U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY
WASHINGTON. D.C. 20234

FIPS PUBLICATION CHANGE NOTICE

CHANGE NO. 1
DATE OF CHANGE
1981 NOVEMBER 20
FIPS PUB. NO.
81

TITLE OF PUBLICATION

FIPS PUB 81, DES MODES OF OPERATION.

This office has a record of your interest in receiving changes to the above FIPS PUB. The change(s) indicated below have been provided by the Maintenance Agency for this publication and will be included in the next published revision to this FIPS PUB. Questions or requests for additional information should be addressed to the Maintenance Agency:
Department Of Commerce
National Bureau Of Standards
Institute for Computer Sciences and Technology
Washington, D.C. 20234

CHANGE ITEM(S)

PAGE	SECTION	PARA.	LINE(S)	ORIGINAL	CHANGE	
8	4		3	All	An acceptable	Replace with (a) below
16	5			All	The 7-bit CFB	Replace with (a) below
16	6			2,4	7-bit	8-bit
16	6			5	14,21,28,35,42, 49, and 56	16,24,32,40,48, 56, and 64
16	6			6	and 7-bit	and 8, 8-bit
16	7			1	7 and 56-bit	8 and 64-bit
20				2,3	7-BIT	8-BIT
20				13	7-BIT	8-BIT
21				2,3	56-BIT	64-BIT
21				13	56-BIT	64-BIT

a) An acceptable alternative for 7-bit CFB that uses an 8-bit feedback path while enciphering 7-bit data units is the 7-bit CFB(a) mode of operation. This alternative always inserts a "1" in bit position one of the 8-bit feedback path so that the feedback is of the form (1 ,C1 ,C2,C3,C4,C5,CS,C7). The cipher is represented as a 7-bit entity of the fom (C1,C2,C3,C4,C5,C6,C7).

An acceptable alternative for 8-bit CFB when enciphering 8-bit data units composed of a non-information bit followed by a 7-bit code (e.g., p,b7,b6,b5,b4,b3,b2,b1) is the 8-bit CFB(a) mode of operation. This alternative is similar to the 8-bit CFB except that a "1" bit is always inserted in bit position one of the 8-bit feedback path so that the feedback is of the form (1, C2, C3,C4,CE,C6,C7,C8). The cipher is represented as an 8-bit entity of the form (C1,C2,C3,C4,C5,C6,C7,C8) or (0,C2, C3,C4,C5,C6,C7,C8) or (1,C2,C3,C4,C5,C6,C7,C8) or (P,C2,C3,C4,C5,C6,C7,C8) where P is a cipher parity bit.

(b) The 7-bit CFB(a) mode is defined in the standard in order to encipher and decipher 7-bit data units and still use an 8-bit feedback path.

Most computer and communication systems are designed to efficiently handle full 8-bit bytes. When using 7-bit codes the eighth bit of the byte is often used as a parity bit so that the byte is of the form (p,b7,b6,b5,b4,b3,b2,b1). These systems often generate the parity bit during transmission and check its validity during reception. In such systems the parity bit on cipher text would be automatically modified during transmission. In this cases the encryption and decryption processes must operate independently of the parity bits and the 8-bit CFB(a) mode should be used.

NOTE: These changes are provided-to make the specification of the 7-bit CFB(a) mode consistent with that specified in a proposed American National Standard for the Modes of Operation of the Data Encryption Algorithm. The 8-bit CFB(a) mode and its extensions are defined in FIPS PUB 81 so that they may be used in many application standards.

FIPS PUB 81
FIPS PUB FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION
1980 December 2
DES MODES OF OPERATION
CATEGORY: ADP OPERATIONS
SUBCATEGORY: COMPUTER SECURITY

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, Secretarv
Jordan J. Baruch, Assistant Secretary for Productivity,
Technology and Innovation

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Foreword

The Federal Information Processing Standards Publication Series of the National Bureau of Standards is the official publication relating to standards adopted and promulgated under the provisions of Public Law 89-306 (Brooks Act) and under Part 6 of Title 15, Code of Federal Regulations. These legislative and executive mandates have given the Secretary of Commerce important responsibilities for improving the utilization and management of computers and automatic data processing in the Federal Government. To carry out the Secretary's responsibilities, the NBS, through its Institute for Computer Sciences and Technology, provides leadership, technical guidance and coordination of Government efforts in the development of guidelines and standards in these areas.

Comments concerning Federal Information Processing Standards Publications are welcomed and should be addressed to the Director, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, DC 20234.
James H. Burrows, Director Institute for Computer Sciences
and Technology

Abstract

The Federal Data Encryption Standard (DES) (FIPS 46) specifies a cryptographic algorithm to be used for the Cryptographic protection of sensitive, but unclassified, computer data. This FIPS defines four modes of operation for the DES which may be used in a wide variety of applications. The modes specify how data will be encrypted (cryptographicaliy protected) and decrypted (returned to original form). The modes included in this standard are the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CF3) mode, and the Output Feedback (OFB) mode.

KEY WORDS: Computer security; cryptography; data security; DES; encryption; Federal Information Processing Standards; modes of operation.

Nat.Bur.Stand. (U.S.), Fed.Info.Process.Stand.Publ.(FIPS PUB)
81, 26 pages.(1981)CODEN: FIPPAT

For sale by the National Technical Information Service, U.s Department of Commerce, Springfield, VA 22161.

A list of currently approved FIPS may be obtained from the Standards Administration Office, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, DC 20234.

7. Applicability. This standard shall be used by Federal departments and agencies when procuring equipment or services which implement the Data Encryption Standard and which are intended for use in the cryptographic protection of sensitive, but unclassified, computer data. This standard may be used by anyone desiring to implement and use the Data Encryption Standard. The selection of one of the specified modes of operation will depend on the particular application being considered.

8. Specifications. Federal Information Processing Standard (FIPS 81) DES Modes of Operation (affixed).

9. Qualifications. The DES modes of operation described in this standard are based upon information provided by many sources within the Federal Government and private industry.
These modes are presently being implemented in cryptographic equipment containing DES devices. However, a standard of this nature must, of necessity, remain flexible enough to adapt to advancements and innovations in science and technology. As such, this standard should not be construed as being either exhaustive or static. It will be reviewed every five years in order to incorporate new implementations whose technical or economic merit justify the issuance of a revised standard. FIPS 46 requires implementation of the DES algorithm in electronic devices when used by Federal departments and agencies. The DES, itself, must therefore be in hardware or firmware for Federal applications. However, the modes of operation specified in this standard may be implemented in software, hardware, or firmware.

10. Export Control. Cryptographic devices and technical data regarding them are subject to Federal Government export controls as specified in Title 22, Code of Federal Regulations, Parts 121 through 128. Cryptographic devices implementing this standard and technical data regarding them must comply with these Federal regulations.

11. Patents. Cryptographic equipment implementing this standard may be covered by U.S. and foreign patents.

12. Implementation Schedule. This standard becomes effective on June 2, 1981.

13. Waivers. Heads of agencies may request that the requirements of this standard be waived in instances where it can be clearly demonstrated that there are appreciable performance or cost advantages to be gained and when the overall interests of the Federal Government are best served by granting the requested waiver. Such waiver requests will be reviewed by and are subject to the approval of the Secretary of Commerce. The waiver request must specify anticipated performance and cost advantages in the justification for the waiver.

Forty-five days should be allowed for review and response by the Secretary of Commerce. Waiver requests shall be submitted to the Secretary of Commerce, Washington, DC 20230, and labeled as a Request for a Waiver to this Federal Information Processing Standard. No agency shall take any action to deviate from this standard prior to the receipt of a waiver approval from the Secretary of Commerce. No agency shall implement or procure equipment using a DES mode of operation not conforming to this standard unless a waiver has been approved.

