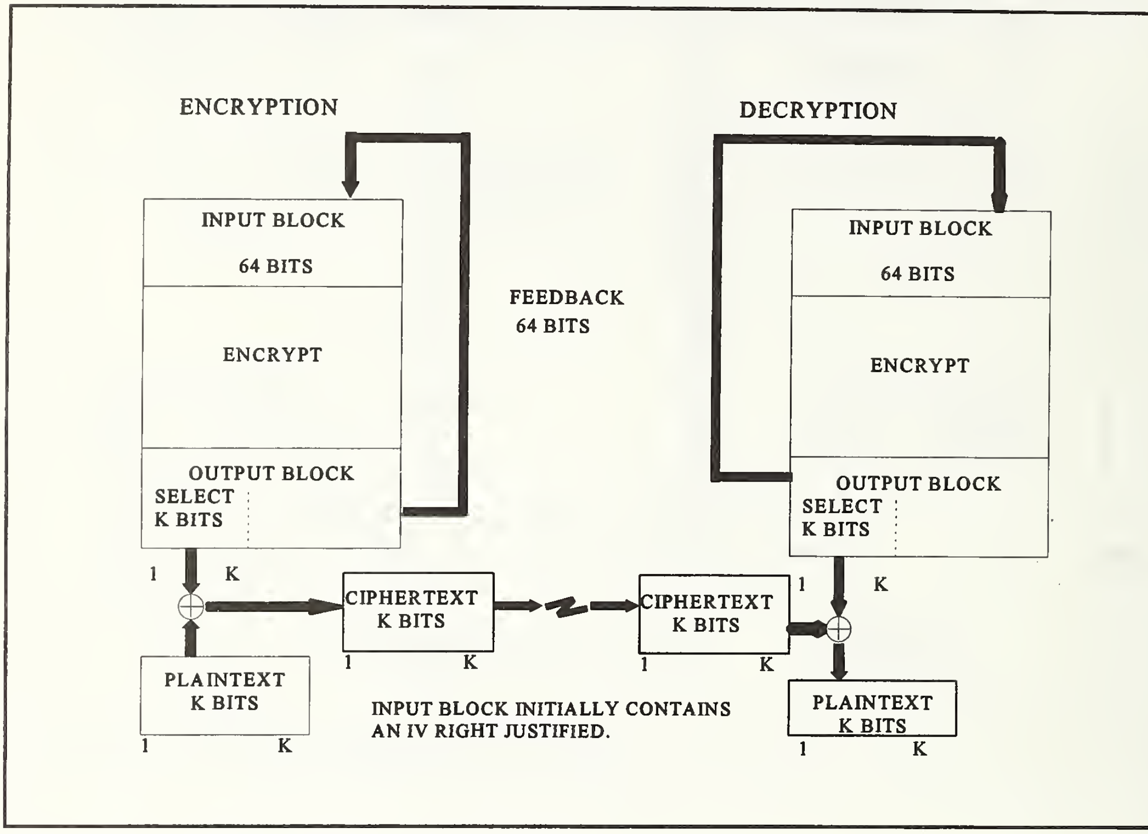


Figure 2.6 ***Output Feedback (OFB) Mode***

The Output Feedback (OFB) mode is shown in Figure 2.6. A message to be encrypted is divided into K-bit data units, where K may equal 1 through 64 inclusively, (K = 1,2,...,64). In both the OFB encrypt and decrypt operations, an initialization vector (IV) of length L is used, where L may equal 1 through 64 inclusively, (L=1,2,...,64). The IV is placed in the least significant bits of the input block with the unused bits set to "0", i.e., $(I_1,I_2,...,I_{64}) = (0,0,...,0,IV_1,IV_2,...,IV_L)$. This input block is processed through the DES or Skipjack algorithm in the encrypt state to produce an output block. During encryption, ciphertext is produced by exclusive-ORing a K-bit plaintext data unit with the most significant K bits of the output block, i.e., $(C_1,C_2,...,C_K) = (D_1 \oplus O_1, D_2 \oplus O_2, ...,D_K \oplus O_K)$. Similarly, during decryption, plaintext is produced by exclusive-ORing a K-bit unit of ciphertext with the most significant K bits of the output block, i.e., $(D_1,D_2,...,D_K) = (C_1 \oplus O_1,C_2 \oplus O_2,...,C_K \oplus O_K)$. In both cases the next input block is assigned the value of the output block, i.e., $(I_1,I_2,...,I_{64}) = (O_1,O_2, ...,O_{64})$. This input block is then processed through the DES or Skipjack algorithm in the encrypt state to produce the next output block. This process continues until the entire plaintext message has been encrypted or until the entire ciphertext message has been decrypted. For each operation of the DES or Skipjack algorithm, one K-bit unit of plaintext produces one K-bit unit of ciphertext or one K-bit unit of ciphertext produces one K-bit unit of plaintext.

Note that, originally, FIPS 81 allowed less than 64 bits of feedback to be used. It was discovered that when this is done, there is a risk of generating short cycles. That is, when the same key is used, and multiple encryptions or decryptions have occurred, then the resulting output block may be equal to an input block from a previous iteration. If that occurs, then further encryption or decryption using the same key will result in a repetition of previously generated output and input blocks. This increases the risk of a cryptanalyst recovering the original plaintext. Because of this short cycle property, NIST does not support the use of the OFB mode for any amount of feedback less than 64 bits. Note that this short cycle property is not a problem with the DES algorithm, and would occur using any block cipher in a similar manner.

3. MODES OF OPERATION VALIDATION SYSTEM FOR THE DES AND SKIPJACK ALGORITHMS

The MOVS for the DES and Skipjack algorithms consists of two types of tests, the Known Answer tests and the Modes tests. The MOVS provides conformance testing for the individual components of an IUT of the DES algorithm and analyzes IUTs of the DES and Skipjack algorithms for apparent operational errors. Note that the individual components of an IUT of the Skipjack algorithm are not tested by the MOVS since Skipjack is classified.

The IUTs of the DES algorithm may be written in software, firmware, hardware, or any combination thereof. The IUTs of the Skipjack algorithm must be implemented in electronic devices (e.g., very large scale integration chips). For the remainder of this document , the word *implementation* will reflect the definition pertaining to the algorithm being discussed.

An IUT must allow the MOVS to have control over the required input parameters for validation to be feasible. The ability to initialize or load known values to the variables required by a specific test may exist at the device level or the chip level in an IUT. If an IUT does not allow the MOVS to have control over the input parameter values, the MOVS tests cannot be performed.

An IUT may implement encryption only, decryption only, or both encryption and decryption. This will determine which MOVS tests will be performed by an IUT.

The following subsections provide an overview of the Known Answer tests and the Modes tests. Also discussed are the various tests required to validate IUTs of the DES and Skipjack algorithms.

3.1 The Known Answer Tests

The Known Answer tests are based on the standard DES test set discussed in SP500-20. When applied to IUTs of the DES algorithm, the Known Answer tests verify that the IUT correctly performs the algorithm. The tests also provide conformance testing for the following components of an IUT of the DES algorithm: the initial permutation IP, the inverse permutation IP^{-1}, the expansion matrix E, the data permutation P, the key permutations PC1 and PC2, and the substitution tables $S_1, S_2, \ldots, S_8$. When applied to IUTs of the Skipjack algorithm, these same tests verify that the implemented algorithm produces the correct results, i.e., given known input, the correct results are produced.

A generic overview of the sets of Known Answer tests required for the validation of IUTs implementing the encryption and/or decryption processes of all modes of operation for both the

DES and Skipjack algorithms are discussed below.

3.1.1 The Encryption Process

An IUT of the DES algorithm which allows encryption requires the successful completion of five Known Answer tests. These are the Variable Plaintext Known Answer test, the Inverse Permutation Known Answer test for the Encryption Process, 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. The Permutation Operation and the Substitution Table Known Answer tests do not apply to the Skipjack algorithm. Therefore, an IUT of the Skipjack algorithm which allows encryption requires only the successful completion of the Variable Plaintext Known Answer test, the Inverse Permutation Known Answer test for the Encryption Process, and the Variable Key Known Answer test for the Encryption Process.

These Known Answer tests are also used in the testing of IUTs implementing the CFB and OFB modes of operation in the decryption process. The reason for this is that both of these modes utilize the encrypt state in the decryption process.

3.1.1.1 The Variable Plaintext Known Answer Test

To perform the Variable Plaintext Known Answer test, the MOVS supplies the IUT with initial values for the plaintext and, if applicable, the initialization vector. These values are dependent upon the mode of operation being implemented. The key should be initialized to zero. Each block of data input into the DES or Skipjack algorithm is represented as a 64-bit basis vector. By definition, a basis vector is a vector consisting of a "1" in the i^{th} position and "0" in all of the other positions. The input block is processed through the algorithm in the encrypt state. The resulting output block is used in the calculation of the ciphertext which is then recorded. Each of the basis vectors is tested. At the completion of the 64^{th} test, all results are verified for correctness.

If correct results are obtained from an IUT of the DES algorithm, the Variable Plaintext Known Answer test has verified the initial permutation (IP) and the expansion matrix E by presenting a full set of basis vectors to the IP and to the E. If the results from each test of an IUT of the Skipjack algorithm match the expected results, the Skipjack algorithm has been verified.

3.1.1.2 The Inverse Permutation Known Answer Test for the Encrypt State

To perform the Inverse Permutation Known Answer test, the MOVS supplies the IUT with initial values for the plaintext and, if applicable, the initialization vector. The plaintext values are set to the ciphertext results obtained from the Variable Plaintext Known Answer test.

The key being used by this test is called 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, it is like encrypting and decrypting with the same key. The key should be initialized to zero, the same value used in the Variable Plaintext Known Answer test.

The input block is processed through the algorithm in the encrypt state. The resulting output block is used in the calculation of the ciphertext which is then recorded. The ciphertext should be the same as the plaintext used as input to the Variable Plaintext Known Answer test. At the completion of the 64^{th} test, all results are verified for correctness.

This test, when applied to an IUT of the DES algorithm, verifies the inverse permutation (IP^{-1}) by presenting each basis vector to the IP^{-1} as the basis vectors are recovered. If the results from each test of an IUT of the Skipjack algorithm match the expected results, the Skipjack algorithm has been verified.

3.1.1.3 The Variable Key Known Answer Test for the Encryption Process

To implement the Variable Key Known Answer test for the Encryption Process, the MOVS supplies the IUT with initial values for the key, the plaintext, and, if applicable, the initialization vector. During the initialization process, the plaintext and the initialization vector are set to zero. The key is initialized to an n-bit vector, where n is 56 if DES is being implemented, and 80 if Skipjack is being implemented. This vector will contain a "1" in the i^{th} significant position and "0"s is all remaining significant positions of a key where i = 1 to n. (Note that the parity bits are not counted in the significant bits. These parity bits may be "1"s or "0"s to maintain odd parity.) An input block is then formed according to the mode of the algorithm being implemented, and encrypted. The resulting output block is used in the calculation of the ciphertext which is recorded for later comparison. This test is repeated n times, allowing for every possible vector to be tested. At the completion of the n^{th} test, all results are verified for correctness.

When this test is performed for an IUT of the DES algorithm, the 56 possible basis vectors which yield unique keys are presented to PC1 verifying the key permutation, PC1. Since the key schedule consists of left shifts, as i ranges over the index set, a complete set of basis vectors is presented to PC2 as well, so this is verified. If the results from each test of an IUT of the Skipjack algorithm match the expected results, the Skipjack algorithm has been verified.

3.1.1.4 The Permutation Operation Known Answer Test for the Encryption Process

The Permutation Operation Known Answer test for the Encryption Process only applies to IUTs of the DES algorithm. To implement this test, the MOVS supplies the IUT with initial values for the key, the plaintext and, if applicable, the initialization vector, with the plaintext and initialization vector being set to zero. Based on the mode of operation of DES implemented, an input block is formed and encrypted. The resulting output block is used in the calculation of the ciphertext which is recorded for later comparison. This test is repeated 32 times, allowing for 32 given values to be tested. At the completion of the 32nd test, all results are verified for correctness.

This test presents a complete set of basis vectors to the permutation operator P. By doing so, P is verified.

3.1.1.5 The Substitution Table Known Answer Test for the Encryption Process

The Substitution Table Known Answer test for the Encryption Process only applies to IUTs of the DES algorithm. The MOVS supplies the IUT with initial values for the key, the plaintext and, if applicable, the initialization vector which is initialized to zero. Based on the mode of operation of DES implemented, an input block is formed and encrypted. The resulting output block is used in the calculation of the ciphertext which is recorded for later comparison. This test is repeated 19 times in order to process a set of 19 key-data pairs. At the completion of the 19th test, all results are verified for correctness.

The set of 19 key-data pairs used in this test result in every entry of all eight S-box substitution tables being used at least once. Thus, this test verifies the eight substitution tables of 64 entries each.

3.1.2 The Decryption Process

The five Known Answer tests required for validation of IUTs implementing the decryption process of the DES or Skipjack algorithms consist of the Variable Ciphertext Known Answer test, the Initial Permutation Known Answer test for the Decryption Process, 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 can only be performed by IUTs that support the Electronic Codebook (ECB) and the Cipher Block Chaining (CBC) modes of operation since only these modes of operation utilize the decrypt state during the decryption process. The CFB and OFB modes of operation utilize the encrypt state in the decryption process and therefore should be tested using the same Known Answer tests used for IUTs that support the encryption process. Only the Variable Ciphertext Known Answer test, the Initial Permutation Known Answer test for the Decryption Process, and the Variable Key Known Answer test for the Decryption Process apply to the Skipjack algorithm.

3.1.2.1 The Variable Ciphertext Known Answer Test

To perform the Variable Ciphertext Known Answer test, the values of the ciphertext, the key, and, if applicable, the initialization vector are initialized, with the key and the initialization vector being initialized to zero. If the IUT performs both encryption and decryption, the values resulting from the encryption performed in the Variable Plaintext Known Answer test will be used to initialize the ciphertext. Otherwise, the MOVS will supply the IUT with the ciphertext values.

The value of the ciphertext is used directly as the input block of data. The input block is processed through the algorithm in the decrypt state, resulting in an output block. The output block is used in the calculation of the plaintext which is then recorded. This test is repeated for 64 cycles and should result in a set of 64 different basis vectors. For IUTs of the DES algorithm, this test verifies the inverse permutation IP^{-1} by presenting the basis vectors to the IP^{-1} as they are recovered.

If the Skipjack algorithm is implemented and the IUT produces correct results (i.e., the basis vectors are recovered), this test ends successfully.

3.1.2.2 The Initial Permutation Known Answer Test for the Decryption Process

To perform the Initial Permutation Known Answer test for the Decryption Process, the values of the ciphertext are set to the resulting plaintext values obtained from the Variable Ciphertext Known Answer test. The key, and, if applicable, the initialization vector are set to the same values used in the Variable Ciphertext Known Answer test, i.e., they are set to zero.

The value of the ciphertext is used directly as the input block of data. The input block is processed through the algorithm in the decrypt state, resulting in an output block. The output block is used in the calculation of the plaintext which is then recorded. This test is repeated for 64 cycles and should result in the set of ciphertext values used as input to the Variable Ciphertext Known Answer test.

For IUTs of the DES algorithm, the initial permutation IP and the expansion matrix E are verified by presenting the full set of basis vectors to both of them.

If the Skipjack algorithm is implemented and the IUT produces correct results (i.e., the basis vectors are recovered), this test ends successfully.

3.1.2.3 The Variable Key Known Answer Test for the Decryption Process

To implement the Variable Key Known Answer test for the Decryption Process, the values of the ciphertext, key, and, if applicable, the initialization vector are initialized. The ciphertext is initialized in one of two ways. If the IUT performs 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. Otherwise, the IUT will obtain the ciphertext values from the MOVS. The IV is set to zero. The key is initialized to an n-bit vector, where n is 56 if DES is being implemented and 80 if Skipjack is being implemented. This vector will contain a "1" in the i^{th} significant position and "0"s is all remaining significant positions of a key where $i = 1$ to n. (Note that the parity bits are not counted in the significant bits. These parity bits may be "1"s or "0"s to maintain odd parity.)

The value of the ciphertext is used directly as the input block of data. The input block is processed through the algorithm in the decrypt state. According to the mode of operation supported by the IUT, the resulting output block is used in the calculation of the plaintext which is recorded for later comparison. This test is repeated n times allowing for every possible vector to be tested. At the completion of the n^{th} test, all results are verified against known values for correctness. If the results are correct for an IUT of the DES algorithm, it can be assumed that this test verifies the right shifts in the key schedule as the basis vectors are recovered.

If the results from each test of an IUT of the Skipjack algorithm match the expected results, the Skipjack algorithm has been verified.

3.1.2.4 The Permutation Operation Known Answer Test for the Decryption Process

The Permutation Operation Known Answer test for the Decryption Process only applies to IUTs of the DES algorithm. To implement this test, 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 MOVS. If applicable, the initialization vector will be set to zero.

The value of the ciphertext is used directly as the input block of data. The input block is processed through the algorithm in the decrypt state. According to the mode of operation supported by the IUT, the resulting output block is used in the calculation of the plaintext which is recorded for later comparison. This test is repeated 32 times allowing for the 32 key-ciphertext values to be tested. At completion, the results of each of the 32 tests is verified to be zero.

The 32 key values used in this test present a complete set of basis vectors to the permutation operator P. By doing so, P is verified.

3.1.2.5 The Substitution Table Known Answer Test for the Decryption Process

The Substitution Table Known Answer test for the Decryption Process only applies to IUTs of the DES algorithm. To implement this test, values for the key and ciphertext 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 MOVS. If applicable, the initialization vector will be set to zero.

The value of the ciphertext is used directly as the input block of data. This input block is processed through the algorithm in the decrypt state. Based on the mode of operation implemented by the IUT, the resulting output block is used in the calculation of the plaintext which is recorded for later comparison. This test is repeated 19 times in order to process the set of 19 key-data pairs that result in every entry of all eight substitution tables being used at least once. At the completion of the 19th test, all results are verified for correctness. If the IUT produces correct results, the eight S-box substitution tables of 64 entries each have been verified.

3.2 The Modes Test

The Modes test is the second type of validation test required to validate IUTs of the DES and Skipjack algorithms. The Modes test is based on the Monte-Carlo test discussed in SP500-20. They are designed to use pseudo-random data to verify that the IUT has not been designed just to pass the Known Answer tests. A successful series of Modes tests gives some assurance that an anomalous combination of inputs does not exist that would cause the test to end abnormally for reasons not directly related to the implementation of the algorithm. An additional purpose of the Modes test is to verify that no undesirable condition within the IUT will cause the key or plaintext to be exposed due to an implementation error. This test also checks for the presence of an apparent operational error.

The MOVS supplies the IUT with initial input values for the key, the plaintext (or ciphertext), and, if applicable, an initialization vector. The Modes test is then performed (as described in the following paragraph) and the resulting ciphertext (or plaintext) values are recorded and compared to known results. If an error is detected, the erroneous result is recorded, and the test terminates abnormally. Otherwise, the test continues. If the IUT's results are correct, the Modes test for the IUT ends successfully.

Each Modes test consists of four million cycles through the DES or Skipjack algorithm implemented in the IUT. These cycles are divided into four hundred groups of 10,000 iterations each. Each iteration consists of processing an input block through the DES or Skipjack algorithm resulting in an output block. At the 10,000th cycle in an iteration, new values are assigned to the variables needed for the next iteration. The results of each 10,000th encryption or decryption cycle are recorded and evaluated as specified in the preceding paragraph.

4. BASIC PROTOCOL

4.1 Overview

Input and output messages used to convey information between the MOVS and the IUT shall consist of specific fields. The format of these input and output messages is beyond the scope of this document and the testing laboratories have the option to determine the specific formats of those messages. However, the results sent to NIST must include certain minimum information, which is specified in Section 4.4 Output Types.

A separate message shall be created for each mode of operation supported by an IUT. The information shall indicate the algorithm used (DES or Skipjack), the mode of operation (ECB, CBC, CFB-including feedback amounts, or OFB), the state (encrypt and/or decrypt), the test being performed (one of the various Known Answer tests, or the Modes tests), and the required data fields. The required data may consist of counts, keys, initialization vectors, and data representing plaintext or ciphertext. Every field in an output message shall be clearly labeled to indicate its contents - this is especially important for NIST to be able to ensure that test results are complete.

4.1.1 Conventions

The following conventions shall be used in the data portion of messages between the MOVS and the IUT:

1. Integers: integers shall be unsigned and shall be represented in decimal notation. (See Section 4.1.2 for these notations.)

2. Hexadecimal strings: shall consist of ASCII hexadecimal characters. The ASCII hexadecimal characters to be used shall consist of the ASCII characters 0-9 and A-F (or a-f), which represent 4-bit binary values.

3. Characters: the characters to be represented are A-Z (or a-z), 0-9, and underscore (_).

4.1.2 Message Data Types

The following data types shall be used in messages between the MOVS and the IUT:

1. Decimal integers: a decimal integer shall have the form

   ddd ... dd

where each 'd' shall represent a decimal character (0-9); one or more characters shall be present. The characters must be contiguous.

2. Hexadecimal strings: a hexadecimal string shall have the form

   hhh ... hh

   where each 'h' shall represent an ASCII character 0-9 or A-F (or a-f). Each 'h' shall represent a 4-bit binary value.

3. Characters: an ASCII character shall have the form

   c

   where 'c' shall represent an ASCII character A-Z (or a-z), 0-9, and underscore (_).

4.2 Message Contents

The information included in a message shall consist of the following:

Algorithm - selections shall consist of DES or Skipjack,
Mode - selections shall consist of ECB, CBC, CFB-including feedback amounts, or OFB,
Process - selections shall consist of ENCRYPT or DECRYPT,
Test - selections shall consist of:

VTEXT for Variable Plaintext/Ciphertext Known Answer test
VKEY for Variable Key Known Answer test
INVPERM for Inverse Permutation Known Answer test
INITPERM for Initial Permutation Known Answer test
PERM for Permutation Operation Known Answer test
SUB for Substitution Table Known Answer test
MODES for Modes test

Input/Output Data

The contents of the input/output data included in a message shall depend on the algorithm, mode, process, and test being performed. These different combinations of data have been organized into input types and output types. The input types shall be used by the MOVS to supply data to the IUT for testing. The output types shall be used by the IUT to supply results from the tests to the MOVS, and eventually to NIST.

4.3 Input Types

Twelve different combinations of input data shall be used by the MOVS to support the various Known Answer tests and Modes tests .

4.3.1 Input Type 1

Input Type 1 shall consist of:

KEY and DATA

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant; and

DATA shall be a 16 character ASCII hexadecimal string representing plaintext if the encrypt process is being tested, or ciphertext if the decrypt process is being tested.

