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

FIPS

FIPS PUB 46-2

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION
(Supersedes FIPS PUB 46-1 – 1988 January 22)

DATA ENCRYPTION STANDARD (DES)

CATEGORY: COMPUTER SECURITY   SUBCATEGORY: CRYPTOGRAPHY

Reaffirmed
1993 December 30

FIPS PUB 46-2

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION
(Supersedes FIPS PUB 46-1 – 1988 January 22)

DATA ENCRYPTION STANDARD (DES)

CATEGORY: COMPUTER SECURITY   SUBCATEGORY: CRYPTOGRAPHY

Foreword

The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST) is the official publication relating to standards and guidelines adopted and promulgated under the provisions of Section 111(d) of the Federal Property and Administrative Services Act of 1949 as amended by the Computer Security Act of 1987, Public Law 100-235. These mandates have given the Secretary of Commerce and NIST important responsibilities for improving the utilization and management of computer and related telecommunications systems in the Federal Government. The NIST, through its Computer Systems Laboratory, provides leadership, technical guidance, and coordination of Government efforts in the development of standards and guidelines in these areas.

Comments concerning Federal Information Processing Standards Publications are welcomed and should be addressed to the Director, Computer Systems Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899.

James H. Burrows, Director
Computer Systems Laboratory

Abstract

The selective application of technological and related procedural safeguards is an important responsibility of every Federal organization in providing adequate security to its electronic data systems. This publication specifies a cryptographic algorithm which may be used by Federal organizations to protect sensitive data. Protection of data during transmission or while in storage may be necessary to maintain the confidentiality and integrity of the information represented by the data. The algorithm uniquely defines the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form. The Data Encryption Standard is being made available for use by Federal agencies within the context of a total security program consisting of physical security procedures, good information management practices, and computer system/network access controls. This revision supersedes FIPS 46-1 in its entirety.

Key words: computer security; data encryption standard; encryption; Federal Information Processing Standard (FIPS); security.

Federal Information
Processing Standards Publication 46-2

1993 December 30

Announcing the

DATA ENCRYPTION STANDARD (DES)

Federal Information Processing Standards Publications (FIPS PUBS) are issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce pursuant to Section 111(d) of the Federal Property and Administrative Services Act of 1949 as amended by the Computer Security Act of 1987, Public Law 100-235.

1. Name of Standard. Data Encryption Standard (DES).

2. Category of Standard. Computer Security, Cryptography.

3. Explanation. The Data Encryption Standard (DES) specifies a FIPS approved cryptographic algorithm as required by FIPS 140-1. This publication provides a complete description of a mathematical algorithm for encrypting (enciphering) and decrypting (deciphering) binary coded information. Encrypting data converts it to an unintelligible form called cipher. Decrypting cipher converts the data back to its original form called plaintext. The algorithm described in this standard specifies both enciphering and deciphering operations which are based on a binary number called a key.

A key consists of 64 binary digits ("O"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, are used for error detection. The 8 error detecting bits are set to make the parity of each 8-bit byte of the key odd, i.e., there is an odd number of "1"s in each 8-bit byte.[1] Authorized users of encrypted computer data must have the key that was used to encipher the data in order to decrypt it. The encryption algorithm specified in this standard is commonly known among those using the standard. The unique key chosen for use in a particular application makes the results of encrypting data using the algorithm unique. Selection of a different key causes the cipher that is produced for any given set of inputs to be different. The cryptographic security of the data depends on the security provided for the key used to encipher and decipher the data.

Data can be recovered from cipher only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original data. A standard algorithm based on a secure key thus provides a basis for exchanging encrypted computer data by issuing the key used to encipher it to those authorized to have the data.

Data that is considered sensitive by the responsible authority, data that has a high value, or data that represents a high value should be cryptographically protected if it is vulnerable to unauthorized disclosure or undetected modification during transmission or while in storage. A risk analysis should be performed under the direction of a responsible authority to determine potential threats. The costs of providing cryptographic protection using this standard as well as alternative methods of providing this protection and their respective costs should be projected. A responsible authority then should make a decision, based on these analyses, whether or not to use cryptographic protection and this standard.

---

[1] Sometimes keys are generated in an encrypted form. A random 64-bit number is generated and defined to be the cipher formed by the encryption of a key using a key encrypting key. In this case the parity bits of the encrypted key cannot be set until after the key is decrypted.

4. Approving Authority. Secretary of Commerce.

5. Maintenance Agency. U.S. Department of Commerce, National Institute of Standards and Technology, Computer Systems Laboratory.

6. Applicability. This standard may be used by Federal departments and agencies when the following conditions apply:

1. An authorized official or manager responsible for data security or the security of any computer system decides that cryptographic protection is required; and

2. The data is not classified according to the National Security Act of 1947, as amended, or the Atomic Energy Act of 1954, as amended.

Federal agencies or departments which use cryptographic devices for protecting data classified according to either of these acts can use those devices for protecting unclassified data in lieu of the standard.

Other FIPS approved cryptographic algorithms may be used in addition to, or in lieu of, this standard when implemented in accordance with FIPS 140-1.

In addition, this standard may be adopted and used by non-Federal Government organizations. Such use is encouraged when it provides the desired security for commercial and private organizations.

7. Applications. Data encryption (cryptography) is utilized in various applications and environments. The specific utilization of encryption and the implementation of the DES will be based on many factors particular to the computer system and its associated components. In general, cryptography is used to protect data while it is being communicated between two points or while it is stored in a medium vulnerable to physical theft. Communication security provides protection to data by enciphering it at the transmitting point and deciphering it at the receiving point. File security provides protection to data by enciphering it when it is recorded on a storage medium and deciphering it when it is read back from the storage medium. In the first case, the key must be available at the transmitter and receiver simultaneously during communication. In the second case, the key must be maintained and accessible for the duration of the storage period. FIPS 171 provides approved methods for managing the keys used by the algorithm specified in this standard.

8. Implementations. Cryptographic modules which implement this standard shall conform to the requirements of FIPS 140-1. The algorithm specified in this standard may be implemented in software, firmware, hardware, or any combination thereof. The specific implementation may depend on several factors such as the application, the environment, the technology used, etc. Implementations which may comply with this standard include electronic devices (e.g., VLSI chip packages), micro-processors using Read Only Memory (ROM), Programmable Read Only Memory (PROM), or Electronically Erasable Read Only Memory (EEROM), and mainframe computers using Random Access Memory (RAM). When the algorithm is implemented in software or firmware, the processor on which the algorithm runs must be specified as part of the validation process. Implementations of the algorithm which are tested and validated by NIST will be considered as complying with the standard. Note that FIPS 140-1 places additional requirements on cryptographic modules for Government use. Information about devices that have been validated and procedures for testing and validating equipment for conformance with this standard and FIPS 140-1 are available from the National Institute of Standards and Technology, Computer Systems Laboratory, Gaithersburg, MD 20899.

9. 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 120 through 128. Some exports of cryptographic modules implementing this standard and technical data regarding them must comply with these Federal regulations and be licensed by the U.S. Department of State. Other exports of cryptographic modules implementing this standard and technical data regarding them fall under the

licensing authority of the Bureau of Export Administration of the U.S. Department of Commerce. The Department of Commerce is responsible for licensing cryptographic devices used for authentication, access control, proprietary software, automatic teller machines (ATMs), and certain devices used in other equipment and software. For advice concerning which agency has licensing authority for a particular cryptographic device, please contact the respective agencies.

10. Patents.   Cryptographic devices implementing this standard may be covered by U.S. and foreign patents issued to the International Business Machines Corporation. However, IBM has granted nonexclusive, royalