U.S. Department of Commerce

National Bureau of Standards

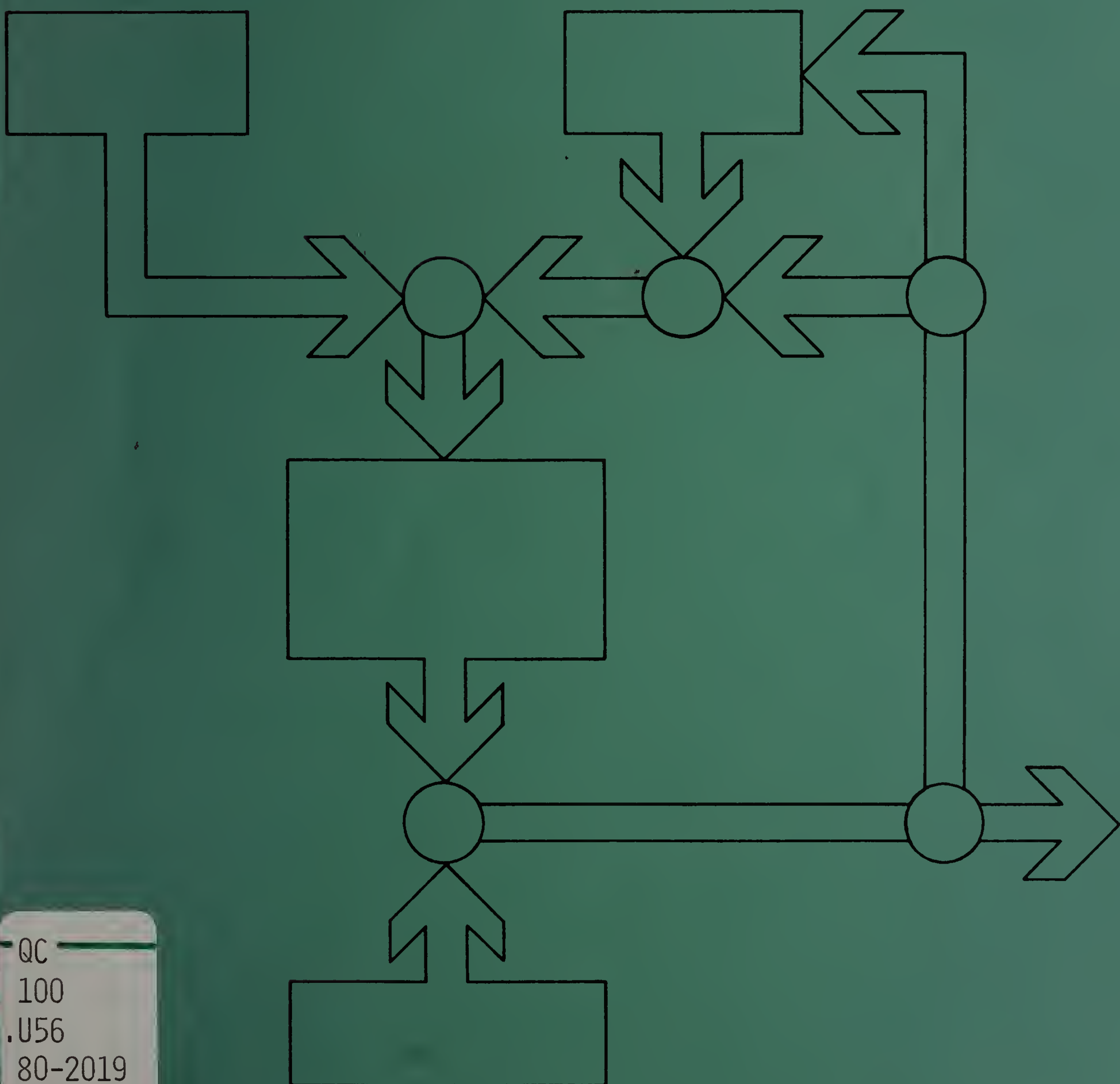
[redacted]

NBSIR 80-2019

Technical Specifications of a Proposed Federal Information Processing Standard on the Modes of Operation for the Data Encryption Standard

Institute for Computer Sciences and Technology

April 1980



Please correct the cryptographic key to read "23016745ab89efcd" in all examples in the appendices of NBSIR 80-2019.

NBSIR 80-2019

Technical Specifications of a Proposed Federal Information Processing Standard on the Modes of Operation for the Data Encryption Standard

MICHAEL J. O'BRIEN

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Issued April 1980

U.S. Department of Commerce
Philip M. Klutznick, Secretary
Luther H. Hodges, Jr., Deputy Secretary
Jordan J. Baruch, Assistant Secretary for
Productivity, Technology, and Innovation

National Bureau of Standards
Ernest Ambler, Director

TABLE OF CONTENTS

1. PREFACE

The algorithm for the Data Encryption Standard (DES) was developed by the International Business Machines Corporation (IBM). It was adopted by the National Bureau of Standards (NBS) as a Federal Information Processing Standard (FIPS) in 1977. FIPS Publication #46 [FIPS 46] specifies the DES algorithm which is to be used within the Federal government for the cryptographic protection of sensitive, but unclassified, computer data. A number of techniques for incorporating this algorithm into a cryptographic system have been identified by both Federal and private organizations. These implementation techniques, external to the DES algorithm, have come to be called the "modes of operation." The Institute for Computer Sciences and Technology within the NBS is proposing a Modes of Operation FIPS for the DES. The purpose of this FIPS will be to describe several techniques for using the DES with sufficient specificity as to facilitate the interoperability of equipment using these modes.

Four implementation techniques using the DES are described in this document: the electronic codebook (ECB) mode, the cipher block chaining (CBC) mode, the cipher feedback (CFB) mode, and the authentication-only mode. ECB is a direct implementation of the IBM algorithm (U.S. patents #3796830 and #3798359); IBM also developed and patented (#4078152) the basic concept of the CBC mode. The Federal Reserve Board, with the technical assistance of the National Security Agency, adopted an 8-bit CFB technique for experimental use on their nationwide data communication network. The authentication-only mode is really an application of CBC or CFB, but it is deemed sufficiently important to be included in a standard. The proposed FIPS is limited to these four modes because they are the only techniques recommended at this time in encryption standards being developed under the auspices of the Federal Telecommunication Standards Committee.

The purpose of this NBS Internal Report is to provide an expedient vehicle for the dissemination of the technical information being considered for the proposed Modes of Operation FIPS.

The proposed FIPS will mandate only those characteristics necessary to specify the mechanics of implementing the modes of operation. Requirements in other concomitant areas which affect the security of a cryptographic system, e.g., key management or cryptographic synchronization, are not addressed in this document. They may be defined in other security or application standards.

The American National Standards Institute has approved the creation of a technical committee (X3T1) in order to begin drafting a national standard addressing the modes of operation for their Data Encryption Algorithm (DEA). The Federal DES and the ANSI DEA use the same cryptographic algorithm.

2. INTRODUCTION

Data to be cryptographically protected is called plain text. Encryption is the process of transforming plain text into cipher text ; decryption is the inverse mapping of cipher text to plain text. The encryption (E) of plain text (P) under a key (K) into cipher text (C) is denoted by E(K, P) = C. The letter D will represent the inverse transformation, so that decryption under K may be written as D(K, C) = D{K, E(K, P)} = P.

Binary data may be cryptographically protected using the Data Encryption Standard (DES) [FIPS 46] in conjunction with a cryptographic variable. A cryptographic variable for the DES consists of sixty-four binary digits of which fifty-six bits are used directly as a key governing the algorithm. The remaining eight bits are employed as an odd parity check. A cryptographic period (or key period) is that interval of DES operation during which the same key is used between two or more cryptographic entities. Since the DES has been publicly defined, cryptographic security depends upon the security provided for the cryptographic variable -- both the key and its parity bits. Given the cipher text and the key, the plain text can be recovered easily.

Mathematically, the DES maps a 64-dimensional input space over the field {0,1} onto itself. The number of elements in this space is two raised to the 64th power (2**64), i.e., it consists of all possible 64-bit vectors. The cryptographic key space provides the user a choice of any one of 2**56 invertible (one-to-one and onto) mappings. A specific DES input value can be mapped to one of 2**56 output values -- the specific value depends upon the particular 56-bit key chosen. The DES mapping has a complementary effect in that if E(K, P) = C then E(K',P') = C' or equivalently E(K, P) = {E(K', P')}' where the apostrophe represents binary complementation.