14. Where to Obtain Copies. Copies of this publication are for sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. When ordering, refer to Federal Information Processing Standards Publication 81 (FIPS PUB 81), and title. When microfiche is desired, this should be specified. Payment may be made by check, money order, or deposit account.

Federal Information Processing Standards Publication 81
1980 December 2
ANNOUNCING THE STANDARD FOR DES MODES OF OPERATION

Federal Information Processing Standards Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949, as amended, Public Law 89-306 (79 Stat. 1127), Executive Order 11717 (38 FR 12315, dated May 11, 1973), and Part 6 of Title 15 Code of Federal Regulations (CFR).

1. Name of Standard. DES Modes of Operation.

2. Category of Standard. ADP Operations, computer security.

3. Explanation. The Federal Data Encryption Standard (DES) (FIPS 46) specifies a crypto-graphic algorithm to be used for the cryptographic protection of sensitive, but unclassified, computer data. This FIPS defines four modes of operation for the DES which may be used in a wide variety of applications. The modes specify how data will be encrypted (cryptographically protected) and decrypted (returned to original form). The modes included in this standard are the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CFB) mode, and the Output Feedback (OFB) mode.

The body of this standard provides specifications of the recommended modes of operation but does not specify the necessary and sufficient conditions for their secure implementation in a particular application. This standard specifies the numbering of data bits, how the bits are encrypted and decrypted, and the data paths and the data processing necessary for encrypting and decrypting data or messages. This standard is based on (and references) the DES and provides the next level of detail necessary for providing compatibility among DES equipment. This standard anticipates the development of a set of application standards which reference it such as communication security standards, data storage standards, password protection standards and key management standards. Cryptographic system designers or security application designers must select one or more of the possible modes of operation for implementing and using the DES in a cryptographic system or security application. The Appendices to this standard provide tutorial information on the modes of operation and examples for validating their correct implementation. The Appendices are guidelines and are not mandatory requirements of this standard.

4. Approving Authority. Secretary of Commerce.

5. Maintenance Agency. U.S. Department of Commerce, National Bureau of Standards, Institute for Computer Sciences and Technology.

6. Related Documents.

FIPS PUB 46, "Data Encryption Standard," January 15, 1977.

(Proposed) Federal Standard 1026, "Telecommunications: Interoperability Requirements for

(Proposed) Federal Standard 1027, "Telecommunications: Security Requirements for Use of the Data Encryption Standard," August 5, 1980, draft.

Federal Information
Processing Standards Publication 81
1980 December 2 :

Specifications for DES MODES OF OPERATION

CONTENTS

I. Introduction. Binary data may be cryptographically protected (encrypted) using devices implementing the algorithm specified in the Data Encryption Standard (DES) (FIPS PUB 46) in conjunction with a cryptographic key. The cryptographic key controls the encryption process and the identical key must also be used in the decryption process to obtain the original data. Since the DES is publicly defined, cryptographic security depends on the secrecy of the cryptographic key.

The binary format of a cryptographic key is: (B1,B2,...,B7,P1,B8,...B14,P2,B15,...,B49,P7,B50,...,B56,P8)

where (B1,B2,...,B56) are the independent bits of a DES key and (PI,P2,...,P8) are reserved for parity bits computed on the preceding seven independent bits and set so that the parity of the octet is odd, i.e., there is an odd number of "1" bits in the octet.

The hexadecimal format of a cryptographic key is: (H1H2 H3H4 ... H15H16)

where (H1,H2,... ,H16) are hexadecimal characters from the set (O,1,...,9,A,B,C,D,E,F). The embedded blanks in the format are optional and lower case letters may be used in place of the upper case letters. This standard assumes that a cryptographic key has been entered into a DES device prior to encryption or decryption.

1.1 Definitions, Abbreviations, and Conventions. The following definitions, abbreviations and conventions shall be used throughout this standard:

BIT: A binary digit denoted as a "0" or a

BINARY VECTOR: A sequence of bits.

BLOCK: A binary vector consisting of sixty-four bits numbered from the left as 1, 2, . 64 and denoted as (B1,B2,...,364).

CBC: Ciplier Block Chaining.

CFB: Cipher Feedback.

CIPHER TEXT: Encrypted data.

CRYPTOGRAPHIC KEY: A 64-bit parameter consisting of 56 independent bits and 8 parity bits used in a DES device to control the encrypt and decrypt operations.
(Synonyms: KEY, KEY VARIABLE).

DATA UNIT: A binary vector of K bits that is encrypted as an entity and denoted as (D1,D2,...,DK) where K = 1,2,...,64 and where D1,D2,...,DK represent bits.

DECRYPTION: The process of changing cipher text into plain text.
Verb: DECRYPT.
(Synonym: DECIPHER).

DECRYPT STATE: The state of a DES device executing the deciphering operation specified in FIPS PUB 46.

DES: Data Encryption Standard; specified in FIPS PUB 46.

DES DEVICE: The electronic component used to implement the DES algorithm, typically an integrated circuit chip or a micro-computer with the DES algorithm specified in a read-only memory program.

DES INPUT BLOCK: A block that is entered into the DES device for either encryption or decryption. The input block shall be designated (I1,I2,...,I64) where I1,12,...,164 represent bits.

DES OUTPUT BLOCK: A block that is the Final result of an encryption or decryption opertion of a DES device. The output block shall be designated (01,02,... ,064) where 01,02,... ,064 represent bits.

ECB: Electronic Codebook.

ENCRYPTION: The process of changing plain text into cipher text.
Verb: ENCRYPT.
(Synonym: ENCIPHER).

ENCRYPT STATE: The state of a DES device executing the enciphering operation specified in FIPS PUB 46.

EXCLUSIVE-OR OPERATION: The bit-by-bit modulo-2 addition of two binary vectors of equal length. This operation is represented by a "≈" in this standard.

INITIALIZATION VECTOR (IV): A binary vector used in the initial input block in the CFB and OFB modes and as the randomizing block that is exclusive-ORed with the first data block in the CBC mode.

LEAST SIGNIFICANT BIT(S): The right-most bit(s) of a binary vector. (Synonym: Low order bit(s)).

MESSAGE (MSG): A logical data entity consisting of a sequence of data units (e.g., bits, octets, characters, fixed length numbers) that is encrypted as an entity.

MOST SIGNIFICANT BIT(S): The lef t-most bit(s) of a binary vector.
(Synonym: High order bit(s)).

OCTET: A group of eight binary digits numbered from left to right: B1,B2,...,B8.

OFB: Output Feedback.

PLAIN TEXT: Unencrypted data.

2. Electronic Codebook (ECB) Mode. The Electronic Codebook (ECB) mode is defined as follows (Figure 1). In ECB encryption, a plain text data block (D1,D2,...,D64) is used directly as the DES input block (11,12,... ,164). The input block is processed through a DES device in the encrypt state. The resultant output block (01,02,...,064) is used directly as cipher text (C1,C2,...,C64) or may be used in subsequent ADP applications.

In ECB decryption, a cipher text block (C1,C2,...,C64) is used directly as the DES input block (I1,I2,...,164). The input block is then processed through a DES device in the decrypt state. The resultant output block (O1,O2,...,064) is the plain text (D1,D2,. ..,D64) or may be used in subsequent ADP applications. The ECB decryption process is the same as the ECB encryption process except that the decrypt state of the DES device is used rather than the encrypt state.