4.3.2 Input Type 2

Input Type 2 shall consist of:

KEY,IV, and DATA

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant;

IV shall be a 16 character ASCII hexadecimal string representing the 64-bit initialization vector; and

DATA shall be 1 to 64 binary bits represented as a 16 character ASCII hexadecimal string representing plaintext if the encrypt process is being tested, or ciphertext if the decrypt process is being tested.

4.3.3 Input Type 3

Input Type 3 shall consist of:

KEY,n,CT_1,CT_2,...CT_n

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant;

n is an integer which shall indicate the number of ciphertext (CT) values to follow; and

each CT_n shall be 1 to 64 binary bits represented as a 16 character ASCII hexadecimal string.

4.3.4 Input Type 4

Input Type 4 shall consist of:

KEY

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant.

4.3.5 Input Type 5

Input Type 5 shall consist of:

KEY,IV,n,$TEXT_1$,$TEXT_2$,...$TEXT_n$

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant;

IV shall be a 16 character ASCII hexadecimal string representing the 64-bit initialization vector;

n is an integer which shall indicate the number of TEXT values to follow; and

each TEXT_n shall be 1 to 64 binary bits represented as a 16 character ASCII hexadecimal string. TEXT shall represent PT, CT, or RESULT.

4.3.6 Input Type 6

Input Type 6 shall consist of:

KEY and IV

where KEY shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant; and

IV shall be a 16 character ASCII hexadecimal string representing the 64-bit initialization vector.

4.3.7 Input Type 7

Input Type 7 shall consist of

$\text{PT},\text{KEY}_1,\text{KEY}_2,\ldots\text{KEY}_{32}$

where PT shall be 1 to 64 binary bits represented as a 16 character ASCII hexadecimal string; and each KEY_i, where i=1 to 32, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant.

4.3.8 Input Type 8

Input Type 8 shall consist of:

$TEXT,IV,KEY_1,KEY_2,...KEY_{32}$

where TEXT shall be 1 to 64 binary bits represented as a 16 character ASCII hexadecimal string. (NOTE: TEXT may be referred to as plaintext or text.);

IV shall be a 16 character ASCII hexadecimal string representing the 64-bit initialization vector; and

each KEY_i, where i=1 to 32, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant.

4.3.9 Input Type 9

Input Type 9 supplies *n* key/input block pairs. It shall consist of:

$n,PAIR_1,PAIR_2,...PAIR_n$

In this input type, the integer *n* shall indicate the number of KEY values to follow. Each $PAIR_i$ shall consist of:

KEY_i and $TEXT_i$

where each KEY_i, where i=1 to n, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant; and

each TEXT_i, for $i = 1$ to n, shall be a 16 character ASCII hexadecimal string representing either plaintext or ciphertext.

4.3.10 Input Type 10

Input Type 10 shall consist of:

$n,\text{KEY}_1,\text{KEY}_2,\ldots\text{KEY}_n$

where n is an integer which shall indicate the number of KEY values to follow; and

each KEY_i, where i=1 to n, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant.

4.3.11 Input Type 11

Input Type 11 shall consist of:

$\text{INITVAL},n,\text{PAIR}_1,\text{PAIR}_2,\ldots\text{PAIR}_n$

where INITVAL shall be a 16 character ASCII hexadecimal string representing either the 64 bit IV or the TEXT, depending on the mode of operation implemented by the IUT. (NOTE: The TEXT may be referred to as plaintext, ciphertext, or text.);

n is an integer which shall indicate the number of KEY/INPUT PAIRs to follow.

Each PAIR_i shall consist of:

KEY_i and IB_i

where each KEY_i, where i=1 to n, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the

KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant; and

each IB_i shall be a 16 character ASCII hexadecimal string representing either the 64 bit IV, PT or CT, depending on the mode of operation implemented.

4.3.12 Input Type 12

Input Type 12 shall consist of:

INITVAL,n,KEY_1,KEY_2,...KEY_n

where INITVAL shall be a 16 character ASCII hexadecimal string representing either the 64 bit IV or the 64 bit TEXT depending on the mode of operation implemented by the IUT. (NOTE: The TEXT may be referred to as ciphertext.);

n is an integer which shall indicate the number of KEYS to follow; and

each KEY_i, where i=1 to n, shall be represented as k bits in hexadecimal notation (i.e., 4 bits per hexadecimal character). If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The 8 parity bits shall be present but ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be presented in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant.

4.4 Output Types

Two different combinations of output data are used by the MOVS to support the various Known Answer tests and Modes tests.

4.4.1 Output Type 1

Output Type 1 shall consist of:

COUNT,KEY,DATA, and RESULT

where COUNT shall be an integer between 1 and 400, i.e., 0 < COUNT <= 400, representing the output line;

KEY shall be represented as k bits in hexadecimal notation. If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The parity bits shall be ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be displayed in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant;

DATA shall be a 16 character hexadecimal string representing plaintext if the encrypt process is being tested or ciphertext if the decrypt process is being tested; and

RESULT shall be a 16 character hexadecimal string indicating the resulting value. Depending on the process of the IUT being tested, the resulting value shall represent ciphertext (if encrypting) or plaintext (if decrypting).

4.4.2 Output Type 2

Output Type 2 shall consist of:

COUNT,KEY,CV,DATA, and RESULT

where COUNT shall be an integer between 1 and 400, i.e., 0 < COUNT <= 400, representing the output line;

KEY shall be represented as k bits in hexadecimal notation. If the IUT implements the DES algorithm, the KEY shall consist of 16 hexadecimal characters (i.e., 64 bits, k = 64). The parity bits shall be ignored, yielding 56 significant bits. For consistency purposes, the DES key shall be displayed in odd parity. If the IUT implements the Skipjack algorithm, the KEY shall consist of 20 hexadecimal characters (i.e. 80 bits, k = 80). Skipjack does not check parity, thus every bit in the key is significant;

CV shall be a 16 character ASCII hexadecimal string;

DATA shall be a 16 character hexadecimal string representing plaintext if the encrypt process is being tested or ciphertext if the decrypt process is being tested.; and

RESULT shall be a 16 character hexadecimal string indicating the resulting value. Depending on the process of the IUT being tested, the resulting value may be ciphertext (if encrypting) or plaintext (if decrypting).

5. TESTS REQUIRED TO VALIDATE AN IMPLEMENTATION OF THE DES OR SKIPJACK ALGORITHM

The validation of IUTs of the DES and Skipjack algorithms shall require the successful completion of an applicable set of Known Answer tests and the successful completion of the appropriate Modes tests. The tests required for validation of an IUT shall be determined by several factors. These include the algorithm implemented (DES or Skipjack), the mode(s) of operation supported (ECB, CBC, CFB, OFB), and the allowed cryptographic processes (encryption, decryption, both).

A separate set of Known Answer tests has been designed for use with each of the four modes of DES and Skipjack. Within these sets of tests are separate subsets of tests corresponding to the encrypt and decrypt processes. If an IUT implements multiple modes of operation but does not implement the ECB mode, each supported mode of operation shall be tested. If an IUT implements multiple modes of operation which does include the ECB mode, the set of Known Answer tests corresponding to the implemented cryptographic state of the ECB mode of operation shall be the only set of Known Answer tests conducted. The reasoning behind this is that other modes of operation implemented should follow the same logic as that for the ECB mode of operation.

The Modes tests have been designed for use with each of the four modes of DES and Skipjack. For the ECB, CBC, and CFB modes of operation, there are two tests associated with each: one to be used for IUTs allowing the encryption process and the other to be used for IUTs allowing the decryption process. If both the encryption and decryption processes are allowed by an IUT, both tests shall be required. The OFB mode of operation only requires one Modes test which is designed for use with both the encryption and decryption processes of an IUT. For example, if an IUT implements the CBC mode of operation in the encryption process only, the Modes test for the encryption process of the CBC mode of operation shall be successfully completed to validate the IUT. Likewise, if an IUT implements both the encryption and decryption processes of the CFB mode of operation, both the Modes test for the CFB encryption process and the Modes test for the CFB decryption process shall be successfully completed to validate the IUT. If an IUT implements both the encryption and decryption processes of the OFB mode of operation, the Modes test for the OFB mode of operation shall be successfully completed to validate the IUT.

If an IUT of the DES or Skipjack algorithm supports more than one mode of operation, the Modes test corresponding to each supported mode shall be performed successfully. For example, if an IUT implements the ECB and CBC modes of operation for the encryption process, the Modes test for the encryption process of the ECB mode of operation and the Modes test for the encryption process of the CBC mode of operation shall be successfully completed to validate the IUT.

The tests required to successfully validate IUTs of the DES and Skipjack algorithms are detailed in the following sections. These sections are categorized by mode of operation. Within each mode of operation, the tests are divided into tests to use with the encryption process and tests to use with the decryption process.

5.1 Electronic Codebook (ECB) Mode

The IUTs of the DES or Skipjack algorithm in the Electronic Codebook (ECB) mode shall be validated by the successful completion of a series of Known Answer tests and Modes tests corresponding to the cryptographic processes allowed by the IUT.

5.1.1 Encryption Process

The process of validating an IUT of the DES algorithm which implements the encryption process of the ECB mode of operation shall involve the successful completion of the following six tests:

1. The Variable Plaintext Known Answer Test - ECB mode
2. The Inverse Permutation Known Answer Test for the Encryption Process - 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
6. The Modes Test for the Encryption Process - ECB mode

The validation process for an IUT of the Skipjack algorithm which implements the encryption process of the ECB mode of operation shall require the successful completion of tests 1,2,3, and 6 only.

An explanation of the tests follows.

5.1.1.1 The Variable Plaintext Known Answer Test - ECB Mode

```
MOVS: Initialize          KEY:    If DES, KEY=0101010101010101 (odd parity set)
                                  If Skipjack, KEY=00000000000000000000
                   PT1 = 8000000000000000
            Send   KEY, PT1

IUT:        FOR i = 1 to 64
              {
                   IBi = PTi
                   Perform algorithm in encrypt state, resulting in CTi
                   Send i, KEY, PTi, CTi
                   PTi+1 = basis vector where single "1" bit is in position I+1
              }

MOVS:  Compare results from each loop with known answers
       If DES, use Appendix B, Table 1.  If Skipjack, use Appendix B, Table 5.
```

Figure 5.1 ***The Variable Plaintext Known Answer Test - ECB Mode***

Figure 5.1 illustrates the Variable Plaintext Known Answer test for the ECB mode of operation.

1. The MOVS shall:

    a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

       For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

    b. Initialize the 64 bit plaintext PT_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $PT_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

    c. Forward this information to the IUT using Input Type 1.

2. The IUT shall perform the following for i=1 through 64:

    a. Set the input block IB_i equal to the value of PT_i, i.e, $(IB1_i,IB2_i,...IB64_i)$ =

($PT1_i, PT2_i, \ldots, PT64_i$).

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current values of the loop number i, KEY, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 1.

d. Retain CT_i for use with the Inverse Permutation Known Answer test for the ECB Mode (Section 5.1.1.2), and, if the IUT supports the decryption process, for use with the Variable Ciphertext Known Answer test for the ECB Mode (Section 5.1.2.1).

e. Assign a new value to PT_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1, where i+1=2..64.

NOTE: This continues until every possible basis vector has been represented by the PT, i.e. 64 times. The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 1 for DES or Table 5 for Skipjack.

5.1.1.2 The Inverse Permutation Known Answer Test - ECB Mode

MOVS: Initialize KEY: If DES, KEY=0101010101010101 (odd parity set)
If Skipjack, KEY=00000000000000000000
PT_i (where i=1-64) = 64 CT values from the Variable Plaintext Known Answer test
Send KEY, 64, PT_1 ... PT_{64}

IUT: FOR i = 1 to 64
{
$IB_i = PT_i$
Perform algorithm in encrypt state, resulting in CT_i
Send i, KEY, PT_i, CT_i
PT_{i+1} = corresponding PT_{i+1} from MOVS
}

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

Figure 5.2 ***The Inverse Permutation Known Answer Test - ECB Mode***

Figure 5.2 illustrates the Inverse Permutation Known Answer test for the ECB mode of operation.

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit plaintext values PT_i (where i=1- 64) to the CT_i results obtained from the Variable Plaintext Known Answer test.

   c. Forward this information to the IUT using Input Type 3.

2. The IUT shall perform the following for i=1 through 64:

a. Set the input block IB_i equal to the value of PT_i, i.e, $(IB1_i, IB2_i, \ldots IB64_i) = (PT1_i, PT2_i, \ldots, PT64_i)$.

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current values of the loop number i, KEY, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 1.

d. Assign a new value to PT_{i+1} by setting it equal to the corresponding output from the Variable Plaintext Known Answer test for the ECB mode.

NOTE: The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values. The CT values should be the set of basis vectors.

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

```
MOVS: Initialize      KEY1:   If DES, KEY1 = 8001010101010101 (with odd parity)
                              If Skipjack, KEY1 = 80000000000000000000
                      PT=0000000000000000
      Send            KEY1, PT

IUT:  FOR i= 1 to n, where n = 64 if DES, 80 if Skipjack
          {
                      IF (algorithm == SKIPJACK) {process every bit}
                      OR
                         (algorithm == DES AND I %8 != 0)
                              {process every bit except parity bits}
                      {
                            IBi = PT
                            Perform algorithm in encrypt state using KEYi, resulting in CTi
                            Send i, KEYi, PT, CTi
                            KEYi+1 = vector consisting of "0" in every significant bit position
                            except for a single "1" bit in position i+1.  Each parity bit may have
                            the value "1" or "0" to make the KEY odd parity.
                      }
          }

MOVS: Compare results of the n encryptions with known answers
      For DES, use Appendix B, Table 2.  For Skipjack, use Appendix B, Table 6.
```

Figure 5.3 ***The Variable Key Known Answer Test for the Encryption Process- ECB Mode***

As summarized in Figure 5.3, the Variable Key Known Answer test for the ECB Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY_1 to contain "0" in every significant bit except for a "1" in the first position. For example, if validating an IUT of the DES algorithm, the 64 bit $KEY_{1\ bin}$ = 10000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01. Note that the parity bits are set to "0" or "1" to get odd parity.

      If validating an IUT of the Skipjack algorithm, the 80 bit $KEY_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80

00 00 00 00 00 00 00 00 00.

b. Initialize the 64 bit plaintext PT to the value of 0, i.e., PT_{hex}=00 00 00 00 00 00 00 00.

c. Forward this information to the IUT using Input Type 1.

2. The IUT shall perform the following for i= 1 to *n*: (NOTE: *n* equals the number of significant bits in a DES or Skipjack key.)

   a. Set the input block IB_i equal to the value of PT, i.e, $(IB1_i, IB2_i, \ldots IB64_i)$ = (PT1,PT2,...,PT64).

   b. Using the corresponding KEY_i, process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

   c. Forward the current values of the loop number i, KEY_i, PT, and the resulting CT_i to the MOVS as specified in Output Type 1.

   d. If the IUT supports the decryption process, retain CT_i for use with the Variable Key Known Answer test for the Decryption Process for the ECB Mode (Section 5.1.2.3).

   e. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in position i+1. The parity bits may contain "1" or "0" to make odd parity.

   NOTE: The above processing continues until every significant basis vector has been represented by the KEY parameter. The output from the IUT for this test shall consist of 56 output strings if DES is implemented and 80 output strings if Skipjack is implemented. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 2 for DES, or Table 6 for Skipjack.

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

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize        KEY_i (where i= 1-32) = 32 KEY values in Appendix B, Table 3
                        PT = 0000000000000000
      Send              PT, 32, KEY_1, KEY_2....,KEY_32
IUT:  FOR i = 1 to 32
       {
            IB_i = PT_i
            Perform DES algorithm in encrypt state using KEY_i, resulting in CT_i
            Send i, KEY_i, PT, CT_i
            KEY_i+1 = KEY_i+1 from MOVS
       }

MOVS: Compare results with known answers
```

Figure 5.4 ***The Permutation Operation Known Answer Test for the Encryption Process - ECB Mode***

Figure 5.4 illustrates the Permutation Operation Known Answer test for the ECB Encryption Process.

1. The MOVS shall:

   a. Initialize the KEY with the 32 constant KEY values from Appendix B, Table 3.

   b. Initialize the plaintext PT to the value of 0, i.e., PT_{hex}= 00 00 00 00 00 00 00 00.

   c. Forward this information to the IUT using Input Type 7.

2. The IUT shall perform the following for i= 1 to 32:

   a. Set the input block IB_i equal to the value of PT, i.e, ($IB1_i$,$IB2_i$,...$IB64_i$) = (PT1,PT2,...,PT64).

   b. Using the corresponding KEY_i, process IB_i through the DES algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current values of the loop number i, KEY_i, PT, and the resulting CT_i to the MOVS as specified in Output Type 1.

d. If the IUT supports the decryption process, retain CT_i for use with the Permutation Operation Known Answer test for the Decryption Process for the ECB mode (Section 5.1.2.4).

e. Set KEY_{i+1} equal to the next KEY supplied by the MOVS.

NOTE: The above processing shall continue until all 32 KEY values are processed. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 3.

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

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize    KEY_i (where i=1-19) = 19 KEY values in Appendix B, Table 4
                    PT_i (where i=1-19) = 19 corresponding PT values in Table 4
      Send          19, KEY_1, PT_1, KEY_2, PT_2,..., KEY_19, PT_19

IUT:  FOR i= 1 to 19
       {
              IB_i = PT
              Perform DES algorithm in encrypt state resulting in CT_i
              Send i, KEY_i, PT_i, CT_i
              KEY_i+1 = KEY_i+1 from MOVS
              PT_i+1 = PT_i+1 from MOVS
       }

MOVS: Compare results with known answers
```

Figure 5.5 ***The Substitution Table Known Answer Test for the Encryption Process - ECB Mode***

As summarized in Figure 5.5, the Substitution Table Known Answer test for the ECB Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY-plaintext (KEY-PT) pairs with the 19 constant KEY-PT values from Appendix B, Table 4.

   b. Forward this information to the IUT using Input Type 9.

2. The IUT shall perform the following for i= 1 to 19:

   a. Set the input block IB_i equal to the value of PT_i, i.e, $(IB1_i, IB2_i, \ldots IB64_i) = (PT1_i, PT2_i, \ldots, PT64_i)$.

   b. Using the corresponding KEY_i, process IB_i through the DES algorithm in the encrypt state, resulting in ciphertext CT_i.

   c. Forward the current values of the loop number i, KEY_i, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 1.

d. If the IUT supports the decryption process, retain CT_i for use with the Substitution Table Known Answer test for the Decryption Process for the ECB mode (Section 5.1.2.5).

e. Set KEY_{i+1} equal to the next KEY supplied by MOVS.

f. Set PT_{i+1} equal to the corresponding PT supplied by MOVS.

NOTE: The above processing shall continue until all 19 KEY-PT pairs are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 4.

5.1.1.6 Modes Test for the Encryption Process - ECB Mode

```
MOVS: Initialize KEY0, PT0
      Send     KEY0, PT0

IUT:  FOR i= 0 TO 399
        {
             Record i, KEYi, PT0
             FOR j = 0 TO 9,999
               {
                    IBj = PTj
                    Perform algorithm in encrypt state, resulting in CTj
                    PTj+1 = CTj
               }
             Record CTj
             Send i, KEYi, PT0, CTj
             KEYi+1= KEYi ⊕ last n bits of CT, where n=64 if DES, n=80 if Skipjack
             PT0 = CT9999
        }

MOVS: Check IUT's output for correctness
```

Figure 5.6 ***The Modes Test for the Encryption Process - ECB Mode***

As summarized in Figure 5.6, the Modes test for the ECB Encryption Process shall be performed as follows:

1. The MOVS shall:

    a. Initialize the KEY and plaintext PT variables. The PT shall consist of 64 bits, while the KEY length shall be dependent on the algorithm implemented by the IUT.

    b. Forward this information to the IUT using Input Type 1.

2. The IUT shall perform the following for i= 0 through 399:

    a. Record the current values of the outer loop number i, KEY_i, and PT_0.

b. Perform the following for j=0 through 9999:

I. Set the input block IB_j equal to the value of PT_j, i.e., $(IB1_j, IB2_j, ..., IB64_j) = (PT1_j, PT2_j,..., PT64_j)$.

ii. Process IB_j through the DES or Skipjack algorithm in the encrypt state resulting in CT_j.

iii. Prepare for loop j+1 by assigning PT_{j+1} with the current value of CT_j, i.e., $(PT1_{j+1}, PT2_{j+1}, ... PT64_{j+1}) = (CT1_j, CT2_j, ..., CT64_j)$.

c. Record CT_j .

d. Forward all recorded information for this loop, as specified in Output Type 1, to the MOVS.

e. Assign a new value to KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY with the current CT. For IUTs of the DES algorithm, this shall equate to $(KEY1_{i+1}, KEY2_{i+1}, ... KEY64_{i+1}) = (KEY1_i \oplus CT1_{9999}, KEY2_i \oplus CT2_{9999}, ... KEY64_i \oplus CT64_{9999})$.

For IUTs of the Skipjack algorithm, CT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished by concatenating the 16 rightmost bits of the previous CT (CT_{9998}) with the 64 bits of the current CT (CT_{9999}). This value shall then be exclusive-ORed with the current KEY to form the new KEY, i.e., $(KEY1_{i+1}, KEY2_{i+1}, ... KEY80_{i+1}) = (KEY1_i \oplus CT49_{9998}, KEY2_i \oplus CT50_{9998}, ... KEY16_i \oplus CT64_{9998}, KEY17_i \oplus CT1_{9999}, KEY18_i \oplus CT2_{9999}, ... KEY80_i \oplus CT64_{9999})$.

f. Assign a new value to PT in preparation for the next outer loop. PT_0 shall be assigned the value of the current CT, i.e., $(PT1_0, PT2_0,...,PT64_0) = (CT1_{9999}, CT2_{9999},...,CT64_{9999})$. (Note that the new PT shall be denoted as PT_0 to be used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.1.2 Decryption Process

The process of validating an IUT for the ECB mode of the DES algorithm which implements the decryption process shall involve the successful completion of the following six tests:

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

The validation process for an IUT of the Skipjack algorithm using the ECB mode of operation in the decryption process shall require the successful completion of tests 1, 2, 3, and 6 only.

An explanation of the tests follows.