FIGURE 1:

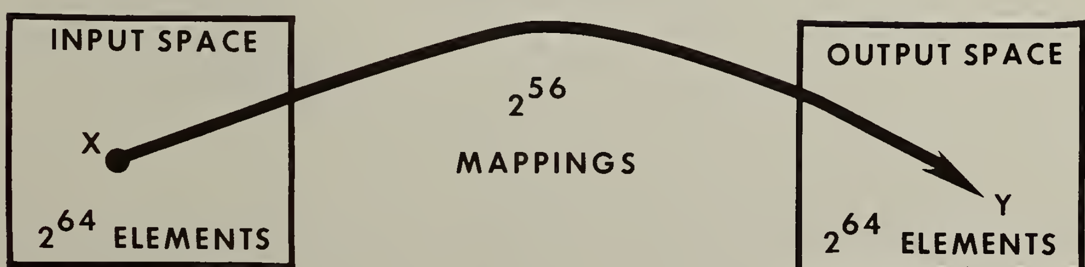
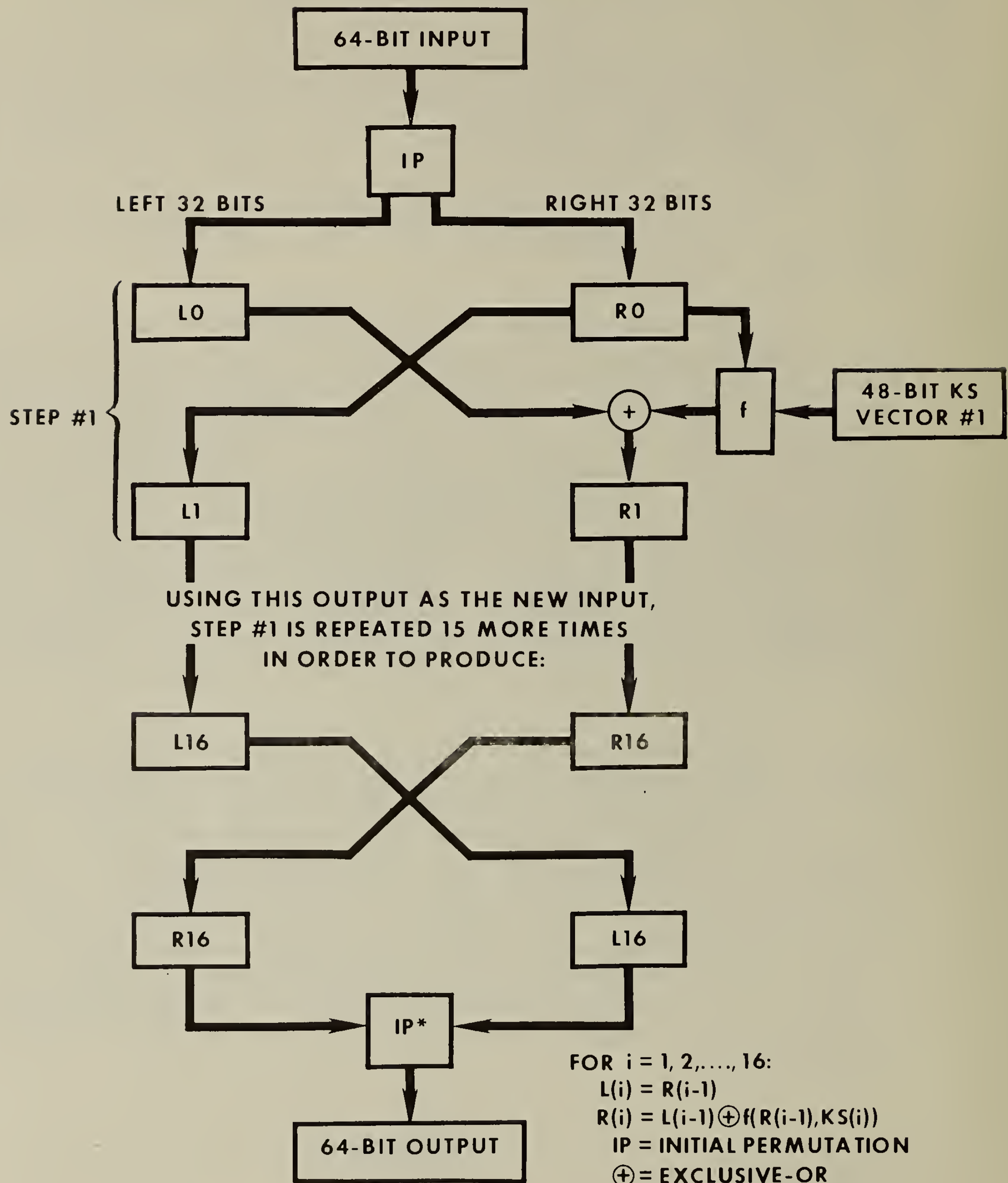


FIGURE 2: THE DES TRANSFORMATION



FOR $i = 1, 2, \ldots, 16$:

$L(i) = R(i-1)$

$R(i) = L(i-1) \oplus f(R(i-1), KS(i))$

IP = INITIAL PERMUTATION

$\oplus$ = EXCLUSIVE-OR

f = CIPHER FUNCTION

KS = KEY SCHEDULE FUNCTION

IP* = INVERSE OF IP

The DES is a nonlinear, iterative, block, product cipher. This product cipher mixes transposition and substitution operations in an alternating manner in order to be most effective. Because this algorithm maps a 64-bit input block into sixty-four output bits, the DES is classified as a block cipher. Iterative refers to the use of the output of an operation as the input for another iteration or round of the same procedure. Nonlinearity is introduced into the algorithm through eight S (substitution) boxes, each of which maps six input bits into four output bits. A block diagram of the encryption operation is illustrated in figure #2. The cipher function, f, and the key schedule function are described in detail in [FIPS 46].

The DES input and output (I/O) blocks are sixty-four bit vectors with the least significant bit (LSB = 2**0) defined to be on the right and the most significant bit (MSB = 2**63) on the left. The bits of a DES I/O block are numbered from left to right: (1, 2, . . . , 64). When a 64-bit cryptographic variable is entered into the DES key schedule, its format in the key input block is: (1,2,...,7,P1,8,9,..., 14,P2,15,...,...,49,P7,50,51,...,56,P8), where {Pi | i=1, 2,...,8} are odd parity bits computed on the preceding seven key bits.

There are two general techniques for incorporating the DES into a cryptographic system: a block cipher and an additive or stream cipher. In both modes each bit of cipher text is a function of every bit of the cryptographic key. In a block cipher, the DES input block is a function of the plain text to be encrypted and the DES output block defines the cipher text. In an additive cipher implementation, a pseudorandom binary sequence is generated using DES output blocks. The binary exclusive-OR operation, represented by a circled plus sign, combines this pseudorandom sequence with the plain text to define the cipher text. This operation is equivalent to bit-by-bit, modulo-2 addition (without carry). Since the exclusive-OR operator is its own inverse over {0,1}, the same pseudorandom binary stream, say O, is used for both the encryption of plain text, P, and the decryption of cipher text, C; i.e., $P \oplus O = C$ and $C \oplus O = P$.

3. ELECTRONIC CODEBOOK (ECB) MODE

The most basic mode of operation for the DES is the electronic codebook (ECB). 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, assuming that a manageable subset of the DES input space is used, a list (or codebook) of corresponding plain and cipher text pairs could be constructed.

In ECB encryption, the plain text data block (D1,D2,.., D64) directly defines the DES input block (I1,I2,...,I64). The input block is processed through a DES device which has been loaded with the appropriate cryptographic variable. The resultant output block (O1,O2,...,O64) is used directly as cipher text (C1,C2,...,C64).

The ECB decryption process is the same as ECB encryption except that the DES key schedule selection is reversed. In general, the DES key schedule function generates a new 48-bit vector from the 56-bit cryptographic key for each of the sixteen rounds of the DES algorithm. For a given cryptographic variable, let the sixteen key schedule encryption vectors be denoted by {KS1, KS2, . . . , KS16}. Then, the corresponding decryption process will use the same basic operation as encryption (figure #2), but now {KS16, KS15, . . . , KS1} will be successively invoked. An example of ECB encryption and decryption may be found in Appendix A.