3. Cipher Block Chaining (CBC) Mode. The Cipher Block Chaining (CBC) mode is defined as follows (Figure 2). A message to be encrypted is divided into blocks. In CBC encryption, the first DES input block is formed by exclusive-ORing the first block of a message with a 64-bit initialization vector (IV), i.e., (I1,I2,...,I64) = (IV1≈D1,IV2≈D2,...,IV64≈D64). The input block is processed through a DES device in the encrypt state, and the resulting output block is used as the cipher text, i.e., (C1,C2,... ,C64) = (01,02,.. ,064). This first cipher text block is then exclusive-ORed with the second plain text data block to produce the second DES input block, i.e., (I1,I2,...,I64) = (C1≈D1,C2≈D2,...,C64≈64). Note that I and D now refer to the second block. The second input block is processed through the DES device in the encrypt state to produce the second cipher text block. This encryption process continues to "chain" successive cipher and plain text blocks together until the last plain text 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

FIGURE 1: ELECTRONIC CODEBOOK (ECB) MODE

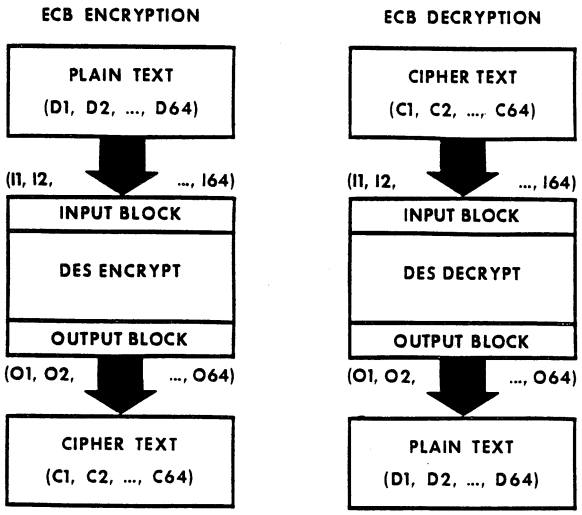
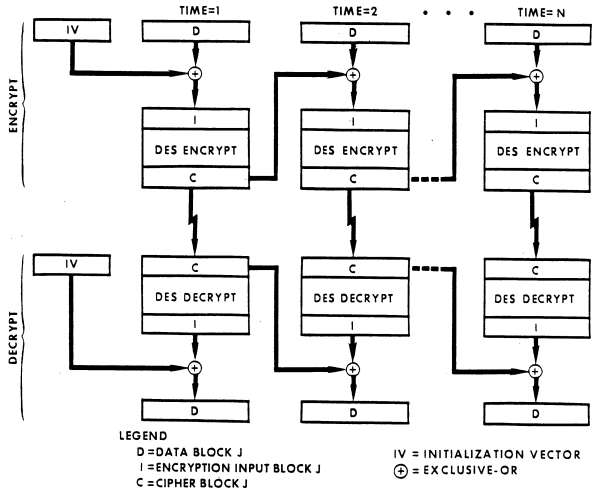


FIGURE 2: CIPHER BLOCK CHAINING (CBC) MODE



encrypted in a manner specified for the application. One such method is described in Appendix C of this standard.

In CBC decryption, the first cipher text block of an encrypted message is used as the input block and is processed through a DES device in the decrypt state, i.e., (I1,12,...,164) = (C1,C2,...,C64). The resulting output block, which equals the original input block to the DES during encryption, is exclusive-ORed with the IV (must be same as that used during encryption) to produce the first plain text block, i.e., (D1,D2,...,D64)= (O1≈IV1,O2≈IV2,...,O64≈IV64). The second cipher text block is then used as the input block and is processed through the DES in the decrypt state and the resulting output block is exclusive-ORed with the first cipher text block to produce the second plain text data block, i.e., (D1,D2,...,D64) = (O1≈C1,02≈C2,...,064≈C64). Note that again the D and O refer to the second block. The CBC decryption process continues in this manner until the last complete cipher text block has been decrypted. Cipher text representing a partial data block must be decrypted in a manner as specified for the application.

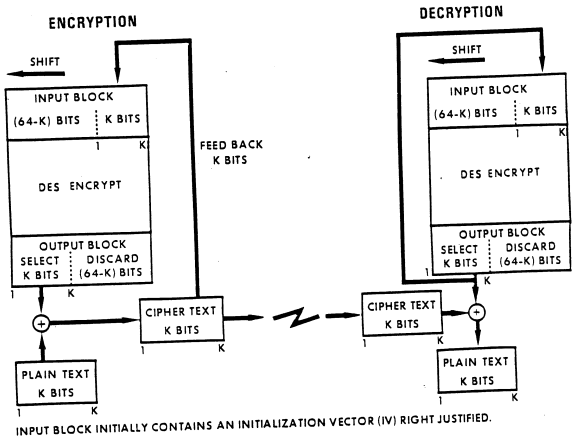
4. Cipher Feedback (CFB) Node. The Cipher Feedback (CFB) mode is defined as follows (Figure 3). A message to be encrypted is divided into data units each containing K bits (K = 1,2,... ,64). In both the CFB encrypt and decrypt operations, an initialization vector (IV) of length L is used. The IV is placed in the least significant bits of the DES input block with the unused bits set to "0's," i.e., (I1,12,...,164) - (0,0,...,0,IV1,IV2,IVL). This input block is processed through the DES device in the encrypt state to produce an output block. During encryption, cipher text is produced by exclusive-ORing a K-bit plain text data unit with the most significant K bits of the output block, i.e., (C1,C2,...,CK) - (D1≈01,D2≈02,... ,DK≈0K). Similarly, during decryption, plain text is produced by exclusive-ORing a K-bit unit of cipher text with the most significant K bits of the output block, i.e., (DI,D2,...,DK) - (CI≈1C2≈02,. .. ,CK≈0K). In both cases the unused bits of the DES output block are discarded. In both cases the next input block is created by discarding the most signif icant K bits of the previous input block, shifting the remaining bats K positions to the left and then inserting the K bits of cipher text just produced in the encryption operation or just used in the decrypt operation into the least significant bit positions, i.e., (I1,12,...,164) = (I[K+1l,I[K+21,...,164,CI,C2,...,CK). This input block is then processed through the DES device in the encrypt state to produce the next output block. This process continues until the entire plain text message has been encrypted or until the entire cipher text message has been decrypted.

The CFB mode may operate on data units of length l through 64 inclusive. K-bit CFB is defined to be the CFB mode operating on data units of length K for K - 1,2,... ,64. For each operation of the DES device one K-bit unit of plain text produces one K-bit unit of cipher text or one K-bit unit of cipher text produces one K-bit unit of plain text.

An acceptable alternative for 8-bit CFB when enciphering 7-bit entities using an 8-bit feedback path is to insert a "1" bit in bit position one of the 8-bit feedback path, i.e., ("1",C1,C2,... ,C7). This results in a "1" always being placed in bit location 57 of the DES input block. This alternative is called the 7-bit CFB(a) mode of operation.