5.1.2.1 The Variable Ciphertext Known Answer Test - ECB Mode

```
MOVS: Initialize KEY:   If DES, KEY=0101010101010101 (odd parity set)
                        If Skipjack, KEY=00000000000000000000

        If encryption is supported by IUT:
                Send KEY
        If encryption is not supported by IUT:
                Initialize CT values:  If DES, use values in Appendix B, Table 1
                                       If Skipjack, use values in Appendix B, Table 5
                Send KEY, 64, CT1, CT2,...CT64

IUT:    If encryption is supported by IUT:
                Initialize CT1 = first value from output of Variable Plaintext Known Answer test.
        Otherwise, use the first value received from the MOVS.

        FOR i = 1 to 64
        {
                IBi = CTi
                Perform algorithm in decrypt state, resulting in PTi
                Send i, KEY, CTi, PTi
                If encryption is supported:
                        CTi+1= corresponding CTi+1 from output of Variable Plaintext Known Answer
                        test
                else
                        CTi+1= the corresponding CTi+1 value from MOVS
        }

MOVS: Compare results from each loop with known answers
```

Figure 5.7 ***The Variable Ciphertext Known Answer Test - ECB Mode***

As summarized in Figure 5.7, the Variable Ciphertext Known Answer test for the ECB Mode of Operation shall be performed as follows:

1. The MOVS shall:

    a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity. For IUTs of the Skipjack algorithm, KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

    b. If the IUT implements the DES algorithm and it does not support encryption, initialize the 64 ciphertext CT values with the 64 constant CT values from

Appendix B, Table 1. Likewise, if the IUT is of the Skipjack algorithm, and it does not support encryption, initialize the 64 ciphertext CT values with the 64 constant CT values from Appendix B, Table 5.

c. If encryption is supported by the IUT, forward the KEY to the IUT using Input Type 4. If encryption is not supported by the IUT, forward the KEY and 64 CT values to the IUT using Input Type 3.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Variable Plaintext Known Answer test for the ECB Mode (Section 5.1.1.1). Otherwise, use the first value received from the MOVS.

b. Perform the following for i=1 through 64:

i. Set the input block IB_i equal to the value of CT_i, i.e., $(IB1_i, IB2_i, ..., IB64_i) = (CT1_i, CT2_i, ..., CT64_i)$.

ii. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in plaintext PT_i.

iii. Forward the current values of the loop number i, KEY, CT_i, and the resulting PT_i to the MOVS as specified in Output Type 1.

iv. Retain PT_i for use with the Initial Permutation Known Answer test for the ECB mode (Section 5.1.2.2).

v. If encryption is supported, set CT_{i+1} equal to the corresponding output from the Variable Plaintext Known Answer test for the ECB mode. If encryption is not supported, assign a new value to CT_{i+1} by setting it equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.1.2.2 The Initial Permutation Known Answer Test - ECB Mode

```
MOVS: Initialize KEY:   If DES, KEY=0101010101010101 (odd parity set)
                        If Skipjack, KEY=00000000000000000000

              CT_i (where i=1-64) = 64 PT values from Variable Ciphertext Known Answer test
              Send KEY, 64, CT_1, CT_2,...CT_64

IUT:    Initialize CT_1 = first value from output of Variable Ciphertext Known Answer test.

        FOR i = 1 to 64
          {
              IB_i = CT_i
              Perform algorithm in decrypt state, resulting in PT_i
              Send i, KEY, CT_i, PT_i
              CT_i+1 = the corresponding CT_i+1 value from MOVS
          }

MOVS: Compare results from each loop with known answers.  For DES, use Appendix B, Table 1.  For
      Skipjack, use Appendix B, Table 5.
```

Figure 5.8 ***The Initial Permutation Known Answer Test - ECB Mode***

As summarized in Figure 5.8, the Initial Permutation Known Answer test for the ECB Mode of Operation shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity. For IUTs of the Skipjack algorithm, KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 CT values with the 64 PT values obtained from the Variable Ciphertext Known Answer test.

   c. Forward the KEY and the 64 CT values to the IUT using Input Type 3.

2. The IUT shall perform the following for i=1 through 64:

   a. Set the input block IB_i equal to the value of CT_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i)$ =

($CT1_i$,$CT2_i$,...,$CT64_i$).

b. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in plaintext PT_i.

c. Forward the current values of the loop number i, KEY, CT_i, and the resulting PT_i to the MOVS as specified in Output Type 1.

d. Set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

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

```
MOVS: Initialize KEY_1:  If DES, KEY_1 = 8001010101010101 (odd parity)
                         If Skipjack, KEY_1 = 80000000000000000000

        If encryption is supported by the IUT:
                Send     KEY_1
        If encryption is not supported by the IUT:
                Initialize CT values:  If DES, initialize CT values with values in Appendix B, Table 2
                                       If Skipjack, initialize CT values with values in Appendix B,
                                     Table 6
                Send     KEY_1, n (where n=64 if DES, 80 if Skipjack), CT_1, CT_2,...,CT_n

IUT:    If encryption is supported by the IUT:
                Initialize CT_i = first value from output of Variable Key Known Answer test for the
                Encryption Process for the ECB Mode.
        Otherwise, use the first value received from the MOVS.

        FOR i = 1 to n, where n = 64 if DES, 80 if Skipjack
                {
                        IF (algorithm == SKIPJACK) {process every bit}
                                   OR
                          (algorithm == DES AND i %8 != 0)
                                {process every bit except parity bits}
                        {
                                IB_i = CT_i
                                Perform algorithm in decrypt state, resulting in PT_i
                                Send i, KEY_i, CT_i, PT_i
                                KEY_i+1 = vector consisting of "0" in every
                                        significant bit position except for a single "1" bit in position
                                        i+1. Note that odd parity is set.
                                If encryption is supported by the IUT:
                                        CT_i+1= corresponding CT_i+1 from output of Variable Key
                                        Known Answer test for the Encryption Process for the ECB
                                        Mode
                                else
                                        CT_i+1= corresponding CT_i+1 from MOVS
                        }
                }

MOVS: Compare results of the n decryptions with known answers
```

Figure 5.9 ***The Variable Key Known Answer Test for the Decryption Process - ECB Mode***

Figure 5.9 illustrates the Variable Key Known Answer test for the ECB Decryption Process.

1.The MOVS shall:

a. Initialize the KEY_1 to contain "0" in every significant bit except for a "1" in the first position. For example, if validating an IUT of the DES algorithm, the 64 bit $KEY_{1\ bin}$ = 1000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01. Note that the parity bits are set to "0" or "1" to set odd parity.

   If validating an IUT of the Skipjack algorithm, the 80 bit $KEY_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00 00 00.

b. If the IUT implements the DES algorithm and encryption is not supported, initialize CT_i values with the 56 constant CT values from Appendix B, Table 2. If the IUT implements the Skipjack algorithm, and encryption is not supported, initialize CT_i values with the 80 constant CT values from Appendix B, Table 6.

c. If encryption is not supported by the IUT, forward KEY and the CT values to the IUT using Input Type 3. Otherwise, forward the KEY to the IUT using Input Type 4.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Variable Key Known Answer test for the Encryption Process for the ECB Mode (Section 5.1.1.3). Otherwise, use the first value received from the MOVS.

b. Perform the following for i=1 to *n*, where *n* = 56 for DES or 80 for Skipjack:

   i. Set the input block IB_i equal to the value of CT_i, i.e., ($IB1_i$, $IB2_i$,..., $IB64_i$) = ($CT1_i$, $CT2_i$,...,$CT64_i$).

   ii. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in plaintext PT_i.

   iii. Forward the current values of the loop number i, KEY_i, CT_i, and the resulting PT_i to the MOVS as specified in Output Type 1.

iv. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in position i+1. The parity bits are set for odd parity.

v. If encryption is supported, set CT_{i+1} equal to the corresponding CT_{i+1} value retained from the Variable Key Known Answer test for the Encryption Process for ECB mode. If encryption is not supported by the IUT, set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The output from the IUT for this test shall consist of 56 output strings if DES is implemented or 80 output strings if Skipjack is implemented. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.1.2.4 Permutation Operation Known Answer Test for Decryption Process - ECB Mode

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize KEY_i (where i=1-32) = KEY values in Appendix B, Table 3

      If encryption is supported by the IUT:
            Send 32, KEY_1, KEY_2,..., KEY_32
      If encryption is not supported by the IUT:
            Initialize CT_i (where i=1-32) = corresponding CT values in Table 3
            Send 32, KEY_1, CT_1, KEY_2, CT_2,...,KEY_32, CT_32

IUT:  If encryption is supported by the IUT:
            Initialize CT_i = first value retained from Permutation Operation Known Answer test for
            the Encryption Process for the ECB Mode.
            Otherwise, use the first values received from the MOVS.

      FOR i = 1 to 32
       {
            IB_i = CT_i
            Perform DES algorithm in decrypt state using KEY_i, resulting in PT_i
            Send i, KEY_i, CT_i, PT_i
            KEY_i+1 = corresponding KEY supplied by MOVS
            If encryption is supported by the IUT:
                  CT_i+1= the corresponding CT_i+1 retained from Permutation Operation Known
                  Answer test for the Encryption Process for the ECB Mode
            else
                  CT_i+1= the corresponding CT_i+1 from MOVS
       }

MOVS: Compare results from each loop with known answers
```

Figure 5.10 ***The Permutation Operation Known Answer Test for the Decryption Process - ECB Mode***

As summarized in Figure 5.10, the Permutation Operation Known Answer test for the ECB Decryption Process shall be performed as follows:

1. The MOVS shall:

   a. If the IUT supports encryption, initialize the KEY values with the 32 constant KEY values supplied from Table 3. If the IUT does not support encryption, initialize the KEY-ciphertext (KEY-CT) pairs with the 32 constant KEY-CT pairs from Appendix B, Table 3.

b. If encryption is supported by the IUT, forward the 32 KEY values using Input Type 10. If encryption is not supported by the IUT, forward the 32 KEY and CT pairs to the IUT using Input Type 9.

2. The IUT shall:

a. If encryption is supported by the IUT, initialize the CT value with the first CT value retained from the Permutation Operation Known Answer test for the Encryption Process for the ECB Mode (Section 5.1.1.4). Otherwise, use the first value received from the MOVS.

b. Perform the following for i = 1 to 32:

i. Set the input block IB_i equal to the value of CT_i, i.e, $(IB1_i,IB2_i,...IB64_i)=(CT1_i,CT2_i,..., CT64_i)$.

ii. Using the corresponding KEY_i, process IB_i through the DES algorithm in the decrypt state, resulting in plaintext PT_i.

iii. Forward the current values of the loop number i, KEY_i, CT_i, and the resulting PT_i to the MOVS as specified in Output Type 1.

iv. Assign a new value to KEY_{i+1} by setting it equal to the corresponding KEY value supplied by the MOVS.

v. If encryption is supported, set CT_{i+1} equal to the corresponding CT value retained from the Permutation Operation Known Answer test for the Encryption Process for ECB mode. If encryption is not supported, set CT_{i+1} equal to the corresponding CT value supplied by the MOVS.

NOTE: The above processing shall continue until all 32 KEY-CT values are passed as specified in Input Type 9 or all 32 KEY values are passed as specified in Input Type 10. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

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

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS:  Initialize KEY_i (where i=1-19) = KEY values in Appendix B, Table 4

       If encryption is supported by the IUT:
              Send 19, KEY_1, KEY_2,...,KEY_19
       If encryption is not supported by the IUT:
              Initialize CT_i (where i=1-19) = corresponding CT values in Table 4
              Send 19, KEY_1, CT_1, KEY_2, CT_2,...,KEY_19, CT_19

IUT:   If encryption is supported by the IUT:
              Initialize CT_1 = first value from output of Substitution Table Known Answer test for the
              Encryption Process for the ECB Mode.
              Otherwise, use the first value received from the MOVS.

       FOR i = 1 to 19
              {
                     IB_i = CT_i
                     Perform DES algorithm in decrypt state using KEY_i, resulting in PT_i
                     Send i, KEY_i, CT_i, PT_i
                     KEY_i+1 = corresponding KEY_i+1 supplied by MOVS
                     If encryption is supported
                            CT_i+1= corresponding CT_i+1 from output of Substitution Table Known
                            Answer test for the Encryption Process for the ECB Mode
                     else
                            CT_i+1= the corresponding CT_i+1 from MOVS
              }

MOVS:  Compare results from each loop with known answers
```

Figure 5.11 ***The Substitution Table Known Answer Test for the Decryption Process - ECB Mode***

Figure 5.11 illustrates the Substitution Table Known Answer test for the ECB Decryption Process.

1. The MOVS shall:

   a. If the IUT supports encryption, initialize the KEY values with the 19 constant KEY values supplied from Appendix B, Table 4. If the IUT does not support encryption, initialize the KEY-ciphertext (KEY-CT) pairs with the 19 constant KEY-CT pairs from Appendix B, Table 4.

b. If encryption is supported by the IUT, forward the 19 KEY values using Input Type 10. Forward the 19 KEY-CT pairs to the IUT using Input Type 9 if encryption is not supported by the IUT.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Substitution Table Known Answer test for the Encryption Process for the ECB Mode (Section 5.1.1.5). Otherwise, use the first value received from the MOVS.

b. Perform the following for i = 1 to 19:

i. Set the input block IB_i equal to the value of CT_i, i.e, $(IB1_i, IB2_i, ...IB64_i) = (CT1_i, CT2_i, ..., CT64_i)$.

ii. Using the corresponding KEY_i, process IB_i through the DES algorithm in the decrypt state, resulting in plaintext PT_i.

iii. Forward the current values of the loop number i, KEY_i, CT_i, and the resulting PT_i to the MOVS as specified in Output Type 1.

iv. Set KEY_{i+1} equal to the corresponding KEY supplied by MOVS.

v. If encryption is supported, set CT_{i+1} equal to the corresponding CT value retained from the Substitution Table Known Answer test for the Encryption Process for the ECB mode. If encryption is not supported, set CT_{i+1} equal to the corresponding CT value supplied by the MOVS.

NOTE: The above processing shall continue until all 19 KEY-CT pairs, as specified in Input Type 9, or all 19 KEY values, as specified in Input Type 10, are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.1.2.6 Modes Test for the Decryption Process - ECB Mode

```
MOVS: Initialize            KEY_0, CT_0
      Send                  KEY_0, CT_0

IUT:  FOR i = 0 TO 399
       {
              Record i, KEY_i, CT_0
              FOR j = 0 TO 9,999
               {
                     IB_j = CT_j
                     Perform algorithm in decrypt state, resulting in PT_j
                     CT_j+1 = PT_j
               }
              Record PT_j
              Send i, KEY_i, CT_0, PT_j
              KEY_i+1 = KEY_i ⊕ last n bits of PT,
                     where n=64 if DES and n=80 if Skipjack
              CT_0 = PT_9999
       }

MOVS:Check IUT's output for correctness
```

Figure 5.12 ***The Modes Test for the Decryption Process - ECB Mode***

Figure 5.12 illustrates the Modes test for the ECB Decryption Process.

1. The MOVS shall:

    a. Initialize KEY and ciphertext CT variables. The CT shall consist of 64 bits, while the KEY length shall be dependent on the algorithm implemented by the IUT.

    b. Forward these values to the IUT using Input Type 1.

2. The IUT shall perform the following for i=0 through 399:

    a. Record the current values of the outer loop number i, the KEY_i, and the CT_0.

    b. Perform the following for j=0 through 9999:

        i. Set the input block IB_j equal to the value of CT_j, i.e., ($IB1_j$, $IB2_j$, ...,

$IB64_j) = (CT1_j, CT2_j, \ldots, CT64_j)$.

ii. Process IB_j through the DES or Skipjack algorithm in the decrypt state, resulting in plaintext PT_j.

iii. Prepare for loop j+1 by assigning CT_{j+1} with the current value of PT_j, i.e., $(CT1_{j+1}, CT2_{j+1}, \ldots CT64_{j+1}) = (PT1_j, PT2_j, \ldots, PT64_j)$.

c. Record the PT_j .

d. Output all recorded information for this loop as specified in Output Type 1.

e. Assign a new value to the KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY with the current PT. For IUTs of the DES algorithm, this shall equate to $(KEY1_{i+1}, KEY2_{i+1}, \ldots KEY64_{i+1}) = ((KEY1_i \oplus PT1_{9999}, KEY2_i \oplus PT2_{9999}, \ldots KEY64_i \oplus PT64_{9999})$.

For IUTs for the Skipjack algorithm, the PT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished by concatenating the 16 rightmost bits of the previous PT (PT_{9998}) with the 64 bits of the current PT (PT_{9999}). This value shall then be exclusive-ORed with the current KEY to form the new KEY, i.e., $(KEY1_{i+1}, KEY2_{i+1}, \ldots KEY80_{i+1}) = (KEY1_i \oplus PT49_{9998}, KEY2_i \oplus PT50_{9998}, \ldots KEY16_i \oplus PT64_{9998}, KEY17_i \oplus PT1_{9999}, KEY18_i \oplus PT2_{9999}, \ldots KEY80_i \oplus PT64_{9999})$.

f. Assign a new value to CT in preparation for the next outer loop. CT_0 shall be assigned the value of the current PT, i.e., $(CT1_0, CT2_0, \ldots, CT64_0) = (PT1_{9999}, PT2_{9999}, \ldots, PT64_{9999})$. (Note that the new CT shall be denoted as CT_0 to be used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings consisting of information included in Output Type 1.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2 Cipher Block Chaining (CBC) Mode

The IUTs for the DES or Skipjack algorithm in the Cipher Block Chaining (CBC) mode shall be validated by successfully completing a series of Known Answer tests and Modes tests corresponding to the cryptographic processes allowed by the IUT.

5.2.1 Encryption Process

The process of validating an IUT for the DES algorithm which implements the encryption process of the CBC mode of operation shall involve the successful completion of the following six tests:

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

The validation process for an IUT of the Skipjack algorithm which implements the encryption process of the CBC mode of operation shall require the successful completion of tests 1, 2, 3, and 6 only.

An explanation of the tests follows.

5.2.1.1 The Variable Plaintext Known Answer Test - CBC Mode

```
MOVS: Initialize      KEY:  If DES, KEY = 0101010101010101 (odd parity set)
                            If Skipjack, KEY = 00000000000000000000
                      IV = 0000000000000000
                      PT1 = 8000000000000000
      Send            KEY, IV, PT1

IUT:  FOR i = 1 to 64
       {
              IBi= PTi⊕ IV
              Perform algorithm in encrypt state, resulting in CTi
              Send i, KEY, IV, PTi, CTi
              PTi+1 = basis vector where single "1" bit is in position i+1
       }

MOVS: Compare results from each loop with known answers
      If DES, use Appendix B, Table 1.  If Skipjack, use Appendix B, Table 5.
```

Figure 5.13 *The Variable Plaintext Known Answer Test - CBC Mode*

Figure 5.13 illustrates the Variable Plaintext Known Answer test for the CBC mode.

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit IV parameter to the constant hexadecimal value 0, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

   c. Initialize the 64 bit plaintext PT_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $PT_{1\,bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

   d. Forward this information to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 1 through 64:

   a. Calculate the input block IB_i by exclusive-ORing PT_i with IV, i.e., $(IB1_i,IB2_i,...IB64_i) = (PT1_i \oplus IV1,PT2_i \oplus IV2,..., PT64_i \oplus IV64)$.

   b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

   c. Forward the current values of the loop number i, KEY, IV, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 2.

   d. Retain CT_i for use with the Inverse Permutation Known Answer test for the CBC Mode of Operation (Section 5.2.1.2), and, if the IUT supports decryption, for use with the Variable Ciphertext Known Answer test for the CBC Mode (Section 5.2.2.1).

   e. Assign a new value to PT_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1, where i+1=2..64.

   NOTE: This continues until every possible basis vector has been represented by the PT, i.e. 64 times. The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 1 for DES or Table 5 for Skipjack.

5.2.1.2 The Inverse Permutation Known Answer Test - CBC Mode

```
MOVS: Initialize       KEY:   If DES, KEY = 0101010101010101 (odd parity set)
                              If Skipjack, KEY = 00000000000000000000
                       IV = 0000000000000000
                       PT_i (where i=1-64) = 64 CT values from the Variable Plaintext Known Answer
                       test
      Send             KEY, IV, 64, PT_1..PT_64

IUT:  FOR i = 1 to 64
       {
              IB_i = PT_i ⊕ IV
              Perform algorithm in encrypt state, resulting in CT_i
              Send i, KEY, IV, PT_i, CT_i
              PT_i+1 = corresponding PT_i+1 from MOVS
       }

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

Figure 5.14 ***The Inverse Permutation Known Answer Test - CBC Mode***

Figure 5.14 illustrates the Inverse Permutation Known Answer test for the CBC mode.

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit IV parameter to the constant hexadecimal value 0, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

   c. Initialize the 64 bit plaintext values PT_i (where i=1-64) to the CT_i results obtained from the Variable Plaintext Known Answer test.

   d. Forward this information to the IUT using Input Type 5.

2. The IUT shall perform the following for i = 1 through 64:

a. Calculate the input block IB_i by exclusive-ORing PT_i with IV, i.e., $(IB1_i, IB2_i, \ldots IB64_i) = (PT1_i \oplus IV1, PT2_i \oplus IV2, \ldots, PT64_i \oplus IV64)$.

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current values of the loop number i, KEY, IV, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 2.

d. Assign a new value to PT_{i+1} by setting it equal to the corresponding output from the Variable Plaintext Known Answer test for the CBC mode.

NOTE: This processing continues until all ciphertext values from the Variable Plaintext Known Answer test have been used as input. The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values. The CT values should be the set of basis vectors that were used as plaintext for the Variable Plaintext Known Answer test.

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

```
MOVS: Initialize      KEY_1:  If DES, KEY_1 = 8001010101010101 (with odd parity)
                              If Skipjack, KEY_1 = 80000000000000000000
                      IV = 0000000000000000
                      PT = 0000000000000000
      Send            KEY_1, IV, PT

IUT:  FOR i = 1 to n, where n = 64 if DES, 80 if Skipjack
              {
                      IF (algorithm == SKIPJACK) {process every bit}
                              OR
                      (algorithm == DES AND i %8 != 0)
                              {process every bit except parity bits}
                      {
                              IB_i = PT ⊕ IV
                              Perform algorithm in encrypt state using KEY_i, resulting in CT_i
                              Send i, KEY_i, IV, PT, CT_i
                              KEY_i+1 = vector consisting of "0" in every significant bit position
                              except for a single "1" bit in position i+1. Note that parity bits are
                              "0" or "1" to make the KEY odd parity.
                      }
              }

MOVS: Compare results of the n encryptions with known answers
      For DES, use Appendix B, Table 2. For Skipjack, use Appendix B, Table 6.
```

Figure 5.15 ***The Variable Key Known Answer Test for the Encryption Process - CBC Mode***

As summarized in Figure 5.15, the Variable Key Known Answer test for the CBC Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize KEY_1 to contain "0" in every significant bit except for a "1" in the first position. For example, if validating an IUT of the DES algorithm, the 64 bit $KEY_{1\ bin}$ = 10000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01. Note that the parity bits are set to "0" or "1" to get odd parity.

If validating an IUT for the Skipjack algorithm, the 80 bit $KEY_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00 00 00.

b. Initialize the 64 bit initialization vector IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

c. Initialize the 64 bit plaintext PT to the value of 0, i.e., PT_{hex}=00 00 00 00 00 00 00 00.

d. Forward this information to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 1 to *n*: (NOTE: *n* equals the number of significant bits in a DES or Skipjack key.)

   a. Calculate the input block IB_i by exclusive-ORing PT with the IV, i.e, $(IB1_i, IB2_i, ... IB64_i) = (PT1 \oplus IV1, PT2 \oplus IV2, ..., PT64 \oplus IV64)$.

   b. Using the corresponding KEY_i, process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in ciphertext CT_i.

   c. Forward the current value of the loop number i, KEY_i, IV, PT, and the resulting CT_i to the MOVS as specified in Output Type 2.

   d. If the IUT supports decryption, retain CT_i for use with the Variable Key Known Answer test for the Decryption Process for the CBC Mode (Section 5.2.2.3).

   e. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in position i+1. The parity bits are set for odd parity.

   NOTE: The above processing continues until every significant basis vector has been represented by the KEY parameter. The output from the IUT for this test shall consist of 56 output strings if DES is implemented and 80 output strings if Skipjack is implemented. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 2 for DES or Table 6 for Skipjack.

5.2.1.4 Permutation Operation Known Answer Test for the Encryption Process - CBC Mode

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize     KEY_i (where i=1-32)  = 32 KEY values in Appendix B, Table 3
                     IV = 0000000000000000
                     PT = 0000000000000000
      Send           PT, IV, KEY_1, KEY_2, ... KEY_32

IUT:  FOR i = 1 to 32
       {
              IB_i = PT ⊕ IV
              Perform DES algorithm in encrypt state using KEY_i, resulting in CT_i
              Send i, KEY_i, IV, PT, CT_i
              KEY_i+1 = KEY_i+1 from MOVS
       }

MOVS: Compare results with known answers
```

Figure 5.16 ***The Permutation Operation Known Answer Test for the Encryption Process - CBC Mode***

Figure 5.16 illustrates the Permutation Operation Known Answer test for the CBC Encryption Process.