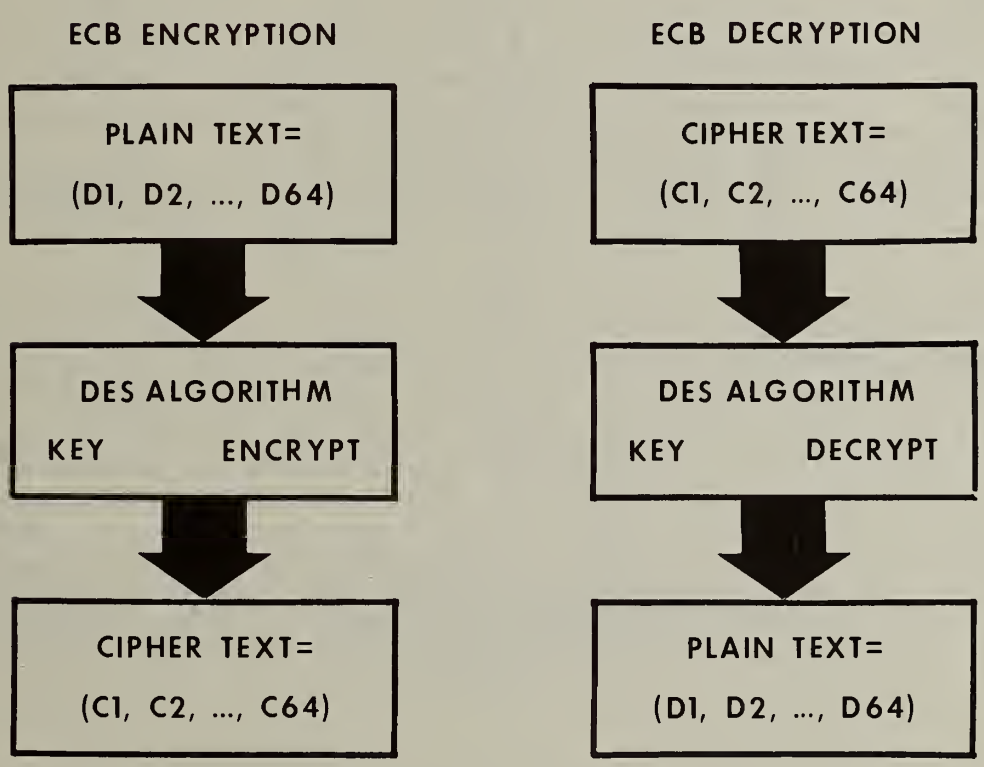
Since each bit of an ECB output block is a complex function of every bit in the input block and the cryptographic key, a single bit error in a cipher text block will cause the decrypted plain text block to have an average error rate of fifty percent. However, an error in one ECB cipher text block will not affect the decryption of other blocks, i.e., there is no error extension between ECB blocks. An example of the effect of cipher text errors on ECB operations may be found in Appendix B.

If block synchronization is lost (e.g., a bit slip), then ECB cryptographic synchronization will also be lost until correct block boundaries are re-established.

Since the ECB mode is a 64-bit block cipher, an ECB device must encrypt information 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 should be padded with random or pseudorandom binary digits prior to ECB encryption. The corresponding decrypting device will have to know when and to what extent padding has taken place so that these padding digits can be ignored or discarded after decryption.

A potentially critical weakness of the ECB mode is the fact that the same plain text always produces the same cipher text under a fixed key. Thus, the compromise of the plain text underlying any cipher text block results in the compromise of all repetitions of this same text for the remainder of the cryptographic period. This is sometimes referred to as a codebook analysis problem.

FIGURE 3: ELECTRONIC CODEBOOK (ECB) MODE



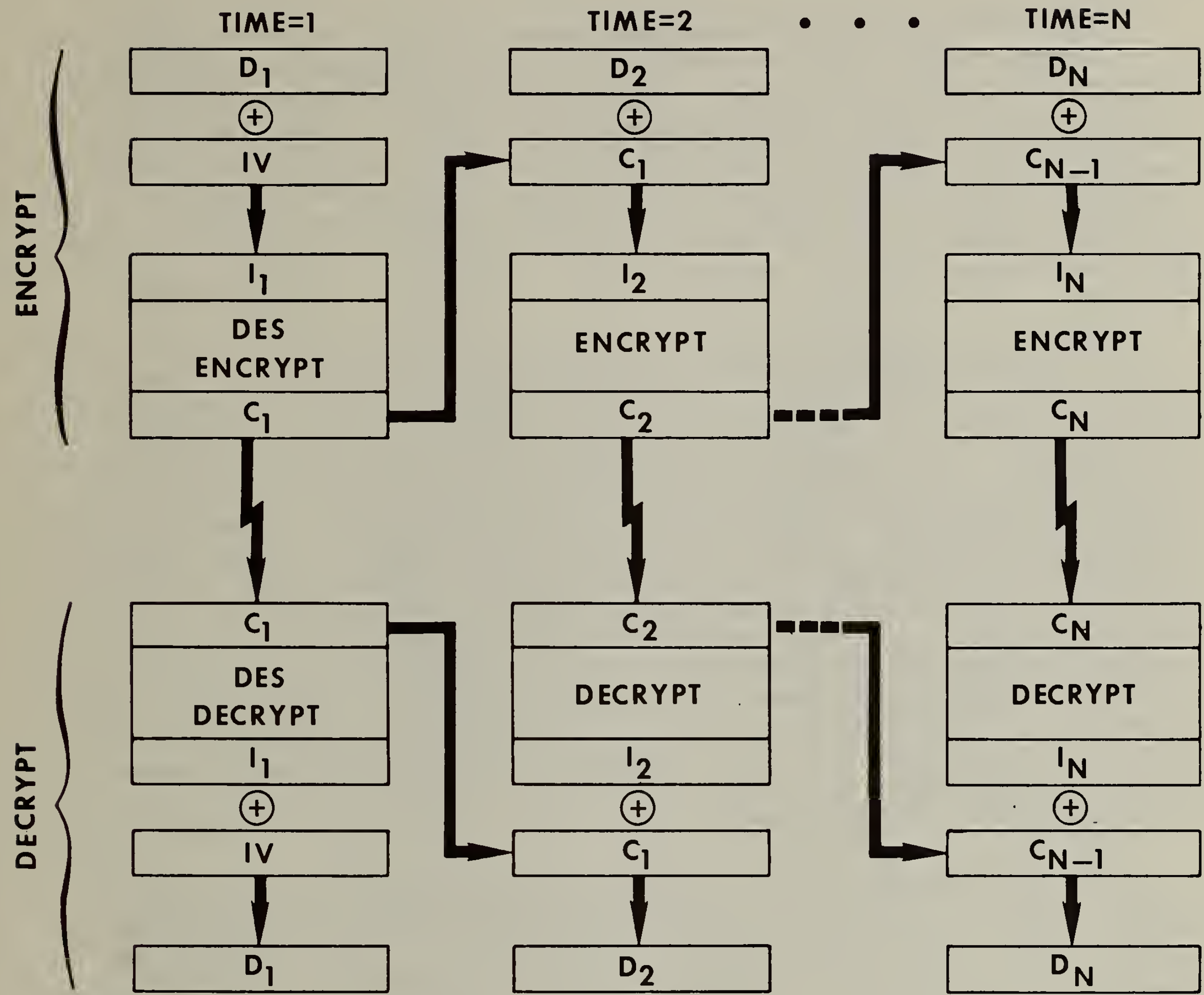
4. CIPHER BLOCK CHAINING (CBC) MODE

CBC is a block cipher in which the plain text is exclusive-ORed with a block of pseudorandom data prior to being processed through a DES device (see figure #4). This technique greatly reduces the codebook analysis problem associated with the ECB mode, and also provides an error extension characteristic which is valuable in protecting against fraudulent data alteration.

In order to commence CBC encryption, the first DES input block is formed by exclusive-ORing the first data block with a 64-bit initialization vector (IV), i.e., (I1,I2,..., I64) = (IV1⊕D1, IV2⊕D2, . . . , IV64⊕D64). This initial CBC input block is processed through a DES device producing a 64-bit DES output block which defines the cipher text, (O1,O2,...,O64) = (C1,C2,...,C64). Then these first 64 bits of cipher text are exclusive-ORed with the second plain text data block in order to construct the second DES input block. The next DES operation produces 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. The DES input block at time t is the bit-by-bit, mod-2 sum of the plain text at time t and the cipher text at time t-1 for t>1 or the IV at t=1. Appendix A contains an example of the CBC encryption and decryption of ASCII characters and Appendix C has a flow chart illustrating the basic CBC logic.

For CBC decryption, the first cipher text block is processed through a DES device using the decrypt operation, i.e., the key schedule vectors are invoked in the reverse order from the encryption process. The first output block is exclusive-ORed with the CBC IV producing the first plain text block. The second cipher text block is then entered into the DES and the resultant output block is exclusive-ORed with the first cipher text block in order to produce the second plain text block. The CBC decryption process continues to exclusive-OR the cipher text block at time t-1 for t>1 with the DES output block to obtain plain text at time t until the end of the message.

FIGURE 4: CIPHER BLOCK CHAINING (CBC) MODE



LEGEND

Di=DATA AT TIME i
Ii=INPUT AT TIME i
Ci=CIPHER AT TIME i
Oi = OUTPUT AT TIME i
IV = INITIALIZATION VECTOR
⊕ EXCLUSIVE-OR

The security of a CBC implementation depends, among other things, upon the management of CBC initialization vectors. Some recommendations in this area are provided in Appendix E.