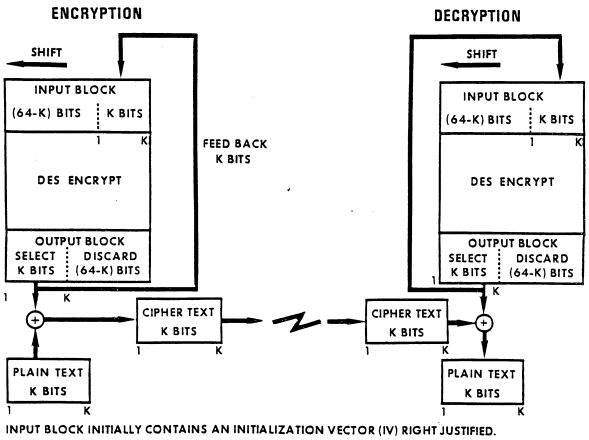
5. Output Feedback (OFB) Node. The Output Feedback (OFB) mode is defined as follows (Figure 4). A message to be encrypted is divided into data units each containing K bits (K = 1,2,...,64). In both the OFB encrypt and decrypt operations, an initialization vector (IV) of length L is used. The IV is placed in the least significant bits of the DES input block with the unused bits set to "O's," i.e., (I1,12,...,164) = (0,0,...,0,IV1,IV2,...,IVL). This input block is processed through the DES device in the encrypt state to produce an output block. During encryption, cipher text is produced by exclusive-ORing a K-bit plain text data unit with the most significant K bits of the output block, i.e., (C1,C2,...,CK) = (D1≈O1,D2≈O2,... ,DK≈OK). Similarly, during decryption, plain text is produced by exclusive-ORing a K-bit unit of cipher text with the most significant K bits of the output block, i.e., (D1,D2,...,DK) = (C1≈01,C2≈02,...,CK≈0K). In both cases the unused bits of the DES output block are discarded. In both cases 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 output just used into the least significant bit positions, i.e., (Il,12,...,164) = I[K+1], I[K+21, ..., I64, 01, 02,...,OK). This input block is then processed through the DES

FIGURE 3: K-BIT CIPHER FEEDBACK (CFB) MODE



INPUT BLOCK INITIALLY CONTAINS AN INITIALIZATION VECTOR (IV) RIGHT JUSTIFIED.

FIGURE 4: K-BIT OUTPUT FEEDBACK (OFB) MODE



INPUT BLOCK INITIALLY CONTAINS AN INITIALIZATION VECTOR (IV) RIGHT JUSTIFIED.

device in the encrypt state to produce the next output block. This process continues until the entire plain text message has been encrypted or until the entire cipher text message has been decrypted.

The OFB mode may operate on data units of length 1 through 64 inclusive. K-bit OFB is defined to be the 0FB mode operating on data units of length K for K = 1,2,...,64. For each operation of the DES device one K-bit unit of plain text produces one K-bit unit of cipher text or one K-bit unit of cipher text produces one K-bit unit of plain text.

APPENDIX A

GENERAL INFORMATION

The National Bureau of Standards issued Federal Information Processing Standards Publication 46 (FIPS PUB 46) in 1977. That standard specifies a cryptographic algorithm, commonly called the Data Encryption Standard (DES) algorithm, to be used within the Federal Government for the cryptographic protection of sensitive, but unclassified, computer data. The DES algorithm was developed by the International Business Machines Corporation (IBM) and submitted to the National Bureau of Standards during an NBS public solicitation for cryptographic algorithms to be used in a Federal Information Processing Standard. Several methods for incorporating this algorithm into a cryptographic system are possible.

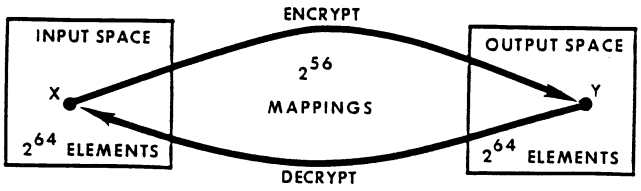
These methods, external to the DES algorithm, have come to be called the "modes of operation." Four modes, called the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC)
mode, the Cipher Feedback (CFB) mode, and the Output Feedback (OFB) mode, are specified in this standard. ECB is a direct application of the DES algorithm to encrypt and decrypt
data; CBC is an enhanced mode of ECB which chains together blocks of cipher text; CFB uses previously generated cipher text as input to the DES to generate pseudo-random outputs which are combined with the plain text to produce cipher text, thereby chaining together the resulting cipher text; OFB is identical to CFB except that the previous output of the DES is used as input in OFB while the previous cipher text is used as input in CFB. OFB does not chain the cipher text. The proposed FIPS specifies these four modes because they are capable of providing acceptable levels of protection for all anticipated unclassified Federal ADP encryption applications.

Unencrypted data is called plain text. Encryption (also called enciphering) is the process of transforming plain text into cipher text. Decryption (also called deciphering) is the inverse transformation. The encryption and decryption processes are performed according to a set of rules, called an algorithm, that is typically based on a parameter called a key. The key is usually the only parameter that must be provided to or by the users of a cryptographic system and must be kept secret. The period of time over which a particular key is used to encrypt or decrypt data is called its cryptoperiod.

Mathematically, the DES maps the set of all possible 64-bit vectors onto itself. See Figure A1. There are 2T64 (2 raised to the 64th power) elements in this set, including all binary numbers from O up to, but not including, 2f64. The DES cryptographic key allows a user to select any one of 2t56 possible invertible mappings, i.e., transformations that are one-to-one. Selecting a key selects one of the mappings. When using the DES in ECB mode and any particular key, each input is mapped onto a unique output in encryption and this output is mapped back onto the input in decryption. The DES is an iterative, block, product cipher system (i.e., encryption algorithm). A product cipher system mixes transposition and substitution operations in an alternating manner. Because the DES algorithm maps a 64-bit input block onto a 64-bit output block the DES is called a block cipher system. Iterative refers to the use of the output of an operation as the input for another iteration of the same procedure. The DES internally uses sixteen iterations of a pair of transposition and substitution operations to encrypt or decrypt an input block. A complete specification of the DES algorithm is found in FIPS PUB 46.

Two categories of methods for incorporating the DES in a cryptographic system are block methods and stream methods. In a block method, the DES input block is (or is a simple function of) the plain text to be encrypted and the DES output block is the cipher text. A stream method is based on generating a pseudo-random binary stream of bits, and then using the exclusive-OR binary operation to combine this pseudo-random sequence with the plain text to produce the cipher text. Since the exclusive-OR operator is its own binary inverse, the same pseudo-random binary stream is used for both the encryption of plain text, P, and the decryption of cipher text, C. If 0 is the pseudo-random binary stream, then $C = P \approx 0$ and inversely, $P = C \approx$.

FIGURE A1: DES MAPPINGS



ELECTRONIC CODEBOOK (ECB) MODE

The Electronic Codebook (ECB) mode is a basic, block, cryptographic method which transforms 64 bits of input to 64 bits of output u specified in FIPS PUB 46. The analogy to a codebook arises because the same plain text block always produces the same cipher text
block for a given cryptographic key. Thus a list (or codebook) of plain text blocks and corresponding cipher text blocks theoretically could be constructed for any given key. In electronic implementation the codebook entries are calculated each time for the plain text to be encrypted and, inversely, for the cipher text to be decrypted.

Since each bit of an ECB output block is a complex function of all 64 bits of the input block and all 56 independent (non-parity) bits of the cryptographic key, a single bit error in either a cipher text block or the non-parity key bits used for decryption will cause the decrypted plain text block to have an average error rate of fifty percent. However, an error in one ECB cipher text block will not affect the decryption of other blocks, i.e., there is no error extension between ECB blocks.

If block boundaries are lost between encryption and decryption (e.~., a bit slip), then synchronization between the encryption and decryption operations will be lost until correct block boundaries are reestablished. The results of all decryption operations will be incorrect until this occurs.

Since the ECB mode is a 64-bit block cipher, an ECB device must encrypt data in integral multiples of sixty-four bits. If a user has less than sixty-four bits to encrypt, then the least significant bits of the unused portion of the input data block must be padded, e.g., filled with random or pseudo-random bits, prior to ECB encryption. The corresponding decrypting device must then discard these padding bits after decryption of the cipher text block.

The same input block always produces the same output block under a fixed keg in ECB mode. If this is undesirable in a particular application, the CBC, CFB or OFB modes should be used. An example of the ECB mode is given in Table B1.