1. The MOVS shall:

   a. Initialize KEY_i, where i=1-32, with the 32 constant KEY values from Appendix B, Table 3.

   b. Initialize the 64 bit IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

   c. Initialize the plaintext PT to the value of 0, i.e., PT_{hex}=00 00 00 00 00 00 00 00.

   d. Forward this information to the IUT using Input Type 8.

2. The IUT shall perform the following for i = 1 to 32:

a. Calculate the input block IB_i by exclusive-ORing PT with IV, i.e, $(IB1_i, IB2_i, \ldots IB64_i) = (PT1 \oplus IV1, PT2 \oplus IV2, \ldots, PT64 \oplus IV64)$.

b. Using the corresponding KEY_i, process IB_i through the DES algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current value of the loop number i, KEY_i, IV, PT, and the resulting CT_i to the MOVS as specified in Output Type 2.

d. If the IUT supports decryption, retain CT_i for use with the Permutation Operation Known Answer test for the Decryption Process for the CBC mode (Section 5.2.2.4).

e. Set KEY_{i+1} equal to the corresponding KEY supplied by the MOVS.

NOTE: The above processing shall continue until all 32 KEY values as specified in Input Type 8 are processed. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 3.

5.2.1.5 Substitution Table Known Answer Test for the Encryption Process - CBC Mode

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize      KEY_i (where i=1-19) = 19 KEY values in Appendix B, Table 4
                      PT_i = (where i=1-19) = 19 corresponding PT values in Table 4
                      IV = 0000000000000000
      Send            IV, 19, KEY_1, PT_1, KEY_2, PT_2,...,KEY_19, PT_19

IUT:  FOR i = 1 to 19
       {
              IB_i = PT_i ⊕ IV
              Perform DES algorithm in encrypt state using KEY_i, resulting in CT_i
              Send i, KEY_i, IV, PT_i, CT_i
              KEY_i+1 = KEY_i+1 from MOVS
              PT_i+1 = corresponding PT_i+1 from MOVS
       }

MOVS: Compare results from each loop with known answers
```

Figure 5.17 ***The Substitution Table Known Answer Test for the Encryption Process - CBC Mode***

As summarized in Figure 5.17, the Substitution Table Known Answer test for the CBC Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY-plaintext (KEY-PT) pairs with the 19 constant KEY-PT values from Appendix B, Table 4.

   b. Initialize IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

   c. Forward this information to the IUT using Input Type 11.

2. The IUT shall perform the following for i = 1 to 19:

   a. Calculate the input block IB_i by exclusive-ORing PT_i with the IV, i.e,

$(IB1_i,IB2_i,...IB64_i) = (PT1_i \oplus IV1,PT2_i \oplus IV2,...,PT64_i \oplus IV64)$.

b. Using the corresponding KEY_i, process IB_i through the DES algorithm in the encrypt state, resulting in ciphertext CT_i.

c. Forward the current value of the loop number i, KEY_i, IV, PT_i, and the resulting CT_i to the MOVS as specified in Output Type 2.

d. If the IUT supports decryption, retain CT_i for use with the Substitution Table Known Answer test for the CBC Decryption Process (Section 5.2.2.5).

e. Set KEY_{i+1} equal to the corresponding KEY value supplied by MOVS.

f. Set PT_{i+1} equal to the corresponding PT value supplied by MOVS.

NOTE: The above processing continues until all 19 KEY-PT pairs, as specified in Input Type 11, are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 4.

5.2.1.6 Modes Test for the Encryption Process - CBC Mode

```
MOVS: Initialize KEY0, IV, PT0
      Send      KEY0, IV, PT0

IUT:  FOR i= 0 TO 399
      {
            If (i==0) CV0 = IV
            Record i, KEYi, CV0, PT0
            FOR j = 0 TO 9,999
             {
                  IBj = PTj ⊕ CVj
                  Perform algorithm in encrypt state, resulting in CTj
                  IF j=0
                    PTj+1 = CV0
                  ELSE
                    PTj+1 = CTj-1
                  CVj+1 = CTj
             }
            Record CTj
            Send i, KEYi, CV0, PT0, CTj
            KEYi+1 = KEYi ⊕ last n bits of CT, where n=64 if DES, n=80 if Skipjack
            PT0 = CT9998
            CV0 = CT9999
      }

MOVS: Check IUT's output for correctness
```

Figure 5.18 ***The Modes Test for the Encryption Process - CBC Mode***

As summarized in Figure 5.18, the Modes test for the CBC Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY, initialization vector IV and plaintext PT variables. The PT and IV shall consist of 64 bits each. The KEY length shall be dependent on the algorithm implemented by the IUT.

   b. Forward these values to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 0 through 399:

   a. If i=0 (if this is the first time through this loop), set the chaining value CV_0 equal to the IV.

b. Record the current value of the outer loop number i, KEY_i, CV_0 and PT_0.

c. For j = 0 through 9999, perform the following:

 i. Set the input block IB_j equal to the value of PT_j exclusive-ORed with the CV_j, i.e., $(IB1_j, IB2_j, ..., IB64_j) = (PT1_j \oplus CV1_j, PT2_j \oplus CV2_j, ..., PT64_j \oplus CV64_j)$.

 ii. Process IB_j through the DES or Skipjack algorithm in the encrypt state, resulting in CT_j.

 iii. Prepare for loop j+1 by doing the following:

  - Assign CV_{j+1} with the current value of CT_j, i.e., $(CV1_{j+1}, CV2_{j+1}, ..., CV64_{j+1}) = (CT1_j, CT2_j, ..., CT64_j)$.

  - If the inner loop being processed is the first loop, i.e., j = 0, assign PT_{j+1} with the current value of CV_0, i.e., $(PT1_1, PT2_1, ..., PT64_1) = (CV1_0, CV2_0, ..., CV64_0)$. Otherwise, assign PT_{j+1} with the CT from the previous inner cycle, CT_{j-1}, i.e., $(PT1_{j+1}, PT2_{j+1}, ..., PT64_{j+1}) = (CT1_{j-1}, CT2_{j-1}, ... CT64_{j-1})$.

d. Record the CT_j .

e. Output all recorded information from this loop, as specified in Output Type 2, to the MOVS.

f. Assign a new value to the KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY with the current CT. For IUTs of the DES algorithm, this shall equate to $(KEY1_{i+1}, KEY2_{i+1}, ... KEY64_{i+1}) = (KEY1_i \oplus CT1_{9999}, KEY2_i \oplus CT2_{9999}, ... KEY64_i \oplus CT64_{9999})$.

 For IUTs of the Skipjack algorithm, CT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished by concatenating the 16 rightmost bits of the previous CT (CT_{9998}) with the 64 bits of the current CT (CT_{9999}). This value shall then be exclusive-ORed with the current KEY to form the new KEY, i.e., $(KEY1_{i+1}, KEY2_{i+1}, ... KEY80_{i+1}) = (KEY1_i \oplus CT49_{9998}, KEY2_i \oplus CT50_{9998}, ... KEY16_i \oplus CT64_{9998}, KEY17_i \oplus CT1_{9999}, KEY18_i \oplus CT2_{9999}, ... KEY80_i \oplus CT64_{9999})$.

g. Assign a new value to CV_0 in preparation for the next outer loop. CV_0 shall be assigned the value of the current CT, i.e., ($CV1_0$, $CV2_0$, ..., $CV64_0$) = ($CT1_{9999}$, $CT2_{9999}$, ..., $CT64_{9999}$).(Note that the new CV shall be denoted as CV_0 because this value is used for the first pass through the inner loop when j=0.)

h. Assign a new value to the PT in preparation of the next outer loop. PT_0 shall be assigned the value of the CT from the previous cycle, i.e., ($PT1_0$, $PT2_0$,...,$PT64_0$) = ($CT1_{9998}$, $CT2_{9998}$,...,$CT64_{9998}$). (Note that the new PT shall be denoted as PT_0 because this value is used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2 Decryption Process

The process of validating an IUT for the CBC mode of the DES algorithm which implements the decryption process shall involve the successful completion of the following six tests:

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

The validation process for an IUT of the Skipjack algorithm using the CBC mode of operation in the decryption process shall require the successful completion of tests 1, 2, 3, and 6 only.

An explanation of the tests follows.

5.2.2.1 The Variable Ciphertext Known Answer Test - CBC Mode

MOVS: If encryption is supported by the IUT:
Initialize KEY: If DES, KEY = 0101010101010101 (odd parity set)
If Skipjack, KEY=00000000000000000000
IV = 0000000000000000
Send KEY, IV

If encryption is not supported by the IUT:
Initialize KEY: If DES, KEY=0101010101010101 (odd parity set)
If Skipjack, KEY=00000000000000000000
IV = 0000000000000000
CT_i (where i=1-64): If DES, CT values in Appendix B, Table 1
If Skipjack, CT values in Appendix B, Table 5
Send KEY, IV, 64, CT_1, CT_2,...,CT_{64}

IUT: If encryption is supported:
Initialize CT_1= first value from output of Variable Plaintext Known Answer test.
Otherwise, use the first value received from the MOVS.

```
FOR i = 1 to 64
 {
      IB_i = CT_i
      Perform algorithm in decrypt state, resulting in OB_i
      PT_i = OB_i ⊕ IV
      Send i, KEY, IV, CT_i, PT_i
      If encryption is supported:
            CT_i+1 = corresponding CT_i+1 from output of Variable Plaintext Known Answer
            test
      else
            CT_i+1 = corresponding CT_i+1 value from MOVS
 }
```

MOVS: Compare results from each loop with known answers

Figure 5.19 ***The Variable Ciphertext Known Answer Test - CBC Mode***

As summarized in Figure 5.19, the Variable Ciphertext Known Answer test for the CBC mode of operation shall be performed as follows:

1. The MOVS shall:

a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity. For Skipjack implementations, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

b. Initialize the initialization vector IV to the constant hexadecimal value 0, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

c. If the IUT is of the DES algorithm, and it does not support encryption, initialize the 64 ciphertext CT values with the 64 constant CT values from Appendix B, Table 1. If the IUT is of the Skipjack algorithm, and it does not support encryption, initialize the 64 ciphertext CT values with the 64 constant values from Appendix B, Table 5.

d. If encryption is supported by the IUT, forward the KEY and IV to the IUT, as specified in Input Type 6. If encryption is not supported by the IUT, forward the KEY, IV and CT to the IUT, as specified in Input Type 5.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Variable Plaintext Known Answer test for the CBC Mode (Section 5.2.1.1). Otherwise, use the first value received from the MOVS.

b. Perform the following for i=1 through 64:

i. Set the input block IB_i equal to the value of CT_i, i.e., $(IB1_i,IB2_i,...,IB64_i) = (CT1_i,CT2_i,...,CT64_i)$.

ii. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in the output block OB_i.

iii. Calculate the plaintext PT_i by exclusive-ORing OB_i with IV, i.e., $(PT1_i, PT2_i,...,PT64_i) = (OB1_i \oplus IV1, OB2_i \oplus IV2,...,OB64_i \oplus IV64)$.

iv. Forward the current value of the loop number i, KEY, IV, CT_i, and the resulting PT_i to the MOVS using Output Type 2.

v. If encryption is supported, set CT_{i+1} equal to the corresponding output from the Variable Plaintext Known Answer test for CBC mode. If encryption is not supported, assign a new value to CT_{i+1} by setting it equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2.2 The Initial Permutation Known Answer Test - CBC Mode

```
MOVS: Initialize        KEY:   If DES, KEY = 0101010101010101 (odd parity set)
                               If Skipjack, KEY=00000000000000000000
                               IV = 0000000000000000
                               CT_i (where i=1-64): 64 PT values from Variable Ciphertext Known
                               Answer test
                  Send  KEY, IV, 64, CT_1, CT_2,...,CT_64

IUT:   Initialize CT_1= first value from output of Variable Ciphertext Known Answer test.

       FOR i = 1 to 64
        {
              IB_i = CT_i
              Perform algorithm in decrypt state, resulting in OB_i
              PT_i = OB_i ⊕ IV
              Send i, KEY, IV, CT_i, PT_i
              CT_i+1 = corresponding CT_i+1 value from MOVS
        }

MOVS: Compare results from each loop with known answers. For DES, use Appendix B, Table 1, For
      Skipjack, use Appendix B, Table 5.
```

Figure 5.20 ***The Initial Permutation Known Answer Test - CBC Mode***

As summarized in Figure 5.20, the Initial Permutation Known Answer test for the CBC mode of operation shall be performed as follows:

1. The MOVS shall:

    a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity. For Skipjack implementations, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

    b. Initialize the initialization vector IV to the constant hexadecimal value 0, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

    c. Initialize the 64 CT values with the 64 PT values obtained from the Variable Ciphertext Known Answer test.

d. Forward the KEY, IV and the 64 CT values to the IUT, as specified in Input Type 5.

2. The IUT shall perform the following for i=1 through 64:

   a. Set the input block IB_i equal to the value of CT_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i) = (CT1_i, CT2_i, \ldots, CT64_i)$.

   b. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in the output block OB_i.

   c. Calculate the plaintext PT_i by exclusive-ORing OB_i with IV, i.e., $(PT1_i, PT2_i, \ldots, PT64_i) = (OB1_i \oplus IV1, OB2_i \oplus IV2, \ldots, OB64_i \oplus IV64)$.

   d. Forward the current value of the loop number i, KEY, IV, CT_i, and the resulting PT_i to the MOVS using Output Type 2.

   e. Set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

   NOTE: The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2.3 The Variable Key Known Answer Test for the Decryption Process - CBC Mode

```
MOVS: Initialize KEY:  If DES, KEY_1 = 8001010101010101 (odd parity set)
                       If Skipjack, KEY_1 = 80000000000000000000
                  IV=0000000000000000
      If encryption is supported by the IUT:
             Send                KEY_1, IV
      If encryption is not supported by the IUT:
             Initialize CT values:  If DES, initialize CT values with values in Appendix B, Table 2
                                    If Skipjack, initialize CT values with values in Appendix B,
                                    Table 6.
             Send                KEY_1, IV, n (where n=64 if DES, 80 if Skipjack), CT_1, CT_2,..., CT_n

IUT:  If encryption is supported by the IUT:
             Initialize CT_1 = first value from output of Variable Key Known Answer test for the
             Encryption Process for the CBC Mode.
      Otherwise, use the first value received from the MOVS.

      FOR i = 1 to n, where n = 56 if DES, 80 if Skipjack
       {
             IF (algorithm == SKIPJACK) {process every bit}
                            OR
               (algorithm == DES AND i %8 != 0)
                        {process every bit except parity bits}
              {
                        IB_i = CT_i
                        Perform algorithm in decrypt state, resulting in OB_i
                        PT_i = OB_i ⊕ IV
                        Send i, KEY_i, IV, CT_i, PT_i
                        KEY_i+1 = vector consisting of "0" in every significant bit position except
                                  for a single "1" bit in the i+1st position.  Note that odd parity is set.
                        If encryption is supported by the IUT:
                                  CT_i+1 = corresponding CT_i+1 from output of Variable Key Known
                                  Answer test for the Encryption Process for CBC Mode
                        else
                                  CT_i+1 = corresponding CT_i+1 value from MOVS
              }
       }
MOVS: Compare results of the n decryptions with known answers
```

Figure 5.21 ***The Variable Key Known Answer Test for the Decryption Process - CBC Mode***

Figure 5.21 illustrates the Variable Key Known Answer test for the CBC Decryption Process.

1. The MOVS shall:

a. Initialize KEY_1 to contain "0" in every significant bit except for a "1" in the first position. (Note that odd parity is set on the KEY.) For example, if validating an IUT of the DES algorithm, the 64 bit $KEY_{1\ bin}$ = 10000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01.

   If validating an IUT of the Skipjack algorithm, the 80 bit $KEY_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00 00 00.

b. Initialize IV to contain the value of zero, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

c. If the IUT is of the DES algorithm, and encryption is not supported, initialize CT_i values with the 56 constant CT values from Appendix B, Table 2. Otherwise, if the IUT is of the Skipjack algorithm, and encryption is not supported, initialize the CT_i values with the 80 constant CT values from Appendix B, Table 6.

d. If encryption is not supported by the IUT, forward the KEY, IV, and the multiple CT values to the IUT, as specified in Input Type 5. Otherwise, forward the KEY and IV to the IUT, as specified in Input Type 6.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Variable Key Known Answer test for the Encryption Process for the CBC Mode (Section 5.2.1.3). Otherwise, use the first value received from the MOVS.

b. Perform the following for i=1 to *n*, where *n* = 56 for DES or 80 for Skipjack:

   i. Set the input block IB_i equal to the value of CT_i, i.e., $(IB1_i, IB2_i,..., IB64_i) = (CT1_i, CT2_i,...,CT64_i)$.

   ii. Process IB_i through the DES or Skipjack algorithm in the decrypt state, resulting in output block OB_i.

   iii. Calculate the plaintext PT_i by exclusive-ORing OB_i with IV, i.e., $(PT1_i, PT2_i,...,PT64_i) = (OB1_i \oplus IV1, OB2_i \oplus IV2,...,OB64_i \oplus IV64)$.

   iv. Forward the current values of the loop number i, KEY_i, IV, CT_i and the resulting PT_i to the MOVS using Output Type 2.

v. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in the $i+1^{st}$ position. The parity bits are set for odd parity.

vi. If encryption is supported, set CT_{i+1} equal to the corresponding CT_{i+1} value retained from the Variable Key Known Answer test for the Encryption Process for CBC mode. If encryption is not supported by the IUT, set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The output from the IUT for this test shall consist of 56 output strings if DES is being implemented, or 80 output strings if Skipjack is implemented. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2.4 Permutation Operation Known Answer Test for Decryption Process - CBC Mode

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize        KEY_i (where i=1-32) = KEY values in Appendix B, Table 3
                        IV = 0000000000000000
      If encryption is supported by the IUT:
              Send IV,32, KEY_1, KEY_2,...,KEY_32
      If encryption not supported by the IUT:
              Initialize CT_i (where i=1-32) = corresponding CT values in Table 3
              Send IV,32, KEY_1, CT_1, KEY_2, CT_2,...,KEY_32, CT_32

IUT:  If encryption is supported by the IUT:
              Initialize CT_1 = first value retained from Permutation Operation Known Answer test for the
              Encryption Process for the CBC Mode.
      Otherwise, use the first value received from the MOVS.

      FOR i = 1 to 32
       {
              IB_i = CT_i
              Perform DES algorithm in decrypt state using KEY_i, resulting in OB_i
              PT_i = OB_i ⊕ IV
              Send i, KEY_i, IV, CT_i, PT_i
              KEY_i+1 = corresponding KEY supplied by MOVS
              If encryption is supported:
                      CT_i+1 = corresponding CT_i+1 from output of Permutation Operation Known
                      Answer test for the Encryption Process for the CBC mode
              else
                      CT_i+1 = corresponding CT_i+1 from MOVS
      }

MOVS: Compare results from each loop with known answers
```

Figure 5.22 ***The Permutation Operation Known Answer Test for the Decryption Process - CBC Mode***

As summarized in Figure 5.22, the Permutation Operation Known Answer test for the CBC Decryption Process shall be performed as follows:

1. The MOVS shall:

    a. If the IUT supports encryption, initialize the KEY values with the 32 constant KEY values supplied from Appendix B, Table 3. If the IUT does not support encryption, initialize the KEY-ciphertext (KEY-CT) pairs with the 32 constant

KEY-CT pairs from Table 3.

b. Initialize IV to contain the value of zero, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

c. If encryption is supported by the IUT, forward the KEY and IV, as specified in Input Type 12. Forward the KEY, CT, and IV to the IUT using Input Type 11 if encryption is not supported by the IUT.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Permutation Operation Known Answer test for the Encryption Process for the CBC Mode (Section 5.2.1.4). Otherwise, use the first value received from the MOVS.

b. Perform the following for i = 1 to 32:

i. Set the input block IB_i equal to the value of CT_i, i.e, $(IB1_i,IB2_i,...IB64_i) = (CT1_i,CT2_i,...,CT64_i)$.

ii. Using the corresponding KEY_i, process IB_i through the DES algorithm in the decrypt state, resulting in OB_i.

iii. Calculate PT_i by exclusive-ORing OB_i with IV, i.e., $(PT1_i, PT2_i,...,PT64_i) = (OB1_i \oplus IV1, OB2_i \oplus IV2,...,OB64_i \oplus IV64)$.

iv. Forward the current values of the loop number i, KEY_i, IV, CT_i and the resulting PT_i to the MOVS using Output Type 2.

v. Set KEY_{i+1} equal to the i+1st value supplied by the MOVS.

vi. If encryption is supported, set CT_{i+1} equal to the corresponding CT_{i+1} value retained from the Permutation Operation Known Answer test for the Encryption Process for CBC Mode. If encryption is not supported, set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The above processing shall continue until all 32 KEY-CT values, as specified in Input Type 11, or all 32 KEY values, as specified in Input Type 12 are processed. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information contained in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2.5 Substitution Table Known Answer Test for the Decryption Process - CBC Mode

NOTE: This test shall only be performed for IUTs of the DES algorithm.

```
MOVS: Initialize:        KEY_i (where i=1-19)= KEY values in Appendix B, Table 4
                         IV = 0000000000000000
      If encryption is supported by the IUT:
                Send IV, 19, KEY_1, KEY_2,...,KEY_19
      If encryption not supported:
                Initialize CT_i (where i=1-19)= CT values in Table 4
                Send IV, 19, KEY_1, CT_1, KEY_2, CT_2,...,KEY_19, CT_19

IUT:  If encryption is supported:
                Initialize CT_1 = first CT value from output of Substitution Table Known Answer test for
                the Encryption Process for the CBC Mode.
      Otherwise, use the first value received from the MOVS.
      FOR i = 1 to 19
       {
                IB_i = CT_i
                Perform DES algorithm in decrypt state using KEY_i, resulting in OB_i
                PT_i=OB_i ⊕ IV
                Send i, KEY_i, IV, CT_i, PT_i
                KEY_i+1 = corresponding KEY supplied by MOVS
                If encryption is supported:
                          CT_i+1 = corresponding CT from output of Substitution Table Known Answer test
                          for the Encryption Process for the CBC mode
                else
                          CT_i+1 = corresponding CT from MOVS
       }

MOVS: Compare results from each loop with known answers
```

Figure 5.23 ***The Substitution Table Known Answer Test for the Decryption Process - CBC Mode***

Figure 5.23 illustrates the Substitution Table Known Answer test for the CBC Decryption Process.