The CBC mode reproduces the same cipher text whenever the same plain text is encrypted under a fixed key and IV. In the ECB mode, the cipher text repetition characteristic occurs at the block level; in the CBC mode, cipher text repetition is at the message level. If CBC users are concerned about this potential security problem, then their CBC systems should incorporate a unique identifier (e.g., a one-up counter) at the beginning of each CBC message in order to insure unique cipher text within a cryptographic period.

Since the CBC mode is a 64-bit block cipher, it must operate on a 64-bit input block with each CBC operation. Thus, partial data blocks (< 64 bits) will require special handling. For example, a partial data block may be padded in its least significant bit positions with arbitrary binary digits whenever the application environment can tolerate the overhead. The decrypting CBC device will have to know when and to what extent padding has occurred. This can be accomplished explicitly, e.g., using a control indicator, or implicitly, e.g., using constant length transactions. Another suggestion for handling partial data blocks is to switch to a 1-bit cipher feedback (CFB) mode in order to process the final $k < 64$ bits of a message. The last CBC cipher block would be used as an IV to initiate this CFB process. When using this scheme, the last bit of the message should not contain sensitive information.

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 about 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. Of course, if errors occur in the last cipher text block, then the last plain text block is the only one affected. An example of the effect of cipher text errors in CBC operations may be found in Appendix B.

If CBC block synchronization is lost, then CBC cryptographic synchronization will also be lost. However, cryptographic synchronization will automatically be reacquired sixty-four bits after block boundaries have been established.

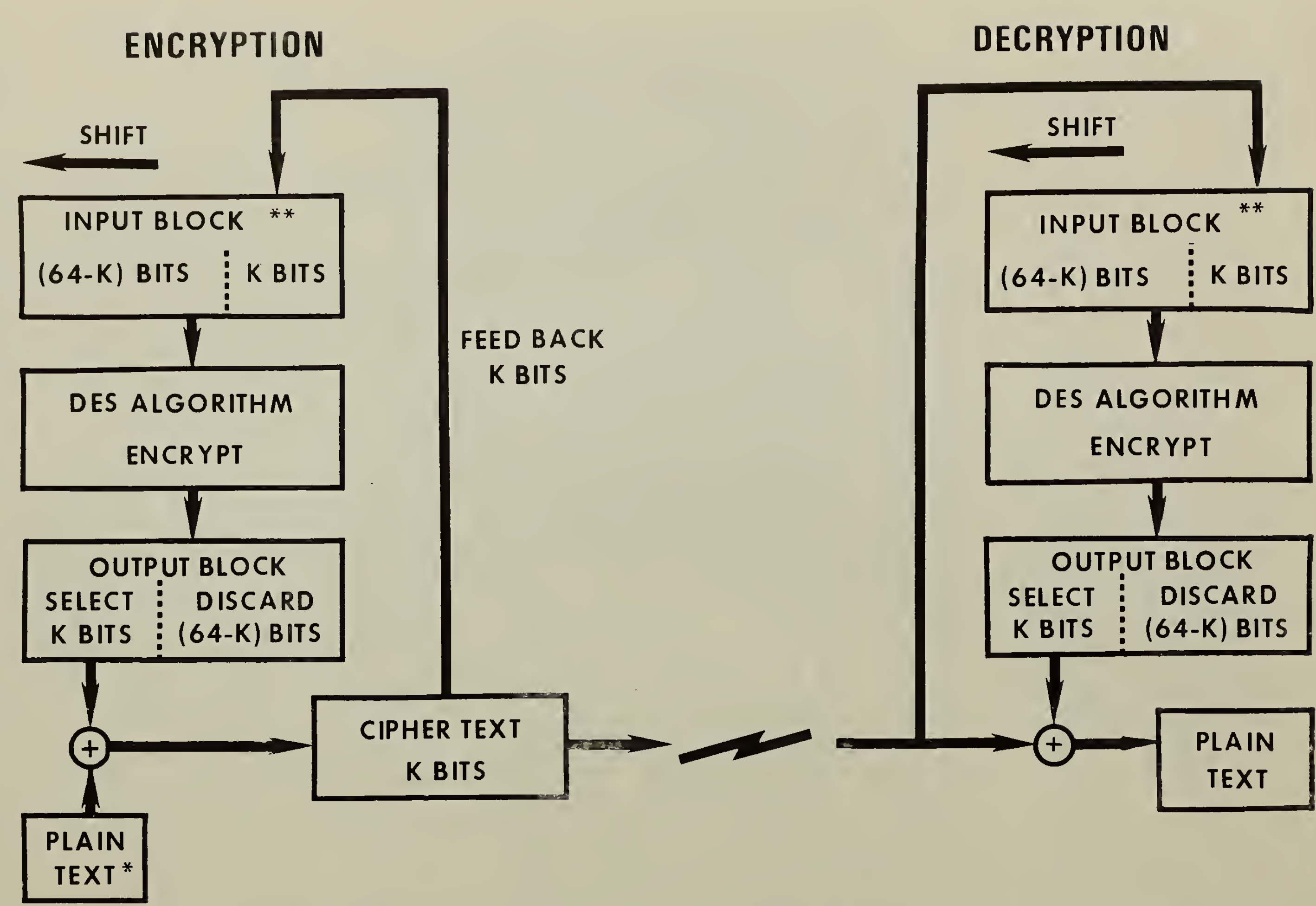
5. CIPHER FEEDBACK (CFB) MODE

The CFB mode is an additive cipher technique in which the DES is used to generate a pseudorandom binary stream. This stream is exclusive-ORed with the binary plain text to form the cipher text. The cipher text is fed back to form the next DES input block. The pseudorandom binary stream is sometimes referred to as a key stream, and the DES is then called a key generator or KG. However, this terminology will not be used in order to avoid confusion with the cryptographic key.

One through sixty-four bit CFB operation may be used. A 64-bit initialization vector (IV) or starter input block is used to begin CFB operations. A CFB IV is placed in the DES input block so that any zero fill is left-justified. This 64-bit input block is encrypted through a DES device producing a 64-bit, pseudorandom output block. The DES device is operated once for each new k-bit $(0 < k < 65)$ character to be encrypted. In all CFB implementations, the leftmost or most significant k bits of the DES output block are used in the exclusive-OR operation. These output block bits, (O1,O2,...,Ok), are exclusive-ORed with the corresponding k bits of data, (D1,D2,...,Dk) to form the cipher text: $(C1,C2,\ldots,Ck) = (D1 \oplus O1, D2 \oplus O2, \ldots, Dk \oplus Ok)$. In order to define this operation when the length of the plain text character to be encrypted is less than k bits, zeros are concatenated to the left hand side or most significant bits of the plain text. Obviously, users must agree on the representation of a plain text "character." Bits (Ok+1, Ok+2, . . . , O64) of the DES output block are discarded. The k bits of cipher text are fed back to the LSB positions of the DES input (I) block such that:

Bits at time t	--->	Bits at (t+1)
I[k+1]	--->	I1
I[k+2]	--->	I2
.	.	.
.	.	.
.	.	.
I64	--->	I[64-k]
C1	--->	I[64-(k-1)]
C2	--->	I[64-(k-2)]
.	.	.
.	.	.
.	.	.
Ck	--->	I64

FIGURE 5: K-BITS CIPHER FEEDBACK (CFB) MODE



* IF PLAIN TEXT CHARACTER IS LESS THAN K BITS IN LENGTH, THEN APPEND ZEROS TO THE LEFT HAND SIDE (MSB) OF THE DATA TO DEFINE THE EXCLUSIVE-OR (⊕) OPERATION. ONLY ONE K-BIT CHARACTER IS PROCESSED WITH EACH OUTPUT BLOCK.

** INPUT BLOCK INITIALLY CONTAINS 64-BIT INITIALIZATION VECTOR (IV). THIS IV IS USED TO GENERATE FIRST OUTPUT BLOCK TO ENCRYPT (DECRYPT) THE FIRST PLAIN (CIPHER) TEXT CHARACTER.

As an example, consider an 8-bit CFB implementation and an IV with forty-eight pseudorandom bits. After each encryption, the eight cipher bits are fed back into the DES input block such that:

AFTER ENCRYPTION #	THE DES INPUT BLOCK CONTAINS:
0	(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,.......,47,48)
1	(0,0,0,0,0,0,0,0,1,2,.......,47,48,C1,C2,......,C8)
2	(1,2,.......,47,48,C1,C2,.....,C8,C9,C10,.....,C16)

After the eighth encryption the entire 64-bit IV will have been shifted out of the DES input block. Let the first character be a seven-bit ASCII character plus parity denoted by (P,b7,b6,...,b1). If the first DES output block is (O1,O2,...,O64), then the first cipher text character will be (C1,C2,...,C7,C8) = (P⊕O1,B7⊕O2,...,b2⊕O7,b1⊕O8). Appendix A contains a detailed example of 8-bit CFB operations and Appendix C has a flow chart diagramming the basic CFB procedures.