TABLE B1

AN EXAMPLE OF THE ELECTRONIC CODEBOOK (ECB) MODE The ECB mode in the

encrypt state has been selected.

Cryptographic Key = 0123456789abcdef

The plain text is the ASCII code for "Now is the time for all ." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...,b1).

TIME	PLAIN TEXT BLOCK	DES INPUT	DES OUTPUT	CIPHER TEXT BLOCK
1	4e6f772069732074	4e6f772069732074	3fa40e8a984d4315	3fa40e8a984d4815
2	68652074696d652o	68652074696d652o	6a271787ab8883f9	6a271787ab8883f9
3	666f7220616c6c20	666f7220616c6c20	893d51ec4b563b53	893d51ec4b563b53

The ECB mode in the decrypt state has been selected.

TIME	CIPHER TEXT BLOCK	DES INPUT	DES OUTPUT	PLAIN TEXT BLOCK
1	3fa40e8a984d4815	3fa40e8a984d4815	4e6f772o69732o74	4e6f772069732074
2	6a271787ab8883f9	6a271787ab8883f9	68652074696d6520	68652074696d6520
3	893d51ec4b563b53	893d51ec4b563b53	666f7220616c6c20	666f7220616c6c20

APPENDIX C

CIPHER BLOCK CHAINING (CBC) MODE

CBC is a block cipher system in which the first plain text data block is exclusive-ORed with a block of pseudo-random data prior to being processed through the DES. The resulting cipher text block is then exclusive-ORed with the next plain text data block to form the next input block to the DES, thus chaining together blocks of cipher text. The chaining of cipher text blocks provides an error extension characteristic which is valuable in protecting against fraudulent data alteration. A CBC authentication technique is described in Appendix F.

The CBC mode produces the same cipher text whenever the same plain text is encrypted using the same key and IV. Users who are concerned about this characteristic should incorporate a unique identifier (e.g., a one-up counter) at the beginning of each CBC message within a cryptographic period in order to insure unique cipher text. If the key and the IV are the same and no identifier precedes each message, messages that have the same beginning will have the same cipher text when encrypted in the CBC mode until the blocks that differ in the two messages are encrypted.

Since the CBC mode is a block method of encryption, it must operate on 64-bit data blocks. Partial data blocks (blocks of less than 64 bits) require special handling. One method of encrypting a final partial data block of a message is described below. Others may be defined for special applications.

The following method may be used for applications where the length of the cipher text can be greater than the length of the plain text. In this case the final partial data block of a message is padded in the least significant bits positions with "0"s, "1"s or pseudo- random bits. The decryptor will have to know when and to what extent padding has occurred This can be accomplished explicitly, e.g., using a padding indicator, or implicitly, e.g., using constant length transactions. The padding indicator will depend on the data being encrypted. If the data is pure binary, then the partial data block should be left justified in the input block and the unused bits of the block set to the complement of the last data bit, i.e., if the last data bit of the message is "0" then "1"s are used as padding bits and if the last data bit is "1" then "0"s are used. The input block is then encrypted. The resulting output block is the cipher text. The cipher text message must be marked as being padded so that the decryptor can reverse the padding process, remove the padding bits and produce the original plain text. The decryptor scans the decrypted padded block and discards the least significant bits that are all identical. If the data consists of bytes (e.g., 8-bit ASCII characters) then the padding indicator should be a character denoting the number of padding bytes, including itself, and should be placed in the least significant byte of the input block before encrypting. For example if there are five ASCII data characters in the final partial block of a message to be encrypted, then an ASCII "3" is put in the least significant byte of the input block (any pad characters may be used in the other two pad positions) bef ore encryption. Again the cipher text message must be marked as being padded.

In the CBC mode, one or more bit errors within a single cipher text block will affect the decryption of two blocks (the block in which the error occurs and the succeeding block). If the errors occur in the n-th cipher text block, then each bit of the n-th plain text block will have an average error rate of fifty percent. The (n+1)st plain text block will have only those bits in error which correspond directly to the cipher text bits in error.

Block synchronization between encrypt and decrypt operations is required for the CBC mode. If bits are added or are lost in a cipher text block so that block boundaries are lost between the encryption and decryptinn operations, then synchronization is lost. However, cryptographic synchronization will automatically be reestablished 64 bits after block boundaries have been established. This property is known as self-synchronization.

An example of the CBC mode is given in Table C1.

TABLE C1

AN EXAMPLE OF THE CIPHER BLOCK CHAINING (CBC) MODE

The CBC mode in the encrypt state has been selected.

Cryptographic Key = 0123456789abcdef

Initialization Vector = 1234567890abcdef

The plain text is the ASCII code for "Now is the time for all ." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...b1).

TIME	PLAIN TEXT BLOCK	DES INPUT	DES OUTPUT	CIPHER TEXT BLOCK
1	4e6f772069732074	5c5b2158f9d8ed9b	e5c7cdde872bf27c	e5c7cdde872bf27c
2	68652074696d6520	8da2edaaee46975c	43e934008c389c0f	43e934008c389c0f
3	666f7220616c6c20	25864620ed54f02f	683788499a7c05f6	683788499a7c05f6

The CBC mode in the decrypt state has been selected.

TIME	CIPHER TEXT BLOCK	DES INPUT	DES OUTPUT	PLAIN TEXT BLOCK
1	e5c7cdde872bf27c	e5c7cdde872bf27c	5c5b2158f9d8ed9b	4e6f772069732074
2	43e934008c389c0f	43e934008c389c0f	8da2edaaee46975c	68652074696d6520
3	683788499a7c05f6	683788499a7c05f6	25864620ed54f02f	666f7220616c6c20

APPENDIX D

CIPHER FEEDBACK (CFB) MODE

The CFB mode is a stream method of encryption in which the DES is used to generate pseudorandom bits which are exclusive-ORed with binary plain text to form cipher text. The cipher text is fed back to form the next DES input block. Identical messages that are encrypted using the CFB mode and different IVs will have different cipher texts. IVs that are shorter than 64 bits should be put in the least significant bits of the first DES input block and the unused, most significant, bits initialized to "0's."

In the CFB mode, errors in any K-bit unit of cipher text will affect the decryption of the garbled cipher text and also the decryption of succeeding cipher text until the bits in error have been shifted out of the CFB input block. The first af fected K-bit unit of plain text will be garbled in exactly those places where the cipher text is in error. Succeeding decrypted plain text will have an average error rate of fifty percent until all errors have been shifted out of the DES input block. Assuming no additional errors are encountered during this time, the correct plain text will then be obtained.

If K-bit boundaries are lost during decryption, then cryptographic synchronization will be lost until cryptographic initialization is performed or until 64 bits after the K-bit boundaries have been reestablished.

The encryption and decryption processes in the CFB mode both use the encrypt state of the DES. Examples of 1, 8, and 64-bit CFB mode are given in Tables D1, D2, and D3, respectively.

The 7-bit CFB alternative mode is defined in the standard in order to encipher and decipher 7-bit codes and still use an 8-bit feedback path. Most commercial implementations of the DES are designed to efficiently handle 8-bit bytes of data and key. Most computer and communication systems of recent architecture are also designed to efficiently handle full 8-bit bytes. However, some systems use the most significant bit as a parity bit. These systems often generate the parity bit during transmission and check its validity during reception. In such systems the parity bit on cipher text would be automatically modified during transmission. In this case, the encryption and decryption processes must operate independently of the parity bits and the 7-bit CFB (a) mode should be used. If the encryptor and the decryptor both set the most significant bit of the 8-bit cipher byte to be a "1" bit in the feedback, the systems are compatible. Holding no more than eight bits of the DES input constant provides an acceptable level of security for government applications.