1. The MOVS shall:

   a. If the IUT supports encryption, initialize the KEY values with the 19 constant KEY values supplied from Appendix B, Table 4. If the IUT does not support encryption, initialize the KEY-ciphertext (KEY-CT) pairs with 19 constant KEY-CT pairs from Appendix B, Table 4.

   b. Initialize IV to contain the value of zero, i.e., IV_{hex} = 00 00 00 00 00 00 00 00.

c. If encryption is supported by the IUT, forward the IV and the 19 KEY values, as specified in Input Type 12. Otherwise, forward the IV and the 19 KEY-CT pairs to the IUT, as specified in Input Type 11.

2. The IUT shall:

a. If encryption is supported, initialize the CT value with the first CT value retained from the Substitution Table Known Answer test for the Encryption Process for the CBC Mode (Section 5.2.1.5). Otherwise, use the first CT value received from the MOVS.

b. Perform the following for i = 1 to 19:

i. Set the input block IB_i equal to the value of CT_i, i.e, $(IB1_i, IB2_i, ... IB64_i) = (CT1_i, CT2_i, ..., CT64_i)$.

ii. Using the corresponding KEY_i, process IB_i through the DES algorithm in the decrypt state, resulting in the output block OB_i.

iii. Calculate PT_i by exclusive-ORing OB_i with IV, i.e., $(PT1_i, PT2_i, ..., PT64_i) = (OB1_i \oplus IV1, OB2_i \oplus IV2, ..., OB64_i \oplus IV64)$.

iv. Forward the current values of the loop number i, KEY_i, IV, CT_i and the resulting PT_i to the MOVS as specified in Output Type 2.

v. Set KEY_{i+1} equal to $i+1^{st}$ value supplied by MOVS.

vi. If encryption is supported, set CT_{i+1} equal to the corresponding CT_{i+1} value retained from the Substitution Table Known Answer test for the Encryption Process for the CBC Mode. If encryption is not supported, set CT_{i+1} equal to the corresponding CT_{i+1} value supplied by the MOVS.

NOTE: The above processing shall continue until the IV and all 19 KEY-CT pairs, as specified in Input Type 11, or the IV and all 19 KEY values, as specified in Input Type 12, are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.2.2.6 Modes Test for the Decryption Process - CBC Mode

```
MOVS: Initialize        KEY0, IV0, CT0
      Send              KEY0, IV0, CT0

IUT:  FOR i = 0 TO 399
      {
           If (i==0) CV0 = IV0
           Record i, KEYi, CV0, CT0
           FOR j = 0 TO 9,999
           {
                IBj = CTj
                Perform algorithm in decrypt state, resulting in OBj
                PTj = OBj ⊕ CVj
                CVj+1 = CTj
                CTj+1 = PTj
           }
           Record PTj
           Send i, KEYi, CV0, CT0, PTj
           KEYi+1 = KEYi ⊕ last n bits of PT, where n=64 if DES, n=80 if Skipjack
           CV0 = CT9999
           CT0 = PT9999
      }

MOVS: Check IUT's output for correctness
```

Figure 5.24 ***The Modes Test for the Decryption Process - CBC Mode***

Figure 5.24 illustrates the Modes test for the CBC Decryption Process.

1. The MOVS shall:

    a. Initialize KEY, the initialization vector IV and ciphertext CT variables. The CT and IV shall consist of 64 bits, while the KEY length shall be dependent on the algorithm implemented by the IUT.

    b. Forward these values to the IUT using Input Type 2.

2. The IUT shall perform the following for i=0 through 399:

   a. If i=0 (if this is the first time through this loop), set the chaining value CV_0 equal to IV.

   b. Record the current value of the outer loop number i, KEY_i, CV_0, and CT_0.

   c. For j=0 through 9999, perform the following:

      i. Set the input block IB_j equal to the value of CT_j, i.e., $(IB1_j, IB2_j, ..., IB64_j) = (CT1_j, CT2_j, ..., CT64_j)$.

      ii. Process the IB_j through the DES or Skipjack algorithm in the decrypt state, resulting in an output block OB_j.

      iii. Form the plaintext PT_j by exclusive-ORing OB_j with the current CV_j, i.e., $(PT1_j, PT2_j, ..., PT64_j) = (OB1_j \oplus CV1_j, OB2_j \oplus CV2_j, ..., OB64_j \oplus CV64_j)$.

      iv. Prepare for the j+1 loop by:

      - Assigning CV_{j+1} with the value of the current CT_j, i.e., $(CV1_{j+1}, CV2_{j+1}, ..., CV64_{j+1}) = (CT1_j, CT2_j, ..., CT64_j)$;

      - Assigning CT_{j+1} with the value of the current PT_j, i.e., $(CT1_{j+1}, CT2_{j+1}, ..., CT64_{j+1}) = (PT1_j, PT2_j, ..., PT64_j)$.

   d. Record PT_j.

   e. Output all the recorded information from this loop using Output Type 2.

   f. Assign a new value to the KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY with the current PT. For IUTs of the DES algorithm, this shall equate to $(KEY1_{i+1}, KEY2_{i+1}, ... KEY64_{i+1}) = (KEY1_i \oplus PT1_{9999}, KEY2_i \oplus PT2_{9999}, ... KEY64_i \oplus PT64_{9999})$.
   For IUTs of the Skipjack algorithm, the PT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished by concatenating the 16 rightmost bits of the previous PT (PT_{9998}) with the 64 bits of the current PT (PT_{9999}). This value shall then be exclusive-ORed with the current KEY to form the new KEY, i.e., $(KEY1_{i+1}, KEY2_{i+1}, ... KEY80_{i+1}) = (KEY1_i \oplus PT49_{9998}, KEY2_i \oplus PT50_{9998}, ... KEY16_i \oplus PT64_{9998}, KEY17_i \oplus PT1_{9999}, KEY18_i \oplus PT2_{9999}, ... KEY80_i \oplus PT64_{9999})$.

   g. Assign a new value to CV in preparation for the next outer loop. CV_0 shall be assigned the value of the current CT, i.e., $(CV1_0, CV2_0,\ldots,CV64_0) = (CT1_{9999}, CT2_{9999},\ldots,CT64_{9999})$. (Note that the new CV shall be denoted as CV_0 to be used for the first pass through the inner loop when j=0.)

h. Assign a new value to CT in preparation for the next outer loop. CT_0 shall be assigned the value of the current PT, i.e., $(CT1_0, CT2_0,\ldots,CT64_0) = (PT1_{9999}, PT2_{9999},\ldots,PT64_{9999})$. (Note that the new CT shall be denoted as CT_0 to be used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings consisting of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.3 The Cipher Feedback (CFB) Mode

The IUTs of the DES or Skipjack algorithm in the Cipher Feedback (CFB) mode of operation shall be validated by successfully completing (1) a set of Known Answer tests applicable to both IUTs supporting encryption and/or decryption and (2) a Modes test for each cryptographic process supported by the IUT.

The process of validating an IUT of the DES algorithm which supports the encryption and/or decryption processes of the K-bit CFB mode shall involve the successful completion of the following six tests:

1. The Variable Text Known Answer Test - K-bit CFB mode
2. The Inverse Permutation Known Answer Test - K-bit CFB mode
3. The Variable Key Known Answer Test - K-bit CFB mode
4. The Permutation Operation Known Answer Test - K-bit CFB mode
5. The Substitution Table Known Answer Test - K-bit CFB mode
6. The Modes Test for the Encryption Process - K-bit CFB mode (if encryption is supported)

   OR

   The Modes Test for the Decryption Process - K-bit CFB mode (if decryption is supported)

Note, for IUTs of the DES algorithm, K can range from 1 to 64 bits.

The validation process for an IUT of the Skipjack algorithm which supports the encryption and/or decryption process of the 64-bit CFB mode of operation shall involve the successful completion of tests 1, 2, 3, and 6 only.

An explanation of the tests follows.

5.3.1 The Known Answer Tests - CFB Mode

The K-bit CFB mode shall only have one set of Known Answer tests which shall be used regardless of supported process, i.e., the same set of Known Answer tests shall be used for IUTs supporting the encryption and/or decryption processes.

Throughout this section, TEXT and RESULT will refer to different variables depending on whether the encryption or decryption process is being tested. If the IUT performs CFB encryption, TEXT refers to plaintext, and RESULT refers to ciphertext. If the IUT performs CFB decryption, TEXT refers to ciphertext, and RESULT refers to plaintext.

5.3.1.1 The Variable Text Known Answer Test - CFB Mode

NOTE: If Skipjack, K shall equal 64.

MOVS: Initialize KEY: If DES, KEY = 0101010101010101 (odd parity set)
If Skipjack, KEY = 00000000000000000000
IV_1 = 8000000000000000
K-bit TEXT = 0
Send KEY, IV_1, K-bit TEXT

IUT: FOR i = 1 to 64
{
$IB_i = IV_i$
Perform algorithm in encrypt state, resulting in OB_i
K-bit $RESULT_i = LM^K(OB_i) \oplus$ K-bit TEXT
Send i, KEY, IV_i, K-bit TEXT, K-bit $RESULT_i$
IV_{i+1} = basis vector where single "1" bit is in position i+1
}

MOVS: Compare RESULT from each loop with known answers
If DES, use K bits of output in Appendix B, Table 1. If Skipjack, use 64 bits of output in Appendix B, Table 5.

Figure 5.25 ***The Variable Text Known Answer Test - CFB Mode***

As summarized in Figure 5.25, the Variable Text Known Answer test for the CFB mode shall be performed as follows (Note, in the following text, if the IUT is of the Skipjack algorithm, K shall equal 64.):

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit initialization vector IV_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $IV_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

c. Initialize the K-bit TEXT parameter to the constant hexadecimal value 0, where K = 1 ... 64 for DES and K = 64 for Skipjack.

d. Forward this information to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 1 through 64:

   a. Assign the value of the initialization vector IV_i to the input block IB_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i) = (IV1_i, IV2_i, \ldots, IV64_i)$.

   b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in a 64-bit output block OB_i.

   c. Calculate the K-bit $RESULT_i$ by exclusive-ORing the leftmost K-bits of OB_i with the K-bit TEXT, i.e., $(RESULT1_i, RESULT2_i, \ldots, RESULTK_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, \ldots, OBK_i \oplus TEXTK)$.

   d. Forward the current values of the loop number i, KEY, IV_i, K-bit TEXT and K-bit $RESULT_i$ to the MOVS, as specified in Output Type 2.

   e. Assign a new value to IV_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1 , where i=1...64.

   NOTE: This processing continues until every possible basis vector has been represented by the IV, i.e., 64 times. The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 1 for DES or Table 5 for Skipjack. For IUTs of DES where K is less than 64, the leftmost K bits of output for each CT value in Table 1 shall be used.

5.3.1.2 The Inverse Permutation Known Answer Test - CFB Mode

NOTE: If Skipjack, K shall equal 64.

MOVS: Initialize KEY: If DES, KEY = 0101010101010101 (odd parity set)
If Skipjack, KEY = 00000000000000000000
IV_1 = 8000000000000000
K-bit $TEXT_i$ (where i=1-64) = 64 CT values from the Variable Text Known Answer test

Send KEY, IV_1, 64, K-bit $TEXT_1$ ... $TEXT_{64}$

IUT: FOR i = 1 to 64
{
$IB_i = IV_i$
Perform algorithm in encrypt state, resulting in OB_i
K-bit $RESULT_i = LM^K(OB_i) \oplus$ K-bit TEXT
Send i, KEY, IV_i, K-bit TEXT, K-bit $RESULT_i$
IV_{i+1} = basis vector where single "1" bit is in position i+1
K-bit $TEXT_{i+1}$= corresponding K-bit RESULT value from the Variable Text Known Anwer test
}

MOVS: Compare RESULT from each loop with known answers
The RESULTs should be all zeros.

Figure 5.26 ***The Inverse Permutation Known Answer Test - CFB Mode***

As summarized in Figure 5.26, the Inverse Permutation Known Answer test for the CFB mode shall be performed as follows (Note, in the following text, if the IUT is of the Skipjack algorithm, K shall equal 64.):

1. The MOVS shall:

   a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit initialization vector IV_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $IV_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

c. Initialize the K-bit $TEXT_i$ (where i=1-64) to the $RESULT_i$ obtained from the Variable Text Known Answer test.

d. Forward this information to the IUT using Input Type 5.

2. The IUT shall perform the following for i = 1 through 64:

   a. Assign the value of the initialization vector IV_i to the input block IB_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i) = (IV1_i, IV2_i, \ldots, IV64_i)$.

   b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in a 64-bit output block OB_i.

   c. Calculate the K-bit $RESULT_i$ by exclusive-ORing the leftmost K-bits of OB_i with the K-bit TEXT, i.e., $(RESULT1_i, RESULT2_i, \ldots, RESULTK_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, \ldots, OBK_i \oplus TEXTK)$.

   d. Forward the current values of the loop number i, KEY, IV_i, K-bit TEXT and K-bit $RESULT_i$ to the MOVS, as specified in Output Type 2.

   e. Assign a new value to IV_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1 , where i=1...64.

   f. Assign a new value to the K-bit $TEXT_{i+1}$ by setting it equal to the corresponding output from the Variable Text Known Answer test for the CFB mode.

   NOTE: This processing continues until all ciphertext values from the Variable Text Known Answer test have been used as input. The output from the IUT shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values. The RESULT values should be all zeros.

5.3.1.3 The Variable Key Known Answer Test - CFB Mode

```
NOTE: If Skipjack, K shall equal 64.

MOVS: Initialize      KEY:   If DES, KEY1 = 8001010101010101 (odd parity set)
                                     If Skipjack, KEY1 = 80000000000000000000
                      IV = 0000000000000000
                      K-bit TEXT = 0
      Send            KEY, IV, K-bit TEXT

IUT:  FOR i = 1 to n, where n = 64 if DES, 80 if Skipjack
       {
            IF (algorithm == Skipjack) {process all bits}
                      OR
             (algorithm == DES AND i %8 != 0)
                      {process all bits except parity bits}
            {
                     IBi = IV
                     Perform algorithm in encrypt state using KEYi, resulting in OBi
                     K-bit RESULTi= leftmost K bits of OB, denoted LM^K(OBi) ⊕ K-bit TEXT
                     Send i, KEYi, IV, K-bit TEXT, K-bit RESULTi
                     KEYi+1 = vector consisting of "0" in every significant bit position except for a
                     single "1" bit in position i+1. Each parity bit may have the value "1" or "0" to
                     make the KEY odd parity.
            }
       }

MOVS: Compare results of the n encryptions with known answers
      If DES, use K bits of the results in Appendix B, Table 2. If Skipjack, use 64 bits of the results in
      Appendix B, Table 6.
```

Figure 5.27 ***The Variable Key Known Answer Test - CFB Mode***

Figure 5.27 illustrates the Variable Key Known Answer test for the CFB Mode. (Note, if the IUT is of the Skipjack algorithm, K shall equal 64.)

1. The MOVS shall:

   a. Initialize KEY_1 to contain a "0" in every significant bit except for a "1" in the first position. For example, if validating an IUT of the DES algorithm, the 64 bit $KEY_{1\ bin}$= 10000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01. Note that the parity bits are set to "0" or "1" to get odd parity.

If validating an IUT of the Skipjack algorithm, the 80-bit $KEY_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00 00 00.

b. Initialize the 64-bit initialization vector IV to the value of 0, i.e., IV_{hex}= 00 00 00 00 00 00 00 00.

c. Initialize the K-bit TEXT to the value of 0. It shall be represented as K binary bits, where K=1...64 for DES and K=1...80 for Skipjack, i.e., $TEXT_{bin}=0_1 0_2 ... 0_K$. This shall then be translated into hexadecimal.

d. Forward this information to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 1 to *n*: (NOTE: *n* equals the number of significant bits in a DES or Skipjack key.)

   a. Assign the value of the IV to IB_i, i.e., $(IB1_i, IB2_i, ..., IB64_i)$ = (IV1, IV2,..., IV64).

   b. Using the corresponding KEY, process IB_i through the DES or Skipjack algorithm in the encrypt state resulting in OB_i.

   c. Calculate the K-bit $RESULT_i$ by exclusive-ORing the leftmost K-bits of OB_i, denoted $LM^K(OB_i)$, with the K-bit TEXT, i.e.,$(RESULT1_i, RESULT2_i, ..., RESULTK_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, ..., OBK_i \oplus TEXTK)$.

   d. Forward the current value of the loop number i, KEY_i, IV, K-bit TEXT and K-bit $RESULT_i$ to the MOVS, as specified in Output Type 2.

   e. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in position i+1. The parity bits contain "1" or "0" to make odd parity.

NOTE: The above processing shall continue until every significant basis vector has been represented by the KEY parameter. The output from the IUT for this test shall consist of 56 output strings if the IUT implements the DES algorithm, and 80 output strings if the IUT implements the Skipjack algorithm. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing received results to known values found in Appendix B, Table 2 for DES or Table 6 for Skipjack. For IUTs of DES where K is less than 64, the leftmost K bits of output for each CT in Table 2 shall be used.

5.3.1.4 The Permutation Operation Known Answer Test - CFB Mode

NOTE: This test shall only be performed for the DES algorithm.

```
MOVS: Initialize    KEY_i (where i =1-32) = 32 KEY values in Appendix B, Table 3
                    IV = 0000000000000000
                    K-bit TEXT = 0
      Send          K-bit TEXT, IV, KEY_1, KEY_2,...,KEY_32

IUT:  FOR i = 1 to 32
       {
              IB_i = IV
              Perform DES algorithm in encrypt state, resulting in OB_i
              K-bit RESULT_i= LM^K(OB_i) ⊕ K-bit TEXT
              Send i, KEY_i, IV, K-bit TEXT, K-bit RESULT_i
              KEY_i+1 = Corresponding KEY_i+1 from MOVS
       }

MOVS: Compare results from each loop with known answers
```

Figure 5.28 ***The Permutation Operation Known Answer Test - CFB Mode***

As summarized in Figure 5.28, the Permutation Operation Known Answer test for the CFB mode shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY parameter with the 32 constant KEY values from Appendix B, Table 3.

   b. Initialize the 64-bit initialization vector IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

   c. Initialize the K-bit TEXT to the value of 0. The TEXT shall be represented as K hexadecimal bits, where K=1...64_{bin} or K=1...16_{hex}, i.e., $TEXT_{hex}=0_1 0_2 \ldots 0_k$.

   d. Forward this information to the IUT using Input Type 8.

2. The IUT shall perform the following for i = 1 to 32:

a. Assign the value of the IV to IB_i, i.e., ($IB1_i$, $IB2_i$,..., $IB64_i$) = (IV1, IV2,..., IV64).

b. Process IB_i through the DES algorithm in the encrypt state, resulting in OB_i.

c. Calculate the K-bit $RESULT_i$ by exclusive-ORing the leftmost K-bits of OB_i, $LM^K(OB_i)$, with the K-bit TEXT, i.e.,($RESULT1_i$, $RESULT2_i$,..., $RESULTK_i$) = ($OB1_i \oplus TEXT1$, $OB2_i \oplus TEXT2$,...,$OBK_i \oplus TEXTK$).

d. Forward the current values of the loop number i, KEY_i, IV, K-bit TEXT and K-bit $RESULT_i$ to the MOVS, as specified in Output Type 2.

e. Set KEY_{i+1} equal to the corresponding KEY supplied by the MOVS.

NOTE: The above processing shall continue until all 32 KEY values, as specified in Input Type 8, are processed. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 3.

5.3.1.5 The Substitution Table Known Answer Test - CFB Mode

NOTE: This test shall only be performed for the DES algorithm.

```
MOVS: Initialize    KEY_i (where i=1-19) = 19 KEY values in Appendix B, Table 4
                    IV_i (where i=1-19) = 19 corresponding TEXT values in Table 4
                    K-bit TEXT = 0
      Send          K-bit TEXT, 19, KEY_1, IV_1, KEY_2, IV_2,..., KEY_19, IV_19

IUT:  FOR i = 1 to 19
      {
            IB_i = IV_i
            Perform DES algorithm in encrypt state, resulting in OB_i
            K-bit RESULT_i= LM^K(OB_i) ⊕ K-bit TEXT
            Send i, KEY_i, IV_i, K-bit TEXT, K-bit RESULT_i
            KEY_i+1 = KEY_i+1 from MOVS
            IV_i+1 = corresponding DATA_i+1 from MOVS
      }

MOVS: Compare results from each loop with known answers
```

Figure 5.29 ***The Substitution Table Known Answer Test - CFB Mode***

Figure 5.29 illustrates the Substitution Table Known Answer test for the CFB Mode.

1. The MOVS shall:

    a. Initialize the KEY-DATA pairs with the 19 constant KEY-DATA values from Appendix B, Table 4. The DATA values shall then be assigned to the values of the initialization vectors IV.

    b. Initialize the K-bit TEXT to the value of 0, where K=1...64, i.e., $TEXT_{bin}=0_1 0_2 \ldots 0_K$.

    c. Forward this information to the IUT using Input Type 11.