The security of a CFB system depends, among other things, upon the management of CFB IVs. Some recommendations in this area are provided in Appendix E.

To protect against undetected bit manipulation of the last k-bit cipher text character using k-bit CFB, the last character to be encrypted must consist of error detection, terminal flag, or permanently fixed data.

In CFB, each cipher bit is a function of one bit of the DES output block and one bit of plain text. Therefore, any bits of a CFB cipher text character may be changed without affecting the decryption of other bits within the same character so long as the bits which are fed back at the encrypting and decrypting stations are the same. An application of this characteristic would be the deletion of encrypted parity bits and their replacement with a new parity computed on the cipher text. This feature is useful in networks which are sensitive to parity checks.

Bit errors within one CFB cipher text character will affect not only the decryption of the garbled cipher text but also the decryption of succeeding characters until the bit errors are flushed out of the CFB input block. The first affected plain text character will be garbled in exactly those places where the cipher text character is in error. Succeeding plain text characters will experience an average error rate of about fifty percent until all errors have been shifted out of the DES input block. Assuming no additional errors are encountered during this time, the CFB decryption device will then automatically regain cryptographic synchronization. This characteristic is sometimes referred to as a limited error extension or a self-synchronizing capability -- depending upon the point of view. In the previous 8-bit CFB example, errors in one cipher text character affect the decryption of nine characters. A detailed example of cipher text errors in CFB operations may be found in Appendix B.

If k-bit character boundaries are lost during decryption, then cryptographic synchronization will be lost until a cryptographic restart (new IV) is initiated or until after character boundaries have been re-established. In the latter case, if 64 (mod k) = 0, then resynchronization will occur automatically after 64/k characters with proper boundary definition have entered the DES input block; otherwise, one additional character will be required.

The encryption and decryption processes in the CFB mode invoke the DES key schedule vectors in the same order, i.e., both processes will use {KS1, KS2, . . . , KS16}. This produces a partial involution under the same IV and key; i.e., if E[K, (P1,P2,...,Pn)] = (C1,C2,...,Cn), then E[K, (C1,C2,...,Cn)] = (P1,N1,N2,...,N(n-1)) where P, C, and N represent k-bit plain text, cipher text, and nonsense (pseudorandom) characters, respectively.

6. A DES AUTHENICATION-ONLY MODE

The DES algorithm may also be used for the authentication of plain text. This technique is useful in applications which require maintaining data integrity but do not require that the plain text be protected from disclosure. The authentication-only mode protects against bit manipulation within the data as well as the insertion and deletion of messages, and the replay of a previously valid message. In the authentication-only mode, two message authentication codes (MACs) are independently computed on the same data -- one at the data source and one at the data destination. This data consists of a unique message identifier (MID) and the text to be protected. If the data source MAC and the data destination MAC are in agreement and if the MID agrees with its expected value, then the plain text is accepted as authentic at the data destination. The authentication-only format is depicted in figure #6. The MAC must be generated using either the CFB or the CBC mode.

FIGURE 6:

AUTHENTICATION-ONLY FORMAT

FIELD 1	FIELD 2	FIELD 3
MID	PLAIN TEXT	MAC

In order to commence CFB authentication operations, a unique message identifier is used as an initialization vector. The plain text is encrypted in the normal CFB manner except that the cipher text is not communicated to the decrypting device. After the encryption of the final plain text unit (character or block), the last cipher text is fed back into the DES input block as if another plain text unit were to be encrypted. Then the DES device is operated one more time and the left-most or most significant k bits ($0 < k < 65$) in the next DES output block are used as the MAC.

To begin CBC authentication operations, the MID is again used as an initialization vector. However, for this application the first plain text block to be encrypted is the all zero block, i.e., the MID alone defines the first DES input block. Thereafter, the plain text is encrypted in the normal CBC manner. The CBC MAC is defined to be the left-most or most significant k bits of the DES output block resulting from encrypting the final plain text block. Messages which terminate in partial data blocks are to be padded on the right (LSB) with zeros.

The MID, plain text, and the MAC are conveyed to the data destination; the intermediate cipher text is not transmitted. The probability that one could randomly select a correct k-bit MAC is 1/(2**k). For most applications, MACs of at least 24 bits are strongly recommended. Two examples of the authentication-only mode may be found in Appendix D.

In general, the MID is a unique and deterministic message identifier within a cryptographic period; the MID should also be varied across cryptographic periods. The value of the MID will be checked by the recipient of an authentication-only message to verify that messages have not been deleted, inserted, or replayed. The uniqueness <u>within</u> a cryptographic period may be achieved through the use of a one-up binary counter. MID values are not to be repeated within the same cryptographic period; this constraint also applies to multiuser environments under control of a common cryptographic key. The MID variation <u>across</u> cryptographic periods may be satisfied by selecting a random or pseudorandom starting value from the total range of a "within" counter. In using this approach it is recommended that only a small (say << 5%) segment of the MID's total range be used within any key period. Another acceptable technique would be to form the MID by concatenating a unique message identifier together with a unique identifier for each cryptographic period, e.g., a second one-up binary counter could be used for this purpose.

A MID is not encrypted. However, whenever MID values are exchanged through an unsecured channel to establish or re-establish MID synchronization, then these values must be protected. This protection includes the detection of bit alteration, the insertion of bogus messages, and the replay or deletion of valid messages.

7. REFERENCES

[FIPS 46] Federal Information Processing Standard Publication 46, Data Encryption Standard. 1977 January 15; FIPS PUB 46: 18 pages. Available from: National Technical Information Services; U.S. Department of Commerce; Springfield, Virginia 22161; NBS-FIPS-PUB-46.

8. DEFINITIONS, ABBREVIATIONS, AND CONVENTIONS

AUTHENTICATION-ONLY: A DES technique for protecting the integrity of plain text which does not have to be protected from disclosure; see section 6.

BLOCK: A binary vector consisting of sixty-four bits numbered from the left as 1, 2, . . . , 64.

CBC: Cipher block chaining; see section 4.

CFB: Cipher feedback; see section 5.

CIPHER TEXT: Encrypted data.

CRYPTOGRAPHIC KEY: The 56 random bits of the cryptographic variable which are used to govern the DES device. Also simply called KEY.

CRYPTOGRAPHIC PERIOD: That period of DES operation during which a unique data-encrypting key is used between two or more cryptographic facilities. Keys from different cryptographic periods are independent. Synonym: Key period.

CRYPTOGRAPHIC VARIABLE: The 64 bit vector containing the 56-bit DES key and its eight associated parity bits. Synonym: Key Variable.

DECRYPTION: The process of changing cipher text into plain text. Verb: DECRYPT.

DES: Data Encryption Standard; specified in [FIPS 46].

DES DEVICE: The hardware used to implement the DES algorithm. This is usually an integrated circuit chip which is sometimes referred to as a "crypto-engine."

DES INPUT BLOCK: A 64-bit data vector that is entered into the DES device.

DES OUTPUT BLOCK: A 64-bit vector that is the final result of a DES device.

ECB: Electronic codebook mode; see section 3.

ENCRYPTION: A process of changing plain text into cipher text. Verb: ENCRYPT.

EXCLUSIVE-OR OPERATION: the bit-by-bit modulo-2 addition without carry of binary numbers. This operation is represented by a circled +.

INITIALIZATION VECTOR (IV): A 64-bit vector used to help form the initial DES input block for the CFB and CBC modes of operation; a 64-bit cryptographic synchronization vector.

KEY: Cryptographic key; the 56 random bits of the cryptographic variable.

KEY SCHEDULE FUNCTION: A logical unit within the DES algorithm which generates a different 48-bit vector from the 64-bit cryptographic variable for each of the sixteen rounds of the DES process.

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

MAC: Message authentication code; see section 6.

MESSAGE (MSG): A generic term used to describe a logical data entity. In general it is an ambiguous term; therefore, for specific applications it should be precisely defined.

MID: A message identifier used with the authentication-only mode.

MOST SIGNIFICANT BIT (MSB): The left-most bit of a binary row vector. Synonym: High order bit.

OCTET: A group of eight binary digits numbered from left to right: 1,2,...,8.

PLAIN TEXT: Decrypted data or data to be encrypted.

PSEUDORANDOM BINARY PROCESS: A deterministic technique for producing a sequence of binary digits which satisfy the statistical properties of a random bit stream.

APPENDIX A

SAMPLE DES ENCRYPTIONS AND DECRYPTIONS

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

DES MODES OF OPERATION

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	DES INPUT BLOCK	DES OUTPUT BLOCK	CIPHER TEXT
1	4e6f772069732074	4e6f772069732074	3d10fae4418fb83d	3d10fae4418fb83d
2	68652074696d6520	68652074696d6520	73b5bc13ffcd86fd	73b5bc13ffcd86fd
3	666f7220616c6c20	666f7220616c6c20	ff8f8fe4583a5a0f	ff8f8fe4583a5a0f

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

The ECB mode in the decrypt state has been selected.