An extension of this technique is useful in applications requiring very efficient use of the DES device. If several 7-bit data units are to be enciphered simultaneously, then a "l". bit may be put in the most significant bit position of each 8-bit byte of the feedback path. This extension of the 7-bit CFB alternative mode should be called the K-bit CFB (a) for K= 14, 21, 28, 35, 42, 49, and 56 for implementations which encipher, respectively, 2, 3, 4, 5, 6, 7, and 7-bit data units simultaneously. These alternatives provide an acceptable level of security for government applications.

Examples of 7 and 56-bit CFB (a) mode are given in tables D4 and D5, respectively.

TABLE D1

AN EXAMPLE OF THE 1-BIT CIPHER FEEDBACK (CFB) MODE

The 1-bit CFB mode in the encrypt state has been selected.

Cryptographic Key = 0123456789abcdef

Initialization Vector = 1234567890abcdef

The plain text is the binary vector (010011100110111101110111). The DES input and output blocks are written in hexadecimal notation. The ≈ represents bit-by-bit, modulo 2 addition.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	≈	0	= C
1	1234567890abcdef	bd661569ae874e25	0	≈	1	= 1
2	2468acf121579bdf	48b3169c1fac7a10	1	≈	0	= 1
3	48d159e242af37bf	0a0143394c9959fe	0	≈	0	= 0
4	91a2b3c4855e6f7e	6d52f55fd8bo2711	0	≈	0	= 0
5	234567890abcdefc	3a38debb3a2fa892	1	≈	0	= 1
6	468acf121579bdf9	719b70bd3dce7acc	1	≈	0	= 1
7	8d159e242af37bf3	81809c230adc0d23	1	≈	1	= 0
8	1a2b3c4855e6f7e6	83d14a6da6926604	0	≈	1	= 1
9	34567890abcdefcd	311e9dc8d6d52d8a	0	≈	0	= 0
10	68acf121579bdf9a	db47c7feb6fc4272	1	≈	1	= 0
11	d159e242af37bf34	b73850afa3b8ed89	1	≈	1	= 0
12	a2b3c4855e6f7e68	f5fb19ddoos9o8oo	0	≈	1	= 1
13	4567890abcdefcd1	0f4351a9bbffe5a5	1	≈	0	= 1
14	8acf121579bdf9a3	769593c58e20d41b	1	≈	0	= 1
15	159e242af37bf347	0e949d3f3a293d64	1	≈	0	= 1
16	2b3c4855e6f7e68f	921eb7ffeacd0db9	1	≈	1	= 0
17	567890abcdefcd1e	d2ad109c8895fb95	0	≈	1	= 1
18	acf121579bdf9a3d	3c36317828a9bd04	1	≈	0	= 1
19	59e242af37bf347b	e7248586e7e4ecac	1	≈	1	= 0
20	b3c4855e6f7e68f6	f9a58e16a7597c5e	1	≈	1	= 0
21	67890abcdefcd1ec	e939fdf63d177946	0	≈	1	= 1
22	cf121579bdf9a3d9	f325eac046bad58d	1	≈	1	= 0
23	9e242af37bf347b2	8385a6d975ffdbba	1	≈	1	= 0
24	3c4855e6f7e68f64	70a54baceae7ba6b	1	≈	0	= 1

TABLE D2

AN EXAMPLE OF THE 8-BIT CIPHER FEEDBACK (CFB) MODE

The 8-bit CFB mode in the encrypt state has been selected.

Cryptographic Key - 0123456789abcdef

Initialization Vector = 1234567890abcdef

The plain text is the ASCII code for "Now is the." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...b1). The ≈ represents bit-by-bit, modulo 2 addition.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	≈	0	= C
1	1234567890abcdef	bd661569ae874e25	4e	≈	bd	= f3
2	34567890abcdeff3	7039546f9a0f6330	6f	≈	70	= 1f
3	567890abcdeff31f	ad1b78b0bb371be7	77	≈	ad	= da
4	7890abcdeff31fda	2735b01d5ca31f7	20	≈	27	= 07
5	90abcdeff31fda07	68863426e397685d	69	≈	68	= 01
6	abcdeff31fda0701	6798240e8c6b685f	73	≈	67	= 14
7	cdeff31fda070114	421feefb3f8ca64f	20	≈	42	= 62
8	eff31fda07011462	9a169a9b50666575	74	≈	9a	= ee
9	f31fda07011462ee	703b1799be9a5748	68	≈	70	= 18
10	1fda07011462ee18	1a4aee195be70077	65	≈	1a	= 7f

The 8-bit GFB mode in the decrypt state has been selected.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	C	≈	O	= P
1	1234567890abcdef	bd661569ae874e25	f3	≈	bd	= 4e
2	34567890abcdeff3	7039546f9a0f6330	1f	≈	70	= 6f
3	567890abcdeff31f	ad1b78b0bb371be7	da	≈	ad	= 77
4	7890abcdeff31fda	27350b01d5ca31f7	07	≈	27	= 20
5	90abcdeff31fda07	68863426e397685d	01	≈	68	= 69
6	abcdeff31fda0701	6798240e8c6b685f	14	≈	67	= 73
7	cdeff31fda070114	421feefb3f8ca64f	62	≈	42	= 20
8	eff31fda07011462	9a169a9b50666575	ee	≈	9a	= 74
9	f31fdaf17011462ee	703b1799be9a5748	18	≈	70	= 68
10	1fda07011462ee18	1a4aee195be70077	7f	≈	1a	= 65

TABLE D3

AN EXAMPLE OF ThE 64-BIT CIPHER FEEDBACK (CFB) MODE

The 64-bit CFB mode in the encrypt state has been selected.

Cryptographic Key - 0123456789abcdef

Initialization Vector - 1234567890abcdef

The plain text is the ASCII code for "Now is the time for all ." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...,b1).

TIME	PLAIN TEXT BLOCK	DES INPUT	DES OUTPUT	CIPHER TEXT BLOCK
1	4e6f772069732o74	1234567890abcdef	bd661569ae874e25	f3096249c7f46e51
2	68652o74696d6s20	f3096249c7f46e51	cefba3ef73ff92a4	a69e839b1a92f784
3	666f7220616c6c20	a69e839b1a92f784	65290313e8e2ca02	03467133898ea622

The 64-bit CFB mode in the decrypt state hu been selected.

TIME	CIPHER TEXT BLOCK	DES INPUT	DES OUTPUT BLOCK	PLAIN TEXT
1	f3096249c7f46e51	234567890abcdef	bd661569ae874e25	4e6f772069732074
2	a69e839b1a92f784	f3096249e7f46e51	cefba3ef73ff92a4	68652o74696d6520
3	03467133898ea622	a69e839b1a92f784	65290313e8e2ca02	666f7220616c6c20

TABLE D4

AN EXAMPLE OF THE 7-BIT CIPHER FEEDBACK ALTERNATIVE MODE

The 7-bit CFB(a) mode in the encrypt state has been selected. Cryptographic Key - 0123456789abcdef