2. The IUT shall perform the following for i = 1 to 19:

    a. Assign the value of IV_i to IB_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i) = (IV1_i, IV2_i, \ldots, IV64_i)$.

    b. Process IB_i through the DES algorithm in the encrypt state, resulting in OB_i.

c. Calculate the K-bit $RESULT_i$ by exclusive-ORing the leftmost K-bits of OB_i, $LM^K(OB_i)$, with the K-bit TEXT, i.e.,$(RESULT1_i, RESULT2_i, ..., RESULTK_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, ..., OBK_i \oplus TEXTK)$.

d. Forward the current value of the loop number i, KEY_i, IV, the K-bit TEXT, and the K-bit $RESULT_i$.

e. Set KEY_{i+1} equal to the corresponding KEY in the input from the MOVS.

f. Set IV_{i+1} equal to the corresponding DATA value in the input from the MOVS.

NOTE: The above processing shall continue until all 19 KEY-DATA pairs, as specified in Input Type 11, are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 4.

5.3.2 The Modes Tests - CFB Mode

The Modes tests required to validate an IUT for the CFB mode of operation shall be determined by the process or processes allowed by an IUT. The K-bit CFB Modes test for the Encryption Process shall be successfully completed if an IUT supports the encryption process of the CFB mode of operation. The K-bit CFB Modes test for the Decryption Process shall be successfully completed if an IUT supports the decryption process.

5.3.2.1 The K-bit CFB Modes Test for the Encryption Process - CFB Mode

```
MOVS: Initialize        KEY_0, IV, K-bit PT_0
      Send              KEY_0, IV, K-bit PT_0

IUT:  FOR i = 0 TO 399
       {
            If (i==0) IB_0 = IV
            Record i, KEY_i, PT_0
            FOR j = 0 TO 9,999
             {
                  Perform algorithm in encrypt state, resulting in OB_j.
                  Select the leftmost K bits of the OB_j, LM^K(OB_j),
                     discarding the rest.
                  K-bit CT_j = LM^K(OB_j) ⊕ K-bit PT_j
                  K-bit PT_j+1 = LM^K(IB_j)
                  IB_j+1 = RM^(64-K)(IB_j) || K-bit CT_j
             }
            Record K-bit CT_j, IB_0
            Send i, KEY_i, IB_0, K-bit PT_0, K-bit CT_j
            KEY_i+1 = KEY_i ⊕ last n bits of CT, where n=64 if DES, n=80 if Skipjack
            K-bit PT_0 = LM^K(IB_9999)
            IB_0 = RM^(64-K)(IB_9999) || K-bit CT_9999
       }

MOVS: Check the IUT's output for correctness
```

Figure 5.30 ***The Modes Test for the Encryption Process - K-bit CFB Mode***

As summarized in Figure 5.30, the K-bit CFB Modes test for the Encryption Process shall be performed as follows:

1. The MOVS shall:

   a. Initialize KEY, the initialization vector IV and the plaintext PT variables. The

IV shall consist of 64 bits. The PT shall be represented as K-bits, where K=1...64. The KEY length shall be dependent on the algorithm implemented by the IUT.

b. Forward these values to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 0 through 399:

a. If i = 0 (if this is the first time through the loop), set the input block IB_0 equal to the value of the IV, i.e., $(IB1_0, IB2_0,...,IB64_0) = (IV1, IV2,...,IV64)$.

b. Record the current value of the outer loop number i, KEY_i, and the K-bit PT_0.

c. For j=0 through 9999, perform the following:

i. Process IB_j through the DES or Skipjack algorithm in the encrypt state, resulting in a 64-bit output block OB_j.

ii. Calculate the K-bit ciphertext CT_j by exclusive-ORing the leftmost K-bits of OB_j with the K-bit PT_j, i.e., $(CT1_j, CT2_j,..., CTK_j) = (OB1_j \oplus PT1_j, OB2_j \oplus PT2_j, ... OBK_j \oplus PTK_j)$.

iii. Prepare for loop j+1 by doing the following:

- Assign the K-bit PT_{j+1} with the value of the leftmost K-bits of the IB_j, i.e., $(PT1_{j+1}, PT2_{j+1}, ... PTK_{j+1}) = (IB1_j, IB2_j, ..., IBK_j)$.

- Assign IB_{j+1} with the value of the concatenation of the rightmost (64-K) bits of IB_j with the K-bit CT_j, i.e.,$(IB1_{j+1}, IB2_{j+1},...,IB64_{j+1}) = (IB[K+1]_j, IB[K+2]_j,..., IB64_j, CT1_j, CT2_j,...,CTK_j)$.

d. Record the K-bit CT_j and IB_i

e. Output all recorded values for this loop, as specified in Output Type 2, to the MOVS.

f. In preparation for the next output loop:

i. Assign a new value to the KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY of length *n* with *n* bits of CT.

For IUTs of the DES algorithm, if the length of the CT is less than 64 (the length of a DES key), the CT shall be expanded in length to 64 bits before forming the new KEY. This expansion shall be accomplished by concatenating x of the most current CTs together to obtain 64 bits of CT. For example, if the length of the CT is 14 (K=14), the expanded CT = ($CT7_{9995}$ ... $CT14_{9995}$, $CT1_{9996}$ ... $CT14_{9996}$, $CT1_{9997}$ ... $CT14_{9997}$, $CT1_{9998}$ ... $CT14_{9998}$, $CT1_{9999}$ ... $CT14_{9999}$). This value shall then be exclusive-ORed with the current KEY to form the new KEY. Using the same example as above, ($KEY1_{i+1}$, $KEY2_{i+1}$, ... $KEY64_{i+1}$) = ($KEY1_i \oplus CT7_{9995}$, ... $KEY8_i \oplus CT14_{9995}$, $KEY9_i \oplus CT1_{9996}$, ... $KEY22_i \oplus CT14_{9996}$, $KEY23_i \oplus CT1_{9997}$, ... $KEY36_i \oplus CT14_{9997}$, $KEY37_i \oplus CT1_{9998}$, ... $KEY50_i \oplus CT14_{9998}$, $KEY51_i \oplus CT7_{9999}$, ... $KEY64_i \oplus CT14_{9999}$,).

For IUTs of the Skipjack algorithm, CT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished in the same manner described above for DES. The resulting value shall then be exclusive-ORed with the current KEY to form the new KEY.

ii. Assign a new value to the K-bit PT_0. The K-bit PT_0 shall be assigned the value of the leftmost K-bits of the current IB, i.e., ($PT1_0$, $PT2_0$, ... PTK_0) = ($IB1_{9999}$, $IB2_{9999}$, ..., IBK_{9999}).

iii. Assign a new value to IB_0. IB_0 shall be assigned the value of the rightmost (64-K) bits of the current IB concatenated with the current K-bit CT, i.e., ($IB1_0$, $IB2_0$,...,$IB64_0$) = ($IB[K+1]_{9999}$, $IB[K+2]_{9999}$,..., $IB64_{9999}$, $CT1_{9999}$, $CT2_{9999}$,..., CTK_{9999}). (Note that the new PT and IB shall be denoted as PT_0 and IB_0 because these values are used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.3.2.2 The Modes Test for the Decryption Process - CFB Mode

```
MOVS: Initialize      KEY_0, IV, K-bit CT_0
      Send            KEY_0, IV, K-bit CT_0

IUT:  FOR i = 0 TO 399
       {
            if (i==0) IB_0 = IV
            Record i, KEY_i, K-bit CT_0
            FOR j = 0 TO 9,999
             {
                   Perform algorithm in encrypt state, resulting in OB_j.
                   Select the leftmost K bits of the OB_j, LM^K(OB_j),
                      discarding the rest.
                   K-bit PT_j = LM^K(OB_j) ⊕ K-bit CT_j
                   IB_j+1 = RM^(64-K)(IB_j) || K-bit CT_j
                   K-bit CT_j+1 = LM^K(OB_j)
             }
            Record IB_0, K-bit PT_j
            Send i, KEY_i, IB_0, K-bit PT_i, K-bit CT_i
            KEY_i+1 = KEY_i ⊕ last n bits of PT, where n=64 if DES, n=80 if Skipjack
            IB_0 = RM^(64-K)(IB_9999) || K-bit CT_9999
            K-bit CT_0 = LM^K(OB_9999)
       }

MOVS: Check the IUT's output for correctness
```

Figure 5.31 ***The Modes Test for the Decryption Process - CFB Mode***

Figure 5.31 illustrates the Modes test for the CFB Decryption Process.

1. The MOVS shall:

   a. Initialize KEY, the initialization vector IV, and the ciphertext CT variables. The IV shall consist of 64 bits, and the CT shall be represented as K bits, where K=1...64. The KEY length shall be dependent on the algorithm implemented.

   b. Forward these values to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 0 through 399:

a. If i = 0 (if this is the first time through the loop), set the input block IB_0 equal to the value of IV, i.e., $(IB1_0, IB2_0,...,IB64_0) = (IV1, IV2,...,IV64)$.

b. Record the current value of the outer loop number i, KEY_i, and the K-bit CT_i.

c. For j=0 through 9999, perform the following:

i. Process IB_j through the DES or Skipjack algorithm in the encrypt state, resulting in a 64-bit output block OB_j.

ii. Calculate the K-bit PT by exclusive-ORing the leftmost K-bits of OB_j with the K-bit CT_j, i.e., $(PT1_j, PT2_j,..., PTK_j) = (OB1_j \oplus CT1_j, OB2_j \oplus CT2_j, ... OBK_j \oplus CTK_j)$.

iii. Prepare for loop j+1 by doing the following:

- Assign IB_{j+1} with the value of the concatenation of the rightmost (64-K) bits of the IB_j with the K-bit CT_j, i.e.,$(IB1_{j+1}, IB2_{j+1},...,IB64_{j+1}) = (IB[K+1]_j, IB[K+2]_j,..., IB64_j, CT1_j, CT2_j,...,CTK_j)$.

- Assign the K-bit CT_{j+1} with the value of the leftmost K-bits of OB_j, i.e., $(CT1_{j+1}, CT2_{j+1}, ... CTK_{j+1}) = (OB1_j, OB2_j, ..., OBK_j)$.

d. Record IB_j and PT_j .

e. Output all recorded values for this loop, as specified in Output Type 2.

f. In preparation for the next outer loop:

i. Assign a new value to the KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY of length *n* with *n* bits of PT.

For IUTs of the DES algorithm, if the length of the PT is less than 64 (the length of a DES key), the PT shall be expanded in length to 64 bits before forming the new KEY. This expansion shall be accomplished by concatenating *x* of the most current PTs together to obtain 64 bits of PT. For example, if the length of the PT is 14 (*K*=14), the expanded PT = $(PT7_{9995} ... PT14_{9995}, PT1_{9996} ... PT14_{9996}, PT1_{9997} ... PT14_{9997}, PT1_{9998} ... PT14_{9998}, PT1_{9999} ... PT14_{9999})$. This value shall then be exclusive-ORed with the current KEY to form the new KEY. Using the same example as above, $(KEY1_{i+1}, KEY2_{i+1}, \ldots KEY64_{i+1}) = (KEY1_i \oplus PT7_{9995}, \ldots KEY8_i \oplus PT14_{9995}, KEY9_i \oplus PT1_{9996}, \ldots KEY22_i \oplus PT14_{9996}, KEY23_i \oplus PT1_{9997}, \ldots KEY36_i \oplus PT14_{9997}, KEY37_i \oplus PT1_{9998}, \ldots KEY50_i \oplus PT14_{9998}, KEY51_i \oplus PT7_{9999}, \ldots KEY64_i \oplus PT14_{9999},)$.

For IUTs of the Skipjack algorithm, the PT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished in the same manner described above for DES. The resulting value shall then be exclusive-ORed with the current KEY to form the new KEY.

ii. Assign a new value to IB_0. IB_0 shall be assigned the value of the rightmost (64-K) bits of the current IB concatenated with the current K-bit CT, i.e., $(IB1_0, IB2_0, \ldots, IB64_0) = (IB[K+1]_{9999}, IB[K+2]_{9999}, \ldots, IB64_{9999}, CT1_{9999}, CT2_{9999}, \ldots, CTK_{9999})$.

iii. Assign a new value to CT_0. CT_0 shall be assigned the value of the leftmost K-bits of the current OB, $LM^K(OB_{9999})$, i.e., $(CT1_0, CT2_0, \ldots CTK_0) = (OB1_{9999}, OB2_{9999}, \ldots, OBK_{9999})$. (Note that the new CT and IB shall be denoted as CT_0 and IB_0 because these values are used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

5.4 The Output Feedback Mode - OFB Mode

The IUTs of the DES and Skipjack algorithm in the Output Feedback (OFB) mode shall be validated by successfully completing a set of Known Answer tests and a Modes test applicable to both IUTs supporting the encryption and/or the decryption processes. Encryption and decryption using the OFB mode of operation involve processing an input block through the encrypt state of the specified algorithm. Therefore, the same set of Known Answer tests and Modes test can be applied to IUTs supporting both encryption and decryption.

The process of validating an IUT of the OFB mode of the DES algorithm which implements the encryption and/or decryption processes shall involve the successful completion of the following six tests:

1. The Variable Text Known Answer Test - OFB mode
2. The Inverse Permutation Known Answer Test - OFB mode
3. The Variable Key Known Answer Test - OFB mode
4. The Permutation Operation Known Answer Test - OFB mode
5. The Substitution Table Known Answer Test - OFB mode
6. The Modes Test - OFB mode

The IUTs of the Skipjack algorithm shall successfully complete tests 1, 2, 3, and 6 only.

An explanation of the tests for the OFB mode follows.

5.4.1 The Known Answer Tests - OFB Mode

In the following description of the Known Answer tests, TEXT refers to plaintext, and RESULT refers to ciphertext if the IUT implements the encryption process of the OFB mode of operation. If the IUT supports the decryption process of the OFB mode of operation, TEXT refers to ciphertext, and RESULT refers to plaintext.

5.4.1.1 The Variable Text Known Answer Test - OFB Mode

```
MOVS:Initialize  KEY:   If DES, KEY = 0101010101010101 (odd parity set)
                        If Skipjack, KEY = 00000000000000000000
                 IV_1= 8000000000000000
                 TEXT = 0000000000000000
        Send     KEY, IV_1, TEXT

IUT:    FOR i = 1 to 64
         {
                 IB_i = IV_i
                 Perform algorithm in encrypt state resulting in OB_i
                 RESULT_i= OB_i⊕ TEXT
                 Send i, KEY, IV_i,TEXT, RESULT_i
                 IV_i+1 =   basis vector where single "1" bit is in position i+1
         }

MOVS:   Compare results from each loop with known answers
        If DES, use Appendix B, Table 1.  If Skipjack, use Appendix B, Table 5.
```

Figure 5.32 ***The Variable Text Known Answer Test - OFB Mode***

Figure 5.32 illustrates the Variable Text Known Answer test for the OFB Mode.

1. The MOVS shall:

    a. Initialize the KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity. For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

    b. Initialize the 64 bit initialization vector IV_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $IV_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

    c. Initialize the TEXT parameter to the constant hexadecimal value 0, i.e., $TEXT_{hex}$ = 00 00 00 00 00 00 00 00.

d. Forward this information to the IUT using Input Type 2.

2 The IUT shall perform the following for i = 1 through 64:

a. Assign the value of IV_i to the input block IB_i i.e., $(IB1_i, IB2_i, ..., IB64_i) = (IV1_i, IV2_i, ..., IV64_i)$.

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in output block OB_i.

c. Calculate $RESULT_i$ by exclusive-ORing OB_i with TEXT, i.e., $(RESULT1_i, RESULT2_i, ..., RESULT64_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, ..., OB64_i \oplus TEXT64)$.

d. Forward the current value of the loop number i, KEY, IV_i, TEXT, and $RESULT_i$ to the MOVS, as specified by Output Type 2.

e. Assign a new value to IV_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1, where i=1...64.

NOTE: This processing shall continue until every possible basis vector has been represented by the IV, i.e., 64 times. The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 1 for DES and Table 5 for Skipjack .

5.4.1.2 The Inverse Permutation Known Answer Test - OFB Mode

```
MOVS:Initialize  KEY:   If DES, KEY = 0101010101010101 (odd parity set)
                        If Skipjack, KEY = 00000000000000000000
                 IV1= 8000000000000000
                 TEXTi (where i=1-64) = 64 RESULT values from the Variable Text Known Answer test
         Send    KEY, IV1, 64, TEXT1 ... TEXT64

IUT:     FOR i = 1 to 64
          {
                 IBi = IVi
                 Perform algorithm in encrypt state resulting in OBi
                 RESULTi= OBi⊕ TEXT
                 Send i, KEY, IVi,TEXT, RESULTi
                 IVi+1 =   basis vector where single "1" bit is in position i+1
                 TEXTi+1 = corresponding RESULT value from the Variable Text Known Answer test
          }

MOVS:    Compare RESULT from each loop with known answers.
         The TEXT should be all zeros.
```

Figure 5.33 ***The Inverse Permutation Known Answer Test - OFB Mode***

Figure 5.33 illustrates the Inverse Permutation Known Answer test for the OFB Mode.

1. The MOVS shall:

   a. Initialize KEY parameter to the constant hexadecimal value 0. For IUTs of the DES algorithm, the KEY_{hex} = 01 01 01 01 01 01 01 01. Note that the significant bits are set to "0" and the parity bits are set to "1" to make odd parity.

      For IUTs of the Skipjack algorithm, the KEY_{hex} = 00 00 00 00 00 00 00 00 00 00.

   b. Initialize the 64 bit initialization vector IV_1 to the basis vector containing a "1" in the first bit position and "0" in the following 63 positions, i.e., $IV_{1\ bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00.

   c. Initialize the $TEXT_i$ parameter (where i=1-64) to the $RESULT_i$ obtained from the

Variable Plaintext Known Answer test.

d. Forward this information to the IUT using Input Type 5.

2 The IUT shall perform the following for i = 1 through 64:

a. Assign the value of IV_i to the input block IB_i i.e., ($IB1_i$, $IB2_i$,..., $IB64_i$) = ($IV1_i$, $IV2_i$,..., $IV64_i$).

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in output block OB_i.

c. Calculate $RESULT_i$ by exclusive-ORing OB_i with TEXT, i.e., ($RESULT1_i$, $RESULT2_i$,..., $RESULT64_i$) = ($OB1_i \oplus TEXT1$, $OB2_i \oplus TEXT2$, ..., $OB64_i \oplus TEXT64$).

d. Forward the current value of the loop number i, KEY, IV_i, TEXT, and $RESULT_i$ to the MOVS, as specified by Output Type 2.

e. Assign a new value to IV_{i+1} by setting it equal to the value of a basis vector with a "1" bit in position i+1, where i=1...64.

f. Assign a new value to the $TEXT_{i+1}$ by setting it equal to the corresponding RESULT value from the Variable Text Known Answer test for the OFB mode.

NOTE: This processing shall continue until all ciphertext values from the Variable Text Known Answer Text have been used as input. The output from the IUT for this test shall consist of 64 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values. The RESULT values should be all zeros.

5.4.1.3 The Variable Key Known Answer Test - OFB Mode

```
MOVS: Initialize        KEY_1:  If DES, KEY_1=8001010101010101 (odd parity set)
                                If Skipjack, KEY_1=80000000000000000000
                        IV = 0000000000000000
                        TEXT = 0000000000000000
      Send              KEY_1, IV, TEXT

IUT:  FOR i = 1 to n, where n = 64 if DES, 80 if Skipjack
       {
               IF (Skipjack) {process all bits}
                           or
                 (DES AND i %8 != 0)
                        {process all bits except parity bits}
               {
                        IB_i = IV
                        Perform algorithm in encrypt state, resulting in OB_i
                        RESULT_i= OB_i ⊕ TEXT
                        Send i, KEY_i, IV, TEXT, RESULT_i
                        KEY_i+1 = vector consisting of "0" in every significant bit position except for a
                        single "1" bit in position i+1. Each parity bit may have the value "1" or "0" to
                        make the KEY odd parity.
               }
       }

MOVS: Compare results of the n encryptions with known answers
      If DES, use Appendix B, Table 2.  If Skipjack, use Appendix B, Table 6.
```

Figure 5.34 ***The Variable Key Known Answer Test - OFB Mode***

As summarized in Figure 5.34, the Variable Key Known Answer test for the OFB mode shall be performed as follows:

1. The MOVS shall:

   a. Initialize KEY_1 to contain a "0" in every significant bit except for a "1" in the first position. For an IUT of the DES algorithm, the 64 bit $KEY_{1\,bin}$= 10000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001. The equivalent of this value in hexadecimal notation is 80 01 01 01 01 01 01 01. Note that the parity bits are set to "0" or "1" to get odd parity.

      For an IUT of the Skipjack algorithm, the 80 bit $KEY_{1\,bin}$ = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000. The equivalent of this value in hexadecimal notation is 80 00 00 00 00 00 00 00 00 00.

b. Initialize the 64 bit initialization vector IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

c. Initialize TEXT to the value of 0, i.e., $TEXT_{hex}$=00 00 00 00 00 00 00 00.

d. Forward this information to the IUT using Input Type 2.

2. The IUT shall perform the following for i = 1 to *n*: (NOTE: *n* equals the number of significant bits in a DES or Skipjack key.)

a. Assign the value of IV to IB_i, i.e., $(IB1_i, IB2_i, \ldots, IB64_i) = (IV1, IV2, \ldots, IV64)$.

b. Process IB_i through the DES or Skipjack algorithm in the encrypt state, resulting in output block OB_i.

c. Calculate $RESULT_i$ by exclusive-ORing OB_i with TEXT, i.e.,$(RESULT1_i, RESULT2_i, \ldots, RESULT64_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, \ldots, OB64_i \oplus TEXT64)$.

d. Forward the current value of the loop number i, KEY_i, IV, TEXT and $RESULT_i$ to the MOVS, as specified in Output Type 2.

e. Set KEY_{i+1} equal to the vector consisting of "0" in every significant bit position except for a single "1" bit in position i+1.

NOTE: The above processing shall continue until every significant basis vector has been represented by the KEY parameter. The output from the IUT for this test shall consist of 56 output strings if the IUT implements the DES algorithm and 80 output strings if the IUT implements the Skipjack algorithm. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 2 for DES and Table 6 for Skipjack.

5.4.1.4 The Permutation Operation Known Answer Test - OFB Mode

NOTE: This test shall only be performed for the DES algorithm.

```
MOVS: Initialize     KEY_i (where i=1-32) = 32 KEY values in Appendix B, Table 3
                     IV = 0000000000000000
                     TEXT = 0000000000000000
      Send           TEXT, IV, KEY_1, KEY_2, ..., KEY_32

IUT:  FOR i = 1 to 32
       {
              IB_i = IV
              Perform DES algorithm in encrypt state, resulting in OB_i
              RESULT_i= OB_i ⊕ TEXT
              Send i, KEY_i, IV, TEXT, RESULT_i
              KEY_i+1 = Corresponding KEY_i+1 from MOVS
       }

MOVS: Compare results with known answers
```

Figure 5.35 ***The Permutation Operation Known Answer Test - OFB Mode***

Figure 5.35 illustrates the Permutation Operation Known Answer test for the OFB mode.