Cryptographic Key = 0123456789abcdef

TIME	CIPHER TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	PLAIN TEXT
1	3d10fae4418fb83d	3d10fae4418fb83d	4e6f772069732074	4e6f772069732074
2	73b5bc13ffcd86fd	73b5bc13ffcd86fd	68652074696d6520	68652074696d6520
3	ff8f8fe4583a5a0f	ff8f8fe4583a5a0f	666f7220616c6c20	666f7220616c6c20

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

DES MODES OF OPERATION

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	DES INPUT BLOCK	DES OUTPUT BLOCK	CIPHER TEXT
1	4e6f772069732074	5c5b2158f9d8ed9b	c785361299fdf8c5	c785361299fdf8c5
2	68652074696d6520	afe01666f0909de5	9bebded075e1096a	9bebded075e1096a
3	666f7220616c6c20	fd84acf0148d654a	d0745b3dd304f9e1	d0745b3dd304f9e1

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

The CBC mode in the decrypt state has been selected.

Cryptographic   Key   = 0123456789abcdef

Initialization Vector = 1234567890abcdef

TIME	CIPHER TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	PLAIN TEXT
1	c785361299fdf8c5	c785361299fdf8c5	5c5b2158f9d8ed9b	4e6f772069732074
2	9bebded075e1096a	9bebded075e1096a	afe01666f0909de5	68652074696d6520
3	d0745b3dd304f9e1	d0745b3dd304f9e1	fd84acf0148d654a	666f7220616c6c20

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

DES MODES OF OPERATION

AN EXAMPLE OF THE EIGHT-BIT CIPHER-TEXT 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	⊕	O	=	C
1	1234567890abcdef	3494ccf2bda243a6	4e	⊕	34	=	7a
2	34567890abcdef7a	bd914d9386658541	6f	⊕	bd	=	d2
3	567890abcdef7ad2	adaf8c45c53f1f92	77	⊕	ad	=	da
4	7890abcdef7ad2da	e3a3acc86c429d97	20	⊕	e3	=	c3
5	90abcdef7ad2dac3	5cb1ebb51b70388d	69	⊕	5c	=	35
6	abcdef7ad2dac335	f3af29ce4cb9661f	73	⊕	f3	=	80
7	cdef7ad2dac33580	aa22a74f73723ad1	20	⊕	aa	=	8a
8	ef7ad2dac335808a	92d94cce188f15af	74	⊕	92	=	e6
9	7ad2dac335808ae6	4fbbea151d3cfc54	68	⊕	4f	=	27
10	d2dac335808ae627	6dabad1fdba8bc3b	65	⊕	6d	=	08

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

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

Cryptographic Key = 0123456789abcdef

Initialization Vector = 1234567890abcdef

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	C	⊕	O	=	P
1	1234567890abcdef	3494ccf2bda243a6	7a	⊕	34	=	4e
2	34567890abcdef7a	bd914d9386658541	d2	⊕	bd	=	6f
3	567890abcdef7ad2	adaf8c45c53f1f92	da	⊕	ad	=	77
4	7890abcdef7ad2da	e3a3acc86c429d97	c3	⊕	e3	=	20
5	90abcdef7ad2dac3	5cb1ebb51b70388d	35	⊕	5c	=	69
6	abcdef7ad2dac335	f3af29ce4cb9661f	80	⊕	f3	=	73
7	cdef7ad2dac33580	aa22a74f73723ad1	8a	⊕	aa	=	20
8	ef7ad2dac335808a	92d94cce188f15af	e6	⊕	92	=	74
9	7ad2dac335808ae6	4fbbea151d3cfc54	27	⊕	4f	=	68
10	d2dac335808ae627	6dabad1fdba8bc3b	08	⊕	6d	=	65

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

DES MODES OF OPERATION

AN EXAMPLE OF THE ONE-BIT CIPHER-TEXT 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 ASCII code for "Now."
These seven-bit characters are written in binary as (0,b7,b6,...,b1). The DES input and output blocks are written in hexadecimal notation. The ⊕ represents bit-by-bit, modulo-2 addition.
Note: A draft, ANSI X3S3.8, link encryption standard requires that the bits of an ASCII character be processed in the reverse order of this example for 1-bit CFB. This draft data communications standard is designed for the transmission of ASCII characters in the standard, serial-by-bit manner (b1,b2,...,b7).

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P	⊕	O	= C
1	1234567890abcdef	3494ccf2bda243a6	0	⊕	0	= 0
2	2468acf121579bde	0e3c58d16a551826	1	⊕	0	= 1
3	48d159e242af37bd	87647b405d0b8cdf	0	⊕	1	= 1
4	91a2b3c4855e6f7b	e91173c434165ae0	0	⊕	1	= 1
5	234567890abcdef7	ca46c7d8c3aab695	1	⊕	1	= 0
6	468acf121579bdee	c3c832f66747157d	1	⊕	1	= 0
7	8d159e242af37bdc	7dd595cc10590044	1	⊕	0	= 1
8	1a2b3c4855e6f7b9	6fb8b7e872332de4	0	⊕	0	= 0
9	34567890abcdef72	9c91cbdde38c8104	0	⊕	1	= 1
10	68acf121579bdee5	fc4307d1d4e0b047	1	⊕	1	= 0
11	d159e242af37bdca	5fb5a8881bc850dd	1	⊕	0	= 1
12	a2b3c4855e6f7b95	57bba20ef9c025bd	0	⊕	0	= 0
13	4567890abcdef72a	5ea14c224e01dc2c	1	⊕	0	= 1
14	8acf121579bdee55	8e869e2277f5a1ad	1	⊕	1	= 0
15	159e242af37bdcaa	f68135d6b7585f91	1	⊕	1	= 0
16	2b3c4855e6f7b954	dc6d28bbb9061212	1	⊕	1	= 0
17	567890abcdef72a8	3d959a3e4d11eb9b	0	⊕	0	= 0
18	acf121579bdee550	32747b8355a832bf	1	⊕	0	= 1
19	59e242af37bdcaa1	eb5598e0a949d791	1	⊕	1	= 0
20	b3c4855e6f7b9542	e6308121b905adbd	1	⊕	1	= 0
21	67890abcdef72a84	f7998d0e85a6661b	0	⊕	1	= 1
22	cf121579bdee5509	1ccf6e6bef3d34e5	1	⊕	0	= 1
23	9e242af37bdcaa13	1c7da296b765e487	1	⊕	0	= 1
24	3c4855e6f7b95427	bbfdbec304117d48	1	⊕	1	= 0

APPENDIX B

EFFECTS OF CIPHER TEXT ERRORS

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>

DES MODES OF OPERATION

CIPHER TEXT ERRORS AND THE ECB MODE

The ECB mode in the decrypt state has been selected.

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).
The last ASCII character in the first cipher text block has been changed to "3e" instead of the correct value "3d."
The first block of cipher text should have decrypted to "4e6f772069732074."
Bit error(s) within an ECB cipher text block will cause each bit of the corresponding plain text block to be in error with a probability of approximately fifty percent. Thus thirty-two errors are expected in the first block of decrypted plain text. For this specific example there are exactly 31 errors in this block. Note that bit errors within one ECB cipher text block do not affect the decryption of other cipher text blocks.
Asterisks are used to identify the blocks affected by the cipher text errors.

Cryptographic Key = 0123456789abcdef

TIME	CIPHER TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	PLAIN TEXT
1	3d10fae4418fb83e*	3d10fae4418fb83e*	380e655486fb2d08*	380e655486fb2d08*
2	73b5bc13ffcd86fd	73b5bc13ffcd86fd	68652074696d6520	68652074696d6520
3	ff8f8fe4583a5a0f	ff8f8fe4583a5a0f	666f7220616c6c20	666f7220616c6c20

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

DES MODES OF OPERATION

CIPHER TEXT ERRORS AND THE CBC MODE

The CBC mode in the decrypt state has been selected.

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). The last ASCII character in the first cipher text block has been changed to "c6" instead of the correct value "c5." The first plain text block should have been "4e6f772069732074." Note that these bit errors affect the decryption of two blocks before cryptographic synchronization is re-established. Each bit in the first plain text block will be in error with a probability of approximately fifty percent; thus 32 binary errors are expected in this block. In this specific example there are exactly 30 errors in the first plain text block. The second block will have only those bits in error which correspond directly in number and position to the garbled cipher text bits. In this example the last two bits of plain text block #2 are incorrect. The third and succeeding plain text blocks are unaffected by bit errors in the first cipher text block. Asterisks are used to identify the blocks which are affected by the cipher text errors.