Initiaiization Vector - 1234567890abcdef

The plain text is the ASCII code for "Now is the." These seven-bit characters are written in hexadecimal notation (0, b7, b6,...b1). The ≈ represents bit-by-bit, modulo 2 addition.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	+	O	= C
1	1234567890abcdef	bd661569ae874e25	4e	≈	bd	= 73
2	34567890abcdeff3	7039546f9a0f6330	6f	≈	70	= 1f
3	567890abcdeff39f	e86e0d3772221b21	77	≈	e8	= 1f
4	7890abcdeff39f9f	cbb91f82946f3a68	20	≈	cb	= 6b
5	90abcdeff39f9feb	9faf68acc9d1c4f9	69	≈	9f	= 76
6	abcdeff39f9febf6	bf7e7edc468df70f	73	≈	bf	= 4c
7	cdeff39f9febf6cc	6a555c03e8c20cea	20	≈	6a	= 4a
8	eff39f9febf6ccca	d8bb411744869e4a	74	≈	d8	= 2c
9	f39f9febf6cccaac	e656f81f3f1a8c28	68	≈	e6	= 0e
10	9f9febf6cccaac8e	cd1883fe15bf7c26	65	≈	cd	= 28

The 7-bit CFB(a) mode in the decrypt state has been selected.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	C	≈	0	= P
1	1234567890abcdef	bd661569ae874e25	73	≈	bd	= 4e
2	34567890abcdeff3	7039546f9a0f6330	1f	≈	70	= 6f
3	567890abcdeff39f	e86e0d3772221b21	1f	≈	8	= 77
4	7890abcdeff39f9f	cbb91f82946f3a68	6b	≈	cb	= 20
5	90abcdeff39f9feb	9faf68acc9d1c4f9	76	≈	9f	= 69
6	abcdeff39f9febf6	bf7e7edc468df70f	4c	≈	bf	= 73
7	cdeff39f9febf6cc	6a555c03e8c20cea	4a	≈	6a	= 20
8	eff39f9febf6ccca	d8bb411744869e4a	2c	≈	d8	= 74
9	f39f9febf6cccaac	e656f8f3f31a8c28	0e	≈	e6	= 68
10	9f9febf6cccaac8e	cd1883fe15bf7c26	28	≈	cd	= 65

TABLE D5

AN EXAMPLE OF THE 56-BIT CIPHER FEEDBACK ALTERNATIVE MODE

The 56-bit CFB(a) mode in the encrypt state has been selected.

Cryptographic Key - 0123456789abcdef

Initialization Vector - 1234567890abcdef

The plain text is the ASCII code for "Now is the time for all " These seven-bit characters are written in hexadecimal notation (0, b7, b6,. .b1).

TIME	PLAIN TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	CIPHER TEXT
1	4e6f772069732074	1234567890abcdef	bd661569ae874e25	7309624947746e51
2	68652074696d6520	f389e2c9c7f4eed1	8988dd3d6b71f76b	616d7d49021c24b
3	666f7220616c6c20	e1edfdc9829c92cb	314a61d117be7e4d	572513717652126d

The 56-bit CFB(a) mode in the decrypt state has been selected.

TIME	CIPHER TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	PLAIN TEXT
1	7309624947746e51	1234567890abcdef	bd661569ae874e25	4e6f772069732074
2	616d7d49021c124b	f389e2c9c7f4eed1	8988dd3d6b71f76b	68652074696d6520
3	572513717652126d	e1edfdc9829c92cb	314a61d117be7e4d	666f7220616c6c20

APPENDIX E

OUTPUT FEEDBACK (OFB) MODE

The Output Feedback (OFB) mode is an additive stream cipher in which errors in the cipher text are not extended to cause additional errors in the decrypted plain text. One bit in error in the cipher text causes only one bit to be in error in the decrypted plain text. Therefore, this mode cannot be used for data authentication but is useful in applications where a few errors in the decrypted plain text are acceptable.

In the OFB mode, the same K bits of the DES output block that are used to encrypt a K-bit unit of plain text are fed back for the next input block. This feedback is completely independent of all plain text and all cipher text. As a result, there is no error extension in OFB mode.

If cryptographic synchronization is lost in the OFB mode, then cryptographic initialization must be performed. The OFB mode is not a self-synchronizing cryptographic mode.

Examples of 1-bit OFB and 8-bit OFB are given in Tables El and E2, respectively.

TABLE E1

AN EXAMPLE OF THE 1-BIT OUTPUT FEEDBACK (OFB) MODE

The 1-bit OFB mode in the encrypt state has been selected.

Cryptographic Key - 0123456789abcdef Initialization Vector - 1234567890abcdef

The plain text is the binary vector (010011100110111101110111).
The ≈ represents bit-by- bit, modulo 2 addition.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	≈	0	= C
1	1234567890abcdef	bd661569ae874e25	0	≈	1	= 1
2	2468acf121579bdf	48b3169c1fac7a10	1	≈	0	= 1
3	48d159e242af37be	8879ea93c63d77a5	0	≈	1	= 1
4	91a2b3c4855e6f7d	0d36e16101e86d61	0	≈	0	= 0
5	234567890abcdefa	e9eab8cfc00f4ac3	1	≈	1	= 0
6	468acf121579bdf5	9d41640f97df7904	1	≈	1	= 0
7	8d159e242af37beb	32f72fd1899eda45	1	≈	0	= 1
8	1a2b3c4855e6f7d6	ca2a095d20f4e769	0	≈	1	= 1
9	34567890abcdefad	de869588355e1041	0	≈	1	= 1
10	68acf121579bdf5b	11245e6a8720ddce	1	≈	0	= 1
11	d159e242af37beb6	836b0be324094a97	1	≈	1	= 0
12	a2b3c4855e6f7d6d	c07714703b296a5a	0	≈	1	= 1
13	4567890abcdefadb	bf6380ecc496d599	1	≈	1	= 0
14	8acf121579bdf5b7	96ed6856969aef13	1	≈	1	= 0
15	159e242af37beb6f	3823feaa3d170085	1	≈	0	= 1
16	2b3c4855e6f7d6de	2d57dc0c899d6700	1	≈	0	= 1
17	567890abcdefadbc	2fe1c261c0e1a302	0	≈	0	= 0
18	acf121579bdf5b78	778ad641faa047d0	1	≈	0	= 1
19	59e242af37beb6f0	f66ae4359eec3755	1	≈	1	= 0
20	b3c4855e6f7d6de1	cd0bda27e32a13da	1	≈	1	= 0
21	67890abcdefadbc3	9f71f74488551801	0	≈	1	= 1
22	cf121579bdf5b787	a62e89aa6b85be74	1	≈	1	= 0
23	9e242af37beb6f0f	7b0b2e1de987b804	1	≈	0	= 1
24	3c4855e6f7d6de1e	7f41b5ef07c3ea29	1	≈	0	= 1

TABLE E2

AN EXAMPLE OF THE 8-BIT OUTPUT FEEDBACK (OFB) MODE

The 8-bit OFB mode in the encrypt state has been selected.

Cryptographic Key - 0123456789abcdef

Initialization Vector - 1234567890abcdef

The plain text is the ASCII code for "Now is the." These seven-bit characters are written in hexadecimal notation (0,b7,b6,.. .,b1). The ≈ represents bit-by-bit, modulo 2 addition.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	≈	0	= C
1	1234567890abcdef	bd661569ae874e25	4e	≈	bd	= f3
2	34567890abcdefbd	25e73b5d4cbd2359	6f	≈	25	= 4a
3	567890abcdefbd25	5f97oo7o5s3623do	77	≈	5f	= 28
4	7890abcdefbd255f	704ad48bf9eec8fa	20	≈	70	= 50
5	90abcdefbd255f7o	a0b1a091bb7875s3	69	≈	a0	= c9
6	abcdefbd255f7oao	b58127681139ee7f	73	≈	b5	= c6
7	cdefbd255f70a0b5	694d556ef5806a65	20	≈	69	= 49
8	efbd255f70a0b569	f1885324299132a2	74	≈	f1	= 85
9	bd255f70a0b569f1	be639ff6d7b74bo4	68	≈	be	= d6
10	255f70a0b569f1be	e17b6ae22b4bad65	65	≈	e1	= 84

The 8-bit OFB mode in the decrypt state has been selected.