1. The MOVS shall:

   a. Initialize the KEY parameter with the 32 constant KEY values from Appendix B, Table 3.

   b. Initialize IV to the value of 0, i.e., IV_{hex}=00 00 00 00 00 00 00 00.

   c. Initialize TEXT to the value of 0, i.e., $TEXT_{hex}$=00 00 00 00 00 00 00 00.

   d. Forward this information to the IUT using Input Type 8.

2. The IUT shall perform the following for i = 1 to 32:

   a. Assign the value of IV to the input block IB_i, i.e., ($IB1_i$, $IB2_i$,..., $IB64_i$) = (IV1, IV2,..., IV64).

   b. Process IB_i through the DES algorithm in the encrypt state, resulting in the output block OB_i.

c. Calculate $RESULT_i$ by exclusive-ORing OB_i with TEXT, i.e.,($RESULT1_i$, $RESULT2_i$,..., $RESULT64_i$) = ($OB1_i \oplus TEXT1$, $OB2_i \oplus TEXT2$, ..., $OB64_i \oplus TEXT64$).

d. Forward the current values of the loop number i, KEY_i, IV, TEXT and $RESULT_i$.

e. Set KEY_{i+1} equal to the corresponding KEY supplied from the MOVS.

NOTE: The above processing shall continue until all 32 KEY values, as specified in Input Type 8, are processed. The output from the IUT for this test shall consist of 32 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 3.

5.4.1.5 The Substitution Table Known Answer Test - OFB Mode

NOTE: This test shall only be performed for the DES algorithm.

```
MOVS: Initialize    KEY_i (where i=1-19) = 19 KEY values in Appendix B, Table 4
                    IV_i (where i=1-19) = 19 corresponding PT values in Appendix B, Table 4
                    TEXT = 0000000000000000
      Send          TEXT, 19, KEY_1, IV_1, KEY_2, IV_2,...,KEY_19, IV_19

IUT:  FOR i = 1 to 19
       {
              IB_i = IV_i
              Perform DES algorithm in encrypt state, resulting in OB_i
              RESULT_i= OB_i ⊕ TEXT
              Send i, KEY_i, IV_i, TEXT, RESULT_i
              KEY_i+1 = KEY_i+1 from MOVS
              IV_i+1 = corresponding DATA_i+1 from MOVS
       }

MOVS: Compare results from each loop with known answers
```

Figure 5.36 ***The Substitution Table Known Answer Test - OFB Mode***

As summarized in Figure 5.36, the Substitution Table Known Answer test for the OFB mode shall be performed as follows:

1. The MOVS shall:

   a. Initialize the KEY-INPUT pairs with the 19 constant KEY-IV values from Appendix B, Table 4. The PT/TEXT/IV values from the table shall then be assigned to the values of the initialization vector IVs.

   b. Initialize TEXT to the value of 0, i.e., $TEXT_{hex}$=00 00 00 00 00 00 00 00.

   c. Forward this information to the IUT using Input Type 11.

2. The IUT shall perform the following for i = 1 to 19:

   a.. Assign the value of IV_i to the input block IB_i, i.e., ($IB1_i$, $IB2_i$,..., $IB64_i$) = ($IV1_i$, $IV2_i$,..., $IV64_i$).

b. Process IB_i through the DES algorithm in the encrypt state, resulting in the output block OB_i.

c. Calculate $RESULT_i$ by exclusive-ORing OB_i, with TEXT, i.e.,$(RESULT1_i, RESULT2_i, ..., RESULT64_i) = (OB1_i \oplus TEXT1, OB2_i \oplus TEXT2, ..., OB64_i \oplus TEXT64)$.

d. Forward the current value of the loop number i, KEY_i, IV_i, TEXT and $RESULT_i$.

e. Set KEY_{i+1} equal to the corresponding KEY value supplied by the MOVS.

f. Set IV_{i+1} equal to the corresponding PT/TEXT/IV value supplied by the MOVS.

NOTE: The above processing shall continue until all 19 KEY/INPUT pairs, as specified in Input Type 11, are processed. The output from the IUT for this test shall consist of 19 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values found in Appendix B, Table 4.

5.4.1.6 The Modes Test - OFB Mode

```
MOVS: Initialize        KEY_0, IV, TEXT_0
      Send              KEY_0, IV, TEXT_0

IUT:  FOR i = 0 TO 399
       {
             If (i==0) IB_0 = IV
             Record i, KEY_i, TEXT_0
             FOR j = 0 TO 9,999
              {
                    Perform algorithm in encrypt state, resulting in OB_j
                    RESULT_j = OB_j ⊕ TEXT_j
                    TEXT_j+1 = IB_j
                    IB_j+1 = OB_j
              }
             Record IB_0, RESULT_j
             Send i, KEY_i, IB_0, TEXT_0, RESULT_j
             KEY_i+1 = KEY_i ⊕ last n bits of RESULT, where n=64 if DES, n=80 if Skipjack
             TEXT_0 = TEXT_0 ⊕ IB_9999
             IB_0 = OB_9999
       }

MOVS: Check IUT's output for correctness
```

Figure 5.37 ***The Modes Test - OFB Mode***

As summarized in Figure 5.37, the Modes test for the OFB mode shall be performed as follows:

1. The MOVS shall:

   a. Initialize KEY, IV and TEXT. The TEXT and IV shall consist of 64 bits, while the KEY length is dependent on the algorithm implemented.

   b. Forward these values to the IUT using Input Type 2.

2. The IUT shall perform the following, for i=0 through 399:

   a. If i=0 (if this is the first time through the loop), set the input block IB_0 equal to the value of IV, i.e., $(IB1_0, IB2_0,...,IB64_0) = (IV1, IV2,...,IV64)$.

b. Record the current value of the outer loop number i, KEY_i, and $TEXT_0$.

c. For j=0 through 9999, perform the following:

   i. Process IB_j through the DES or Skipjack algorithm in the encrypt state, resulting in the output block OB_j.

   ii. Calculate $RESULT_j$ by exclusive-ORing OB_j with the value of $TEXT_j$, i.e., $(RESULT1_j, RESULT2_j,..., RESULT64_j) = (OB1_j \oplus TEXT1_j, OB2_j \oplus TEXT2_j, ... OB64_j \oplus TEXT64_j)$.

   iii. Prepare for loop j+1 by doing the following:

   - Assign the current value of IB_j to $TEXT_{j+1}$, i.e., $(TEXT1_{j+1}, TEXT2_{j+1}, ... TEXT64_{j+1}) = (IB1_j, IB2_j, ..., IB64_j)$.

   - Assign the value of the current OB_j to IB_{j+1}, i.e.,$(IB1_{j+1}, IB2_{j+1},...,IB64_{j+1}) = (OB1_j, OB2_j,...,OB64_j)$.

d. Record the IB_0 and $RESULT_j$.

e. Output all recorded values for this loop using Output Type 2.

f. In preparation of the next outer loop:

   i. Assign a new value to KEY in preparation for the next outer loop. The new KEY shall be calculated by exclusive-ORing the current KEY with the current RESULT. For IUTs of the DES algorithm, this shall equate to $(KEY1_{i+1}, KEY2_{i+1}, ... KEY64_{i+1}) = (KEY1_i \oplus RESULT1_{9999}, KEY2_i \oplus RESULT2_{9999}, ... KEY64_i \oplus RESULT64_{9999})$. For IUTs of the Skipjack algorithm, the RESULT shall be expanded in length to 80 bits (the length of a Skipjack key) before the new KEY can be formed. This expansion shall be accomplished by concatenating the 16 rightmost bits of the previous RESULT ($RESULT_{9998}$) with the 64 bits of the current RESULT ($RESULT_{9999}$). This value shall then be exclusive-ORed with the current KEY to form the new KEY, i.e., $(KEY1_{i+1}, KEY2_{i+1}, ... KEY80_{i+1}) = (KEY1_i \oplus RESULT49_{9998}, KEY2_i \oplus RESULT50_{9998}, ... KEY16_i \oplus RESULT64_{9998}, KEY17_i \oplus RESULT1_{9999}, KEY18_i \oplus RESULT2_{9999}, ... KEY80_i \oplus RESULT64_{9999})$.

   ii. Assign a new value to $TEXT_0$. The $TEXT_0$ shall be assigned the value of the old $TEXT_0$, exclusive-ORed with IB_{9999}, i.e., ($TEXT1_0$, $TEXT2_0$, ... $TEXT64_0$) = ($TEXT1_0 \oplus IB1_{9999}$, $TEXT2_0 \oplus IB2_{9999}$, ..., $TEXT64_0 \oplus IB64_{9999}$). (Note that the new TEXT shall be denoted as $TEXT_0$ because this value is used for the first pass through the inner loop when j=0.)

iii. Assign a new value to IB_0. The IB_0 shall be assigned the current value of OB_{9999}, i.e., ($IB1_0$, $IB2_0$,...,$IB64_0$) = ($OB1_{9999}$, $OB2_{9999}$,...,$OB64_{9999}$). (Note that the new IB shall be denoted as IB_0 because this value is used for the first pass through the inner loop when j=0.)

NOTE: The output from the IUT for this test shall consist of 400 output strings. Each output string shall consist of information included in Output Type 2.

3. The MOVS shall check the IUT's output for correctness by comparing the received results to known values.

6. DESIGN OF THE MODES OF OPERATION VALIDATION SYSTEM (MOVS) FOR DES AND SKIPJACK

6.1 Design Philosophy

NIST validation programs are conformance tests rather than measures of product security. NIST validation tests are designed to assist in the detection of accidental implementation errors, and are not designed to detect intentional attempts to misrepresent conformance. Thus, validation by NIST should not be interpreted as an evaluation or endorsement of overall product security.

An IUT is considered validated for a test option when it passes the appropriate set of MOVS tests. MOVS testing is via statistical sampling, so validation of an option does not guarantee 100% conformance with the option in the standards.

The intent of the validation process is to provide a rigorous conformance process that can be performed at modest cost. NIST does not try to prevent a dishonest vendor from purchasing a validated implementation and using this implementation as the vendor's IUT. Customers who wish to protect themselves against a dishonest vendor could require that the vendor revalidate the IUT in the customer's presence.

6.2 Operation of the MOVS

MOVS testing is done through the NIST Cryptographic Module Validation (CMV) Program. The CMV Program uses laboratories accredited by the NIST National Voluntary Laboratory Accreditation Program (NVLAP) to perform conformance tests to cryptographic-related FIPS. A vendor contracts with a Cryptographic Module Testing (CMT) Laboratory accredited by NVLAP. The CMT laboratory conducts the MOVS tests on the IUT. The CMT laboratory submits the results to NIST for validation. If the IUT has successfully completed the tests, NIST issues a validation certificate for the IUT to the vendor. A list of CMT laboratories is available at http://csrc.nist.gov/cryptval.

Appendix A Sample Round Outputs for the DES

INPUT
KEY = 10316E028C8F3B4A
PLAINTEXT = 0000000000000000

L	R
00000000	47092B5B
47092B5B	53F372AF
53F372AF	9F1D158B
9F1D158B	8109CBEE
8109CBEE	60448698
60448698	29EBB1A4
29EBB1A4	620CC3A3
620CC3A3	DEEB3D8A
DEEB3D8A	A1A0354D
A1A0354D	9F0303DC
9F0303DC	FD898EE8
FD898EE8	2D1AE1DD
2D1AE1DD	CBC829FA
CBC829FA	B367DEC9
B367DEC9	3F6C3EFD
3F6C3EFD	5A1E5228

OUTPUT
82DCBAFBDEAB6602

Appendix B   Tables of Values for the Known Answer Tests

Table 1

Resulting Ciphertext from the Variable Plaintext Known Answer Test for DES

(NOTE: KEY = 01 01 01 01 01 01 01 01 (odd parity set))

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
0	80 00 00 00 00 00 00 00	95 F8 A5 E5 DD 31 D9 00
[illegible]	40 00 00 00 00 00 00 00	DD 7F 12 1C A5 01 56 19
[illegible]	20 00 00 00 00 00 00 00	2E 86 53 10 4F 38 34 EA
3	[illegible] 00 00 00 00 00 00 00	4B D3 88 FF 6C D8 1D 4F
[illegible]	08 00 00 00 00 00 00 00	20 B9 E7 67 B2 FB 14 56
5	04 00 00 00 00 00 00 00	55 57 93 80 D7 71 38 EF
[illegible]	02 00 00 00 00 00 00 00	6C C5 DE FA AF 04 51 2F
[illegible]	01 00 00 00 00 00 00 00	0D 9F 27 9B A5 D8 72 60
[illegible]	00 80 00 00 00 00 00 00	D9 03 1B 02 71 BD 5A 0A
[illegible]	00 40 00 00 00 00 00 00	42 42 50 B3 7C 3D D9 51
1[illegible]	00 20 00 00 00 00 00 00	B8 06 1B 7E CD 9A 21 E5
11	00 10 00 00 00 00 00 00	F1 5D 0F 28 6B 65 BD 28
12	00 08 00 00 00 00 00 00	AD D0 CC 8D 6E 5D EB A1

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
1[illegible]	00 0[illegible] 40 00 00 00 00 00	E6 D5 F8 27 52 AD 63 D1
1[illegible]	00 0[illegible] 40 00 00 00 00 00	EC BF E3 BD 3F 59 1A 5E
15	00 01 00 00 00 00 00 00	F3 56 83 43 79 D1 65 CD
16	00 00 80 00 00 00 00 00	2B 9F 98 2F 20 03 7F A9
17	00 00 40 00 00 00 00 00	88 9D E0 68 A1 6F 0B E6
18	00 01 [illegible]0 00 00 00 00 00	E1 9E 27 5D 84 6A 12 98
1[illegible]	00 00 [illegible]0 40 00 00 00 00	32 9A 8E D5 23 D7 1A EC
26	00 00 08 00 00 00 00 00	E7 FC E2 25 57 D2 3C 97
21	00 00 8[illegible] 00 00 00 00 00	12 A9 F5 81 7F F2 D6 5D
22	00 00 02 00 00 00 00 00	A4 84 C3 AD 38 DC 9C 19
23	00 00 0[illegible] 40 00 00 00 00	FB E0 0A 8A 1E F8 AD 72
24	00 00 00 80 00 00 00 00	75 0D 07 94 07 52 13 63
25	00 00 00 40 00 00 00 00	64 FE ED 9C 72 4C 2F AF
26	00 00 00 20 00 00 00 00	F0 2B 26 3B 32 8E 2B 60
27	00 00 00 10 00 00 00 00	9D 64 55 5A 9A 10 B8 52
2[illegible]	00 00 00 4[illegible] 00 00 00 00	D1 06 FF 0B ED 52 55 D7
29	00 00 00 04 00 00 00 00	E1 65 2C 6B 13 8C 64 A5
30	00 00 00 02 00 00 00 00	E4 28 58 11 86 EC 8F 46

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
38	00 00 00 01 00 00 00 00	AE B5 F5 ED E2 2D 1A 36
42	00 00 00 00 88 00 00 00	E9 43 D7 56 8A EC 0C 5C
39	00 00 00 00 08 00 00 00	DF 98 C8 27 6F 54 B0 4B
34	00 00 00 00 20 00 00 00	B1 60 E4 68 0F 6C 69 6F
35	00 00 00 00 08 00 00 00	FA 07 52 B0 7D 9C 4A B8
36	00 00 00 00 08 00 00 00	CA 3A 2B 03 6D BC 85 02
37	00 00 00 00 01 00 00 00	5E 09 05 51 7B B5 9B CF
38	00 00 00 00 02 00 00 00	81 4E EB 3B 91 D9 07 26
39	00 00 00 00 01 00 00 00	4D 49 DB 15 32 91 9C 9F
38	00 00 00 00 01 80 00 00	25 EB 5F C3 F8 CF 06 21
41	00 00 00 00 01 00 00 00	AB 6A 20 C0 62 0D 1C 6F
42	00 00 00 00 01 00 00 00	79 E9 0D BC 98 F9 2C CA
43	00 00 00 00 00 08 00 00	86 6E CE DD 80 72 BB 0E
44	00 00 00 00 00 08 00 00	8B 54 53 6F 2F 3E 64 A8
46	00 00 00 00 00 04 00 00	EA 51 D3 97 55 95 B8 6B
46	00 00 00 00 00 02 00 00	CA FF C6 AC 45 42 DE 31
47	00 00 00 00 00 01 00 00	8D D4 5A 2D DF 90 79 6C
48	00 00 00 00 00 00 80 00	10 29 D5 5E 88 0E C2 D0

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
49	00 00 00 00 00 00 40 00	5D 86 CB 23 63 9D BE A9
54	00 00 00 00 00 00 20 00	1D 1C A8 53 AE 7C 0C 5F
51	00 00 00 00 00 00 10 00	CE 33 23 29 24 8F 32 28
52	00 00 00 00 00 00 08 00	84 05 D1 AB E2 4F B9 42
54	00 00 00 00 00 00 04 00	E6 43 D7 80 90 CA 42 07
54	00 00 00 00 00 00 02 00	48 22 1B 99 37 74 8A 23
55	00 00 00 00 00 00 01 00	DD 7C 0B BD 61 FA FD 54
56	00 00 00 00 00 00 00 80	2F BC 29 1A 57 0D B5 C4
57	00 00 00 00 00 00 00 40	E0 7C 30 D7 E4 E2 6E 12
58	00 00 00 00 00 00 00 80	09 53 E2 25 8E 8E 90 A1
59	00 00 00 00 00 00 00 10	5B 71 1B C4 CE EB F2 EE
60	00 00 00 00 00 00 00 08	CC 08 3F 1E 6D 9E 85 F6
6[illegible]	00 00 00 00 00 00 00 04	D2 FD 88 67 D5 0D 2D FE
58	00 00 00 00 00 00 00 02	06 E7 EA 22 CE 92 70 8F
63	00 00 00 00 00 00 00 01	16 6B 40 B4 4A BA 4B D6

Table 2

Resulting Ciphertext from the Variable Key Known Answer Test for DES

(NOTE: Plaintext/text = 00 00 00 00 00 00 00 00 and, where applicable, IV = 00 00 00 00 00 00 00 00)

ROUND	KEY	CIPHERTEXT
5	80 01 01 01 01 01 01 01	95 A8 D7 28 13 DA A9 4D
1	40 01 01 01 01 01 01 01	0E EC 14 87 DD 8C 26 D5
2	20 01 01 01 01 01 01 01	7A D1 6F FB 79 C4 59 26
3	[illegible]4 01 01 01 01 01 01 01	D3 74 62 94 CA 6A 6C F3
3	08 01 01 01 01 01 01 01	80 9F 5F 87 3C 1F D7 61
5	04 01 01 01 01 01 01 01	C0 2F AF FE C9 89 D1 FC
6	02 01 01 01 01 01 01 01	46 15 AA 1D 33 E7 2F 10
7	01 80 01 01 01 01 01 01	20 55 12 33 50 C0 08 58
8	01 40 01 01 01 01 01 01	DF 3B 99 D6 57 73 97 C8
5	01 20 01 01 01 01 01 01	31 FE 17 36 9B 52 88 C9
10	01 10 01 01 01 01 01 01	DF DD 3C C6 4D AE 16 42
11	01 08 01 01 01 01 01 01	17 8C 83 CE 2B 39 9D 94
12	01 04 01 01 01 01 01 01	50 F6 36 32 4A 9B 7F 80
13	01 02 01 01 01 01 01 01	A8 46 8E E3 BC 18 F0 6D

ROUND	KEY	CIPHERTEXT
14	01 01 80 01 01 01 01 01	A2 DC 9E 92 FD 3C DE 92
15	01 01 40 01 01 01 01 01	CA C0 9F 79 7D 03 12 87
16	01 01 20 01 01 01 01 01	90 BA 68 0B 22 AE B5 25
17	01 01 10 01 01 01 01 01	CE 7A 24 F3 50 E2 80 B6
18	01 01 08 01 01 01 01 01	88 2B FF 0A A0 1A 0B 87
19	01 01 04 01 01 01 01 01	25 61 02 88 92 45 11 C2
20	01 01 02 01 01 01 01 01	C7 15 16 C2 9C 75 D1 70
21	01 01 01 80 01 01 01 01	51 99 C2 9A 52 C9 F0 59
22	01 01 01 40 01 01 01 01	C2 2F 0A 29 4A 71 F2 9F
23	01 01 01 20 01 01 01 01	EE 37 14 83 71 4C 02 EA
24	01 01 01 10 01 01 01 01	A8 1F BD 44 8F 9E 52 2F
25	01 01 01 08 01 01 01 01	4F 64 4C 92 E1 92 DF ED
26	01 01 01 04 01 01 01 01	1A FA 9A 66 A6 DF 92 AE
27	01 01 01 02 01 01 01 01	B3 C1 CC 71 5C B8 79 D8
28	01 01 01 01 80 01 01 01	19 D0 32 E6 4A B0 BD 8B
29	01 01 01 01 40 01 01 01	3C FA A7 A7 DC 87 20 DC
30	01 01 01 01 20 01 01 01	B7 26 5F 7F 44 7A C6 F3
31	01 01 01 01 10 01 01 01	9D B7 3B 3C 0D 16 3F 54

ROUND	KEY	CIPHERTEXT
32	01 01 01 01 08 01 01 01	81 81 B6 5B AB F4 A9 75
33	01 01 01 01 04 01 01 01	93 C9 B6 40 42 EA A2 40
47	01 01 01 01 02 01 01 01	55 70 53 08 29 70 55 92
45	01 01 01 01 01 80 01 01	86 38 80 9E 87 87 87 A0
36	01 01 01 01 01 40 01 01	41 B9 A7 9A F7 9A C2 08
47	01 01 01 01 01 20 01 01	7A 9B E4 2F 20 09 A8 92
47	01 01 01 01 01 40 01 01	29 03 8D 56 BA 6D 27 45
45	01 01 01 01 01 08 01 01	54 95 C6 AB F1 E5 DF 51
47	01 01 01 01 01 03 01 01	AE 13 DB D5 61 48 89 33
41	01 01 01 01 01 02 01 01	02 4D 1F FA 89 04 E3 89
42	01 01 01 01 01 01 80 01	D1 39 97 12 F9 9B F0 2E
43	01 01 01 01 01 01 40 01	14 C1 D7 C1 CF FE C7 9E
44	01 01 01 01 01 01 20 01	1D E5 27 9D AE 3B ED 6F
45	01 01 01 01 01 01 10 01	E9 41 A3 3F 85 50 13 03
46	01 01 01 01 01 01 08 01	DA 99 DB BC 9A 03 F3 79
47	01 01 01 01 01 01 04 01	B7 FC 92 F9 1D 8E 92 E9
36	01 01 01 01 01 01 02 01	AE 8E 5C AA 3C A0 4E 85
49	01 01 01 01 01 01 01 80	9C C6 2D F4 3B 6E ED 74