Cryptographic Key = 0123456789abcdef

Initialization Vector = 1234567890abcdef

TIME	CIPHER TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	PLAIN TEXT
1	c785361299fdf8c6*	c785361299fdf8c6*	4d91d5742e497411*	5fa5830cbee2b9fe*
2	9bebded075e1096a	9bebded075e1096a	afe01666f0909de5	68652074696d6523*
3	d0745b3dd304f9e1	d0745b3dd304f9e1	fd84acf0148d654a	666f7220616c6c20

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

DES MODES OF OPERATION

CIPHER TEXT ERRORS AND THE CFB MODE

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

The plain text is the ASCII code for "Now is the time for." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...,b1). The cipher text character at time #4 has been changed to "c4" instead of the correct value "c3." In the general k-bit CFB mode, bit errors within a k-bit character will affect the decryption of one character in those bit positions corresponding to the errors. Succeeding characters will experience an error probability of approximately fifty percent until the garbles have been flushed from the DES input block. In 8-bit CFB, errors within a character affect the decryption of nine cipher text characters before cryptographic synchronization is re-established.
The ⊕ represents bit-by-bit, modulo-2 addition.

Cryptographic Key = 0123456789abcdef
Initialization Vector = 1234567890abcdef

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	C	⊕	O	= P
1	1234567890abcdef	3494ccf2bda243a6	7a	⊕	34	= 4e
2	34567890abcdef7a	bd914d9386658541	d2	⊕	bd	= 6f
3	567890abcdef7ad2	adaf8c45c53f1f92	da	⊕	ad	= 77
		ERROR IN NEXT CIPHER				
4	7890abcdef7ad2da	e3a3acc86c429d97	c4	⊕	e3	= 27
5	90abcdef7ad2dac4	23ae7d3f645f7daa	35	⊕	23	= 16
6	abcdef7ad2dac435	aca8467513ba93ae	80	⊕	ac	= 2c
7	cdef7ad2dac43580	383b374dc55ba511	8a	⊕	38	= b2
8	ef7ad2dac435808a	fbebef188f6ceab0	e6	⊕	fb	= 1d
9	7ad2dac435808ae6	307a55324bc7d098	27	⊕	30	= 17
10	d2dac435808ae627	a3179850047a27f9	08	⊕	a3	= ab
11	dac435808ae62708	0d0eaf64dc7e5ccb	0f	⊕	0d	= 02
12	c435808ae627080f	5cc2582fc7ca06f3	6d	⊕	5c	= 31
		AUTORESYNC ON NEXT CIPHER				
13	35808ae627080f6d	953d fb12d8a4cf4	fc	⊕	95	= 69
14	808ae627080f6dfc	dc365b7e426e726e	b1	⊕	dc	= 6d
15	8ae627080f6dfcb1	57cb75f386e76662	32	⊕	57	= 65
16	e627080f6dfcb132	68bbd6da037c3eb1	48	⊕	68	= 20
17	27080f6dfcb13248	71e50a54bd8c1f1d	17	⊕	71	= 66
18	080f6dfcb1324817	b91c8fd54d1d45f5	d6	⊕	b9	= 6f
19	0f6dfcb1324817d6	009b705bb2ef50d6	72	⊕	00	= 72

APPENDIX C

CBC AND CFB FLOW CHARTS
(FOR ILLUSTRATIVE PURPOSES ONLY)

FIGURE C.1: CIPHER BLOCK CHAINING (CBC) OPERATIONS

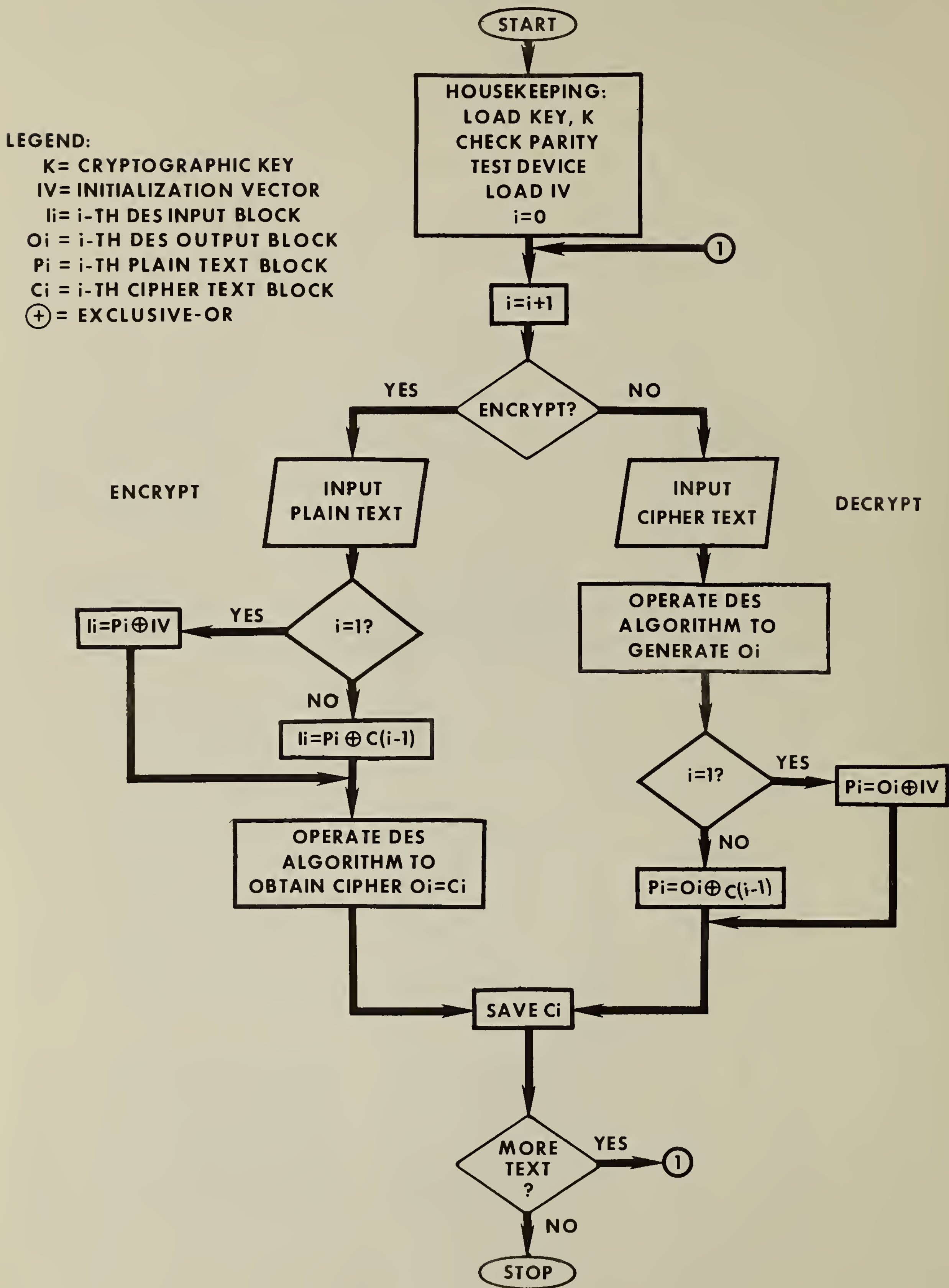
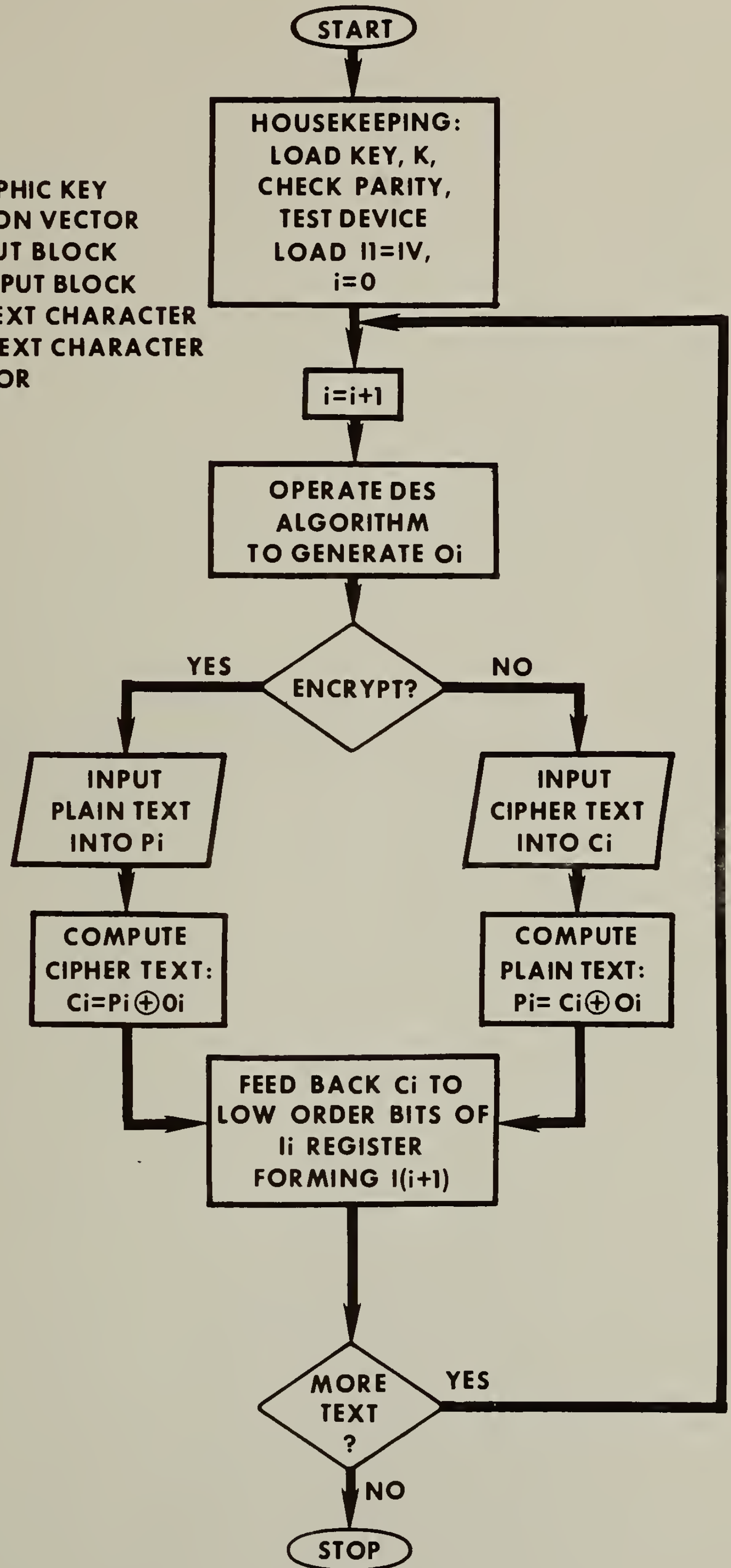


FIGURE C.2: CIPHER FEEDBACK (CFB) OPERATIONS

LEGEND:
K= CRYPTOGRAPHIC KEY
IV= INITIALIZATION VECTOR
Ii= i-TH DES INPUT BLOCK
Oi= i-TH DES OUTPUT BLOCK
Pi= i-TH PLAIN TEXT CHARACTER
Ci= i-TH CIPHER TEXT CHARACTER
⊕= EXCLUSIVE-OR



APPENDIX D

EXAMPLES OF AUTHENTICATION-ONLY MODE

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

DES MODES OF OPERATION

AN EXAMPLE OF THE AUTHENTICATION-ONLY MODE USING CBC

The CBC mode in the encrypt state has been selected.

For this example, the MID structure is a 48-bit binary counter which has been initialized to a pseudorandom value concatenated with sixteen zeros. Note that the first "plain text" block is all zeros; this is done to force the first DES input block to be equal to the MID in the CBC authentication mode. The last plain text block is padded to sixty-four bits with zeros on its right. The MID and the plain text are communicated unencrypted along with the MAC to the data destination. For this example, a 32-bit MAC has been selected. The plain text is the ASCII code for "Now is the time for ." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...,b1).

Cryptographic Key = 0123456789abcdef

Initialization Vector = 00006ac103b28f99

TIME	PLAIN TEXT	DES INPUT BLOCK	DES OUTPUT BLOCK	
1	0000000000000000	00006ac103b28f99	efd5218cc2439930	
2	4e6f772069732074	a1ba56acab30b944	40f0a1601d7547c9	
3	68652074696d6520	28958114741822e9	b50e378f6058b74a	
4	666f722000000000	d36145af6058b74a	7ab019e4dd481bca	=> MAC = 7ab019e4

MESSAGE TEXT:
MID TEXT MAC
00006ac103b28f994e6f77206973207468652074696d6520666f72207ab019e4

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

AN EXAMPLE OF THE AUTHENTICATION-ONLY MODE USING 8-BIT CFB

The 8-bit CFB mode in the encrypt state has been selected. For this example the MID structure consists of a 32-bit date-time code concatenated with a 32-bit, one-up binary counter. The date-time code designates the year (0), day of the year (003), hour of the day (10 or 1000 hours), and minutes (15). This value is used to uniquely identify the cryptographic period. The binary counter has been initialized to 00000001; it is incremented by "1" for each message to be authenticated within the same DES key period. A 24-bit MAC was selected for this example.
The plain text is the ASCII code for "Now is the time for." These seven-bit characters are written in hexadecimal notation (0,b7,b6,...,b1). The ⊕ represents bit-by-bit, modulo-2 addition.

Cryptographic Key = 0123456789abcdef
Initialization Vector = 0003101500000001

TIME	DES INPUT BLOCK	DES OUTPUT BLOCK	P ⊕ O = C
1	0003101500000001	973c7b54e5bdd4da	4e ⊕ 97 = d9
2	03101500000001d9	bde39e43bf3f2258	6f ⊕ bd = d2
3	101500000001d9d2	1998d46fe4175d64	77 ⊕ 19 = 6e
4	1500000001d9d26e	5ddf40f5124917fa	20 ⊕ 5d = 7d
5	00000001d9d26e7d	09342c9870b4fe85	69 ⊕ 09 = 60
6	000001d9d26e7d60	ffedd60c4e72814e	73 ⊕ ff = 8c
7	0001d9d26e7d608c	3f873ff6e2e4cd39	20 ⊕ 3f = 1f
8	01d9d26e7d608c1f	724f0467a5fffdeb	74 ⊕ 72 = 06
9	d9d26e7d608c1f06	9ec1876e54493a52	68 ⊕ 9e = f6
10	d26e7d608c1f06f6	01a5d888c472bcdd	65 ⊕ 01 = 64
11	6e7d608c1f06f664	0b6f90e412246e54	20 ⊕ 0b = 2b
12	7d608c1f06f6642b	992db95a018d921f	74 ⊕ 99 = ed
13	608c1f06f6642bed	95c9b701a4681991	69 ⊕ 95 = fc
14	8c1f06f6642bedfc	b2e4f4318882ca59	6d ⊕ b2 = df
15	1f06f6642bedfcdf	6dfff2d406ea81de	65 ⊕ 6d = 08
16	06f6642bedfcdf08	43148a850bd66dc8	20 ⊕ 43 = 63
17	f6642bedfcdf0863	a26e7ae5815d7b60	66 ⊕ a2 = c4
18	642bedfcdf0863c4	57d8cc8f837534b9	6f ⊕ 57 = 38
19	2bedfcdf0863c438	560cccbc949bb350	72 ⊕ 56 = 24
20	edfcdf0863c43824	63113f72e8a09991	=> MAC = 63113f

MESSAGE FORMAT:
MID TEXT MAC
00031015000000014e6f77206973207468652074696d6520666f7263113f

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

APPENDIX E

RECOMMENDATIONS FOR DES IV MANAGEMENT

The security of a CFB or CBC implementation depends, inter alia, upon cryptographic synchronization procedures. For these modes of operation, this means proper management of the DES initialization vectors (IVs). IV management encompasses the generation, distribution, protection, usage, and disposal of initialization vectors.

At least one standard (*) currently under development is specifying security requirements for CFB and CBC IVs in telecommunication environments. It is conceivable that a special application standard may need to tailor IV management to fit a particular set of application requirements in order to make the standard reasonably efficient as well as effective.

This standard on the modes of operation does not mandate specific techniques in IV management. This standard specifies only those requirements which are essential for unambiguously describing the mechanics needed to implement the modes of operation.

The following suggestions are recommended as preliminary guidelines which may be used until the publication of official guideline(s) or standard(s) in the area of IV management.

<u>CBC IVs</u>: The CBC IV consists of sixty-four binary digits. A single IV may be used throughout an entire CBC cryptographic period, however, this IV should be protected from disclosure. CBC IVs should not be repeated across cryptographic periods (same key) with a probability greater than 2 to the (-64). A 64-bit random or pseudorandom generation technique will satisfy this characteristic.

<u>CFB IVs</u>: CFB IVs consist of 64 binary digits. As with other DES additive stream ciphers, it is desirable that CFB IVs change as frequently as possible in order to insure a unique additive stream to protect the plain text. CFB IVs should not repeat within a specific cryptographic period or across different cryptographic periods with a probability greater than 2 to the (-48). As a corollary, CFB IVs may only contain a maximum of sixteen fixed bits. A 48-bit random or pseudorandom process is sufficient to satisfy this property. CFB IVs do not need to be protected from disclosure, i.e., they may be transmitted unencrypted through an unsecured channel.

----

* Proposed Federal Standard 1027; Telecommunications: Security Requirements for Use of the Data Encryption Standard; 4 September 1979, 18 pages.

-NOTES-

-NOTES-