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	C	≈	O	= P
1	1234567890abcdef	bd661569ae874e25	f3	≈	bd	= 4e
2	34567890abcdefbd	25e73b5d4cbd2359	4a	≈	25	= 6f
3	567890abcdefbd25	5f97oo7oss3623do	28	≈	5f	= 77
4	7890abcdefbd255f	704ad48bf9eec8fa	50	≈	70	= 20
5	90abcdefbd255f7o	a0b1a091bb787553	c9	≈	a0	= 69
6	abcdefbd255f70ao	b58127681139ee7f	c6	≈	b5	= 73
7	cdefbd255f7oa0b5	694d556ef5806a65	49	≈	69	= 20
8	efbd255f70aob569	f1885324299132a2	85	≈	f1	= 74
9	bd255f70a0b569f1	be639ff6d7b74b04	d6	≈	be	= 68
10	255f70a0b569f1be	e17b6ae22b4bad65	84	≈	e1	= 65

APPENDIX F

DES AUTHENTICATION TECHNIQUE

The DES can be used for message (data) authentication. A Message Authentication Code (MAC) is generated (computed) as a cryptographic function of the message (data). The MAC is then stored or transmitted with the data. Only those knowing the secret key can recompute the DEAD for the received message and verify that the message has not been modified by comparing the computed DEAD with the stored or transmitted DEAD. An unauthorized recipient of the data who does not possess the key cannot modify the data and generate a new DEAD to correspond with the modified data. This technique is useful in applications which require maintaining data integrity but which do not require protecting the data from disclosure. For example, computer programs may be stored in plain text form with a computed DEAD appended to the program file. The program may be read and executed without decryption. However, when the integrity of the program is questioned, a MAC can be computed on the program file and compared with the one stored in the file. If the two MAC's are identical and the cryptographic key used to generate the MAC has been protected, then the program file has not been modified.

A MAC may be generated using either the CBC or the CFB mode. In CBC authentication, a message is encrypted in the normal CBC manner but the cipher text is discarded. Messages which terminate in partial data blocks must be padded on the right (LSB) with zeros. In CBC authentication, the most significant M bits of the final output block are used as the MAC, where M is the number of bits in the MAC.

In CFB authentication, a message is encrypted in the normal CFB manner except that the cipher text is discarded. After encrypting the final K bits of data and feeding the resulting cipher text back into the DES input block, the DES device is operated one more time and the most significant M bits of the resulting DES output block are used as the MAC

In both CBC and CFB authentication, a MAC should be used that is as long as practical. Since a MAC is an error detection code (which is computed using cryptographic techniques), a long MAC is desirable. Bit manipulation within a message using a MAC of length M will be detectable with a probability of 1-(1/2≠N). Concluding that a message has not been modified is based upon this probability. The proposed Federal Standard 1026 requires M to be at least 24 for Federal telecommunication applications. Financial transaction application standards are recommending M to be 32. Application designers should select M to optimize security and efficiency requirements.

In ADP communications security applications a message numbering and verifying system should be used to protect against insertion of false messages, deletion of valid messages, and replay of a previously valid message. The combined use of a unique Message Identifier (MID) and a MAC achieves these security objectives in addition to protecting the message against message modification. If the data source MAC and the data destination MAC are in agreement and if the MID agrees with the value expected by the receiver, then these four security objectives have been accomplished. The MID should be unique and deterministic for each message transmitted between a sender and receiver. The uniqueness may be achieved through the use of a simple binary counter.

Examples of the MAC calculation using CBC and 8-bit CFB are given in Tables F1 and F2, respectively.

TABLE F1

AN EXAMPLE OF THE CIPHER BLOCK CHAINING (CBC) MODE
FOR AUTHENTICATION

The CBC mode in the encrypt state has been selected.

Cryptographic Key 0123456789abcdef

Initialization Vector - 1234S67890abcdef

The plain text is the ASCII code for "7654321 Now is the time for ." These seven-bit characters are written in hexadecimal notation (0,b7,b6,..b1).

TIME	PLAIN TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK
1	3736353433323120	2502634ca399fccf	b9916b8ee4c3da64
2	4e6f772069732074	f7felcae8db0falo	b4f44e3cbefb9948
3	68652074696d6520	dc916e48d796fc68	4S2l388fa59ae67d
4	666f722000000000	234e4aafa59ae67d	058d2e77e86062733

32-bit MAC is selected.

TEXT	MAC
3736353433323l204e6f772068732074686S2074696d6520666f722058d2e77e	

TABLE F2

AN EXAMPLE OF THE CIPHER FEEDBACK (CFB)
MODE FOR AUTHENTICATION

The 8-bit CFB mode in the encrypt state has been selected.

Cryptographic Key -123456789abcdef

Initialization Vector - 1234S67890abcdef

TIME	DES	INPUT BLOCK	DES OUTPUT BLOCK	P ≈ O = C
	1	234567890abcdef	bd661569ae874e25	37 ≈ bd = 8a
	2	34567890abcdef8a	b156f27e4084b3e1	36 ≈ b1 = 87
	3	57890abcdef8a87	346594a9f532e7ef	35 ≈ 34 = 01
	4	7890abcdef8a87o1	ed228c3dObo87e56	34 ≈ ed = d9
	5	90abcdef8a87O1d9	12fffb7d10c59f6e	33 + 12 = 21
	6	abcdef8a87o1d921	O2de319634551992	32 + 02 = 30
	7	cdef8a8701d92130	be3ee94f5b0d9337	31 + be = 8f
	8	ef8a87o1d9213o8f	15a8855f3e99o8b3	20 + 15 = 35
	9	8a8701d921308f35	3af549c9c870562c	4e + 3a = 74
	10	87o1d9213o8f3574	d2b323ada61cdeoo	6f + d2 = bd
	11	O1d9213o8f3574bd	6977832969dbbeba	77 + 69 = 1e
	12	d921308f3574bd1e	5473999aba6c9813	20 + 54 = 74
	13	21308f3574bd1e74	9db2dcb11bcefd56	69 + 9d = f4
	14	308f3574bd1e74f4	41dd4dfde3648513	73 + 41 = 32
	15	8f3574bd1e74f432	349de10f1d65672o	20 + 34 = 14
	16	3574bd1e74f43214	0384e72851495e94	74 + 03 = 77
	17	74bd1e74f4321477	64aeb25d7a54bb91	68 + 64 = 0c
	18	bd1e74f4321477oc	IfO7839f59391e53	65 + 1f = 7a
	19	Ie74f4321477oc7a	I4d3c21640e42157	20 + 14 = 34
	20	74f43214770c7a34	fb7a853aadb39183	74 + fb = 8f
	21	f43214770c7a348f	edee83bOao7afcd4	69 + ed = 84
	22	3214770c7a348f84	5065694b1a1b765c	6d + 50 = 3d
	23	I477oc7a348f843d	68ec7ad3602e91c2	65 + 68 = Od
	24	770c7a348f843d0d	28f5c32ae7b4495f	20 + 28 = 08
	25	Oc7a348f843dodo8	523d79cb8d3eb462	66 + 52 = 34
	26	7a348f843d0d 834	dd5816fac447o533	6f + dd = b2
	27	348f843d0d0834b2	b61ec6Of26c3b29a	72 + b6 = c4
	28	8f843dOdO834b2c4	daca26833o988a7d	20 + da = fa
	29	843d0dO834b2c4fa	cd647403bc9oc4c4	

32-bit MAC selected.

TEXT MAC

3736353433332312O4e6f772068732074686520746965d6520666f7220cd647403