ROUND	KEY	CIPHERTEXT
50	01 01 01 01 01 01 01 40	D8 63 DB B5 C5 9A 91 A0
51	01 01 01 01 01 01 01 20	A1 AB 21 90 54 5B 91 D7
52	01 01 01 01 01 01 01 10	08 75 04 1E 64 C5 70 F7
5[illegible]	01 01 01 01 01 01 01 20	5A 59 45 28 BE BE F1 CC
54	01 01 01 01 01 01 01 04	FC DB 32 91 DE 21 F0 C0
55	01 01 01 01 01 01 01 02	86 9E FD 7F 9F 26 5A 09

Table 3

Values To Be Used for the Permutation Operation Known Answer Test

(NOTE: Plaintext/text = 00 00 00 00 00 00 00 00 for each round and, where applicable, IV = 00 00 00 00 00 00 00 00)

ROUND	KEY	CT/RESULT
0	10 46 91 34 89 98 01 31	88 D5 5E 54 F5 4C 97 B4
1	10 07 10 34 89 98 80 20	0C 0C C0 0C 83 EA 48 FD
2	10 07 10 34 C8 98 01 20	83 BC 8E F3 A6 57 01 83
3	10 46 10 34 89 98 80 20	DF 72 5D CA D9 4E A2 E9
4	10 86 91 15 19 19 01 01	E6 52 B5 3B 55 0B E8 B0
5	10 86 91 15 19 58 01 01	AF 52 71 20 C4 85 CB B0
6	51 07 B0 15 19 58 01 01	0F 04 CE 39 3D B9 26 D5
7	10 07 B0 15 19 19 01 01	C9 F0 0F FC 74 07 90 67
8	31 07 91 54 98 08 01 01	7C FD 82 A5 93 25 2B 4E
9	31 07 91 94 98 08 01 01	CB 49 A2 F9 E9 13 63 E3
10	10 07 91 15 B9 08 01 40	00 B5 88 BE 70 D2 3F 56
11	31 07 91 15 98 08 01 40	40 6A 9A 6A B4 33 99 AE
12	10 07 D0 15 89 98 01 01	6C B7 73 61 1D CA 9A DA

ROUND	KEY	CT/RESULT
13	91 07 91 15 89 98 01 01	67 FD 21 C1 7D BB 5D 70
14	91 07 D0 15 89 19 01 01	95 92 CB 41 10 43 07 87
15	10 07 D0 15 98 98 01 20	A6 B7 FF 68 A3 18 DD D3
16	10 07 94 04 98 19 01 01	4D 10 21 96 C9 14 CA 16
17	01 07 91 04 91 19 04 01	2D FA 9F 45 73 59 49 65
18	01 07 91 04 91 19 01 01	B4 66 04 81 6C 0E 07 74
19	01 07 94 04 91 19 04 01	6E 7E 62 21 A4 F3 4E 87
20	19 07 92 10 98 1A 01 01	AA 85 E7 46 43 23 31 99
21	10 07 91 19 98 19 08 01	2E 5A 19 DB 4D 19 62 D6
2[illegible]	10 07 91 19 98 1A 08 01	23 A8 66 A8 09 D3 08 94
23	10 07 92 10 98 19 01 01	D8 12 D9 61 F0 17 D3 20
24	10 07 91 15 98 19 01 0B	05 56 05 81 6E 58 60 8F
25	10 04 80 15 98 19 01 01	AB D8 8E 8B 1B 77 16 F1
26	10 04 80 15 98 19 01 02	53 7A C9 5B E6 9D A1 E1
27	10 04 80 15 98 19 01 08	AE D0 F6 AE 3C 25 CD D8
28	10 02 91 15 98 10 01 04	B3 E3 5A 5E E5 3E 7B 8D
29	10 02 91 15 98 19 01 04	61 C7 9C 71 92 1A 2E F8

ROUND	KEY	CT/RESULT
30	10 02 91 15 98 10 02 01	E2 F5 72 8F 09 95 01 3C
31	10 02 91 16 98 10 01 01	1A EA C3 9A 61 F0 A4 64

Table 4

Values To Be Used for the Substitution Table Known Answer Test

	KEY	PT/TEXT/IV (depending on mode)	CT/RESULT
0	7C A1 10 45 4A 1A 6E 57	01 A1 D6 D0 39 77 67 42	69 0F 5B 0D 9A 26 93 9B
1	01 31 D9 61 9D C1 37 6E	5C D5 4C A8 3D EF 57 DA	7A 38 9D 10 35 4B D2 71
2	07 A1 13 3E 4A 0B 26 86	02 48 D4 38 06 F6 71 72	86 8E BB 51 CA B4 59 9A
[illegible]	38 49 67 4C 26 02 31 9E	51 45 4B 58 2D DF 44 0A	71 78 87 6E 01 F1 9B 2A
[illegible]	04 B9 15 BA 43 FE B5 B6	42 FD 44 30 59 57 7F A2	AF 37 FB 42 1F 8C 40 95
5	01 13 B9 70 FD 34 F2 CE	05 9B 5E 08 51 CF 14 3A	86 A5 60 F1 0E C6 D8 5B
6	01 70 F1 75 46 8F B5 E6	07 56 D8 E0 77 47 61 D2	0C D3 DA 02 00 21 DC 09
7	43 29 7F AD 38 E3 73 FE	76 25 14 B8 29 BF 48 6A	EA 67 6B 2C B7 DB 2B 7A
8	07 A7 13 70 45 DA 2A 16	3B DD 11 90 49 37 28 02	DF D6 4A 81 5C AF 1A 0F
[illegible]	04 68 91 04 C2 FD 3B 2F	26 95 5F 68 35 AF 60 9A	5C 51 3C 9C 48 86 C0 88
10	37 D0 6B B5 16 CB 75 46	16 4D 5E 40 4F 27 52 32	0A 2A EE AE 3F F4 AB 77
11	1F 08 26 0D 1A C2 46 5E	6B 05 6E 18 75 9F 5C CA	EF 1B F0 3E 5D FA 57 5A
12	58 40 23 64 1A BA 61 76	00 4B D6 EF 09 17 60 62	88 BF 0D B6 D7 0D EE 56
13	02 58 16 16 46 29 B0 07	48 0D 39 00 6E E7 62 F2	A1 F9 91 55 41 02 0B 56
13	49 79 3E BC 79 B3 25 8F	43 75 40 C8 69 8F 3C FA	6F BF 1C AF CF FD 05 56
15	4F B0 5E 15 15 AB 73 A7	07 2D 43 A0 77 07 52 92	2F 22 E4 9B AB 7C A1 AC
16	49 E9 5D 6D 4C A2 29 BF	02 FE 55 77 81 17 F1 2A	5A 6B 61 2C C2 6C CE 4A
17	01 83 10 DC 40 9B 26 D6	1D 9D 5C 50 18 F7 28 C2	5F 4C 03 8E D1 2B 2E 41
18	1C 58 7F 1C 13 92 4F EF	30 55 32 28 6D 6F 29 5A	63 FA C0 D0 34 D9 F7 93

Table 5

Resulting Ciphertext from the Variable Plaintext Known Answer Test for Skipjack

(NOTE: KEY = 00 00 00 00 00 00 00 00 00 00)

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
00	80 00 00 00 00 00 00 00	9A 90 BC 0B 75 C7 37 03
01	40 00 00 00 00 00 00 00	CC 68 43 59 8C 73 2B BE
02	20 00 00 00 00 00 00 00	13 72 95 35 09 B3 C1 4C
03	10 00 00 00 00 00 00 00	70 AA AA 84 18 E4 89 30
04	08 00 00 00 00 00 00 00	E4 B0 B4 A1 39 E8 54 6E
05	04 00 00 00 00 00 00 00	70 18 F7 13 66 14 6E AF
06	02 00 00 00 00 00 00 00	B3 8F 3D 7E 4F 2D 25 3D
07	01 00 00 00 00 00 00 00	D6 4B A2 06 51 13 D9 1E
08	00 80 00 00 00 00 00 00	F9 5B 92 2F 14 27 A9 F2
09	00 40 00 00 00 00 00 00	6B 64 2F DE 40 85 85 86
10	00 20 00 00 00 00 00 00	6C F5 2D 5E 61 69 52 17
11	00 10 00 00 00 00 00 00	BC 0F 6B CA 62 E1 39 A6
12	00 08 00 00 00 00 00 00	6A D5 03 DC 2A B0 BF E2
13	00 04 00 00 00 00 00 00	AF AD D7 CA B6 72 35 16
14	00 02 00 00 00 00 00 00	00 42 1B 89 5A F5 C0 0A
15	00 01 00 00 00 00 00 00	CA D0 45 6C F8 6C D5 98
16	00 00 80 00 00 00 00 00	16 F4 1C 8F 8A 6A 5B 79
17	00 00 40 00 00 00 00 00	4C E7 71 C7 51 BA 27 60
18	00 00 20 00 00 00 00 00	72 C9 02 E5 8C E5 5B 87
19	00 00 10 00 00 00 00 00	6D 37 8C 66 64 D0 01 10
20	00 00 08 00 00 00 00 00	AC 27 B8 5B 0A 75 E8 BA
21	00 00 04 00 00 00 00 00	54 DF 3A 75 5B 00 63 D2
22	00 00 02 00 00 00 00 00	31 4F 4D 28 6D B4 90 58
23	00 00 01 00 00 00 00 00	88 AE 06 66 B2 A0 78 46

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
24	00 00 00 80 00 00 00 00	D8 60 A8 D9 A0 2C BC E8
25	00 00 00 40 00 00 00 00	37 CE 5E EA 53 13 53 5D
26	00 00 00 20 00 00 00 00	73 3A F9 2D A1 C1 80 26
27	00 00 00 10 00 00 00 00	34 1C 23 5F 6E 32 98 1D
28	00 00 00 08 00 00 00 00	C6 A6 56 14 47 D9 E0 96
29	00 00 00 04 00 00 00 00	C5 50 66 A8 D8 39 E5 FA
30	00 00 00 02 00 00 00 00	65 86 4B 48 79 11 A1 0C
31	00 00 00 01 00 00 00 00	87 29 07 E2 D3 36 33 2A
32	00 00 00 00 80 00 00 00	AF 03 76 88 E7 A5 24 9C
33	00 00 00 00 40 00 00 00	C1 FC D1 B4 DC C2 AC BB
34	00 00 00 00 20 00 00 00	40 48 48 80 2D 69 3D DA
35	00 00 00 00 10 00 00 00	B2 DC CE E3 3B 15 6D B6
36	00 00 00 00 08 00 00 00	E6 20 F4 2A 7F A9 01 0B
37	00 00 00 00 04 00 00 00	7C F0 67 F3 BD 3E C3 53
38	00 00 00 00 02 00 00 00	06 37 78 1F 1A 34 72 81
39	00 00 00 00 01 00 00 00	47 41 F1 46 4B 71 70 8E
40	00 00 00 00 00 80 00 00	ED AD 33 F4 56 F5 14 DF
41	00 00 00 00 00 40 00 00	ED 81 27 48 B7 F5 23 E9
42	00 00 00 00 00 20 00 00	83 8C 9C C3 83 D4 62 97
43	00 00 00 00 00 10 00 00	FB 2B C0 FC C9 2F 9B 24
44	00 00 00 00 00 08 00 00	E5 9A A1 12 2A 65 44 32
45	00 00 00 00 00 04 00 00	D4 C8 EF 7E 06 43 12 53
46	00 00 00 00 00 02 00 00	32 ED 63 28 14 C2 A8 56
47	00 00 00 00 00 01 00 00	5D C2 9F 7D E9 6E E5 2C
48	00 00 00 00 00 00 80 00	68 A0 7C 7E 8E AD D5 61
49	00 00 00 00 00 00 40 00	B2 70 68 F2 D6 B3 37 E2
50	00 00 00 00 00 00 20 00	1A F5 1E 9C 29 BF DC 7B
51	00 00 00 00 00 00 10 00	92 1D BD 9B 1C 6B EA EB
52	00 00 00 00 00 00 08 00	5B 6A 60 22 35 94 35 D2

ROUND	PLAINTEXT or IV (depending on mode)	CIPHERTEXT
53	00 00 00 00 00 00 04 00	D7 74 C6 23 74 B2 3B 09
54	00 00 00 00 00 00 02 00	FD 9F 05 27 59 4C E3 7B
55	00 00 00 00 00 00 01 00	67 86 01 C8 B3 64 A7 94
56	00 00 00 00 00 00 00 80	D5 18 22 8D 5B 0B E3 D7
57	00 00 00 00 00 00 00 40	A4 5F EE 6B DD 1F 73 4A
58	00 00 00 00 00 00 00 20	D1 BA 95 51 DF 7C D5 68
59	00 00 00 00 00 00 00 10	AE A3 3D 09 DC 9D 13 10
60	00 00 00 00 00 00 00 08	96 B4 91 C1 FE 44 3E 9A
61	00 00 00 00 00 00 00 04	D0 E0 14 CF EE 94 58 9D
62	00 00 00 00 00 00 00 02	0B 9E 44 B5 37 AF 28 79
63	00 00 00 00 00 00 00 01	22 F4 28 E3 EC 49 1E 60

Table 6

Resulting Ciphertext from the Variable Key Known Answer Test for Skipjack

((NOTE: Plaintext/text = 00 00 00 00 00 00 00 00 and, where applicable, IV = 00 00 00 00 00 00 00 00)

ROUND	KEY	CIPHERTEXT
0	80 00 00 00 00 00 00 00 00 00	7A 00 E4 94 41 46 1F 5A
1	40 00 00 00 00 00 00 00 00 00	A1 4F F8 BC D1 BC 9E F9
2	20 00 00 00 00 00 00 00 00 00	D7 E8 10 38 5A 42 AA EA
3	10 00 00 00 00 00 00 00 00 00	28 FE 2C 33 32 AA BD 35
4	08 00 00 00 00 00 00 00 00 00	3F C0 F0 5E E6 CE 78 8A
5	04 00 00 00 00 00 00 00 00 00	44 3D D0 CB 75 26 F7 4B
6	02 00 00 00 00 00 00 00 00 00	AD 81 9E 67 7C F9 03 05
7	01 00 00 00 00 00 00 00 00 00	98 91 75 5E 5E BA 5B 1D
8	00 80 00 00 00 00 00 00 00 00	0E 64 B4 94 63 3B F2 CB
9	00 40 00 00 00 00 00 00 00 00	63 38 1A 08 A4 7F C4 8D
10	00 20 00 00 00 00 00 00 00 00	F4 10 8B 09 9B 04 70 40
11	00 10 00 00 00 00 00 00 00 00	74 02 16 61 4E D0 E2 5B
12	00 08 00 00 00 00 00 00 00 00	80 00 91 7B 2E 16 B9 2A
13	00 04 00 00 00 00 00 00 00 00	A9 76 9B 62 B3 A0 BE 4E
14	00 02 00 00 00 00 00 00 00 00	42 FD B8 72 EA 31 41 21
15	00 01 00 00 00 00 00 00 00 00	1D 67 2B A0 15 6A B3 9D
16	00 00 80 00 00 00 00 00 00 00	F4 44 41 D7 C7 77 F0 57
17	00 00 40 00 00 00 00 00 00 00	EA 48 7D DC 36 0D 15 94
18	00 00 20 00 00 00 00 00 00 00	32 4B 0E 78 5F F2 B9 08
19	00 00 10 00 00 00 00 00 00 00	1A F5 9E C2 B9 D6 4C 4F
20	00 00 08 00 00 00 00 00 00 00	81 9B 7E 10 2E 76 A0 EE
21	00 00 04 00 00 00 00 00 00 00	0B 0B FE 0D 4A 37 AA 9E
22	00 00 02 00 00 00 00 00 00 00	12 B4 3E 37 60 D3 0D A6
23	00 00 01 00 00 00 00 00 00 00	31 77 25 6C 46 15 41 EE

ROUND	KEY	CIPHERTEXT
24	00 00 00 80 00 00 00 00 00 00	36 00 EB 92 83 6C A0 26
25	00 00 00 40 00 00 00 00 00 00	75 A4 35 AD 22 EC F7 93
26	00 00 00 20 00 00 00 00 00 00	71 90 AA 99 13 C1 F9 EC
27	00 00 00 10 00 00 00 00 00 00	AB A7 18 B1 85 A1 1D D0
28	00 00 00 08 00 00 00 00 00 00	40 F6 7A BF CC 3B 87 3C
29	00 00 00 04 00 00 00 00 00 00	38 A0 A5 8F B0 97 28 F2
30	00 00 00 02 00 00 00 00 00 00	CA 70 2E 49 BF 6F A6 45
31	00 00 00 01 00 00 00 00 00 00	45 5D 93 F0 39 EA 08 60
32	00 00 00 00 80 00 00 00 00 00	53 47 64 3F E8 03 88 3F
33	00 00 00 00 40 00 00 00 00 00	F4 0F F1 DC BA 2B C1 E5
34	00 00 00 00 20 00 00 00 00 00	57 4A 48 48 36 9D 41 2E
35	00 00 00 00 10 00 00 00 00 00	B2 BE 93 6E 36 67 06 36
36	00 00 00 00 08 00 00 00 00 00	5C 88 51 7D 27 42 E6 19
37	00 00 00 00 04 00 00 00 00 00	99 3C 89 D0 9A 2F E5 56
38	00 00 00 00 02 00 00 00 00 00	1A 3F 72 DA 69 4C 9F C7
39	00 00 00 00 01 00 00 00 00 00	96 59 D5 22 8F 4C B1 51
40	00 00 00 00 00 80 00 00 00 00	7C 13 F4 9E 75 0F 5C 30
41	00 00 00 00 00 40 00 00 00 00	35 00 BD 40 7B CD 01 F6
42	00 00 00 00 00 20 00 00 00 00	85 C5 8E 3C 49 44 20 28
43	00 00 00 00 00 10 00 00 00 00	84 13 84 0A 2D 48 AB EA
44	00 00 00 00 00 08 00 00 00 00	83 28 50 E6 E5 C4 AE 5A
45	00 00 00 00 00 04 00 00 00 00	29 E9 7F 0D 9F 0F DC 5F
46	00 00 00 00 00 02 00 00 00 00	2C 45 23 04 37 FF 2E 04
47	00 00 00 00 00 01 00 00 00 00	10 C4 09 FB 87 2A 98 4F
48	00 00 00 00 00 00 80 00 00 00	14 69 3B 30 C3 AF 74 70
49	00 00 00 00 00 00 40 00 00 00	91 3A 90 50 D5 85 BA B9
50	00 00 00 00 00 00 20 00 00 00	5B FB 0F 83 AB 0C 6E EA
51	00 00 00 00 00 00 10 00 00 00	6C 0C A7 28 4D 83 6A AE

ROUND	KEY	CIPHERTEXT
52	00 00 00 00 00 00 08 00 00 00	AC 57 27 D6 12 E1 85 E8
53	00 00 00 00 00 00 04 00 00 00	38 D7 D5 96 A3 D2 9D 90
54	00 00 00 00 00 00 02 00 00 00	78 BA DA D3 BC 43 6C A2
55	00 00 00 00 00 00 01 00 00 00	E4 05 77 87 41 B0 4B A0
56	00 00 00 00 00 00 00 80 00 00	72 FF E4 3D EA 02 AF A5
57	00 00 00 00 00 00 00 40 00 00	52 E9 31 DF 24 8C E4 C7
58	00 00 00 00 00 00 00 20 00 00	4B B1 65 FD B3 BF F6 5C
59	00 00 00 00 00 00 00 10 00 00	7C FA FA 68 61 D7 B4 7D
60	00 00 00 00 00 00 00 08 00 00	48 D1 75 52 31 F8 7A 2A
61	00 00 00 00 00 00 00 04 00 00	41 32 07 DA 1C 9B 6A B5
62	00 00 00 00 00 00 00 02 00 00	63 F8 18 E9 38 2A 27 78
63	00 00 00 00 00 00 00 01 00 00	ED AF 2B 85 FC 30 EB 09
64	00 00 00 00 00 00 00 00 80 00	11 FC 59 93 82 07 63 F7
65	00 00 00 00 00 00 00 00 40 00	E5 39 C3 96 99 15 09 2F
66	00 00 00 00 00 00 00 00 20 00	50 6F 6A 1E 83 4A D8 F7
67	00 00 00 00 00 00 00 00 10 00	8B 15 BA 30 47 FA 31 95
68	00 00 00 00 00 00 00 00 08 00	13 0B E1 5C 39 3E 4B 7A
69	00 00 00 00 00 00 00 00 04 00	88 95 EC 31 04 CA 10 41
70	00 00 00 00 00 00 00 00 02 00	E4 40 AC DF 4B 64 C9 C9
71	00 00 00 00 00 00 00 00 01 00	C2 32 80 EB E0 93 F0 02
72	00 00 00 00 00 00 00 00 00 80	52 64 A6 57 41 FE 78 E3
73	00 00 00 00 00 00 00 00 00 40	80 89 2E 76 85 47 CE 61
74	00 00 00 00 00 00 00 00 00 20	09 11 41 2D 72 09 34 75
75	00 00 00 00 00 00 00 00 00 10	9F 21 AA 76 47 83 E6 49
76	00 00 00 00 00 00 00 00 00 08	4C A9 FA BE AD 2C 02 C6
77	00 00 00 00 00 00 00 00 00 04	59 CE 10 97 3A 7B 1F D5
78	00 00 00 00 00 00 00 00 00 02	68 3B 29 34 E0 CC BE AA
79	00 00 00 00 00 00 00 00 00 01	74 D0 E7 C2 E3 B4 50 A8

REFERENCES

1. Data Encryption Standard (DES), FIPS PUB 46-2, December 30, 1993.

2. Escrowed Encryption Standard (EES), FIPS PUB 185, February 9, 1994.

3. Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard, NBS Special Publication 500-20, November, 1977.

4. DES Modes of Operation, FIPS PUB 81, December 2, 1980.

5. Security Requirements for Cryptographic Modules, FIPS PUB 140-1, January 11, 1994.

6. Guidelines for Implementing and Using the NBS Data Encryption Standard, FIPS PUB 74, April 1, 1981.

☆U.S. GOVERNMENT PRINTING OFFICE: 1998-433-235/82678

NIST

[illegible]

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and nongovernment). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards
and Technology
Gaithersburg, MD 20899–0001

Official Business
Penalty for Private Use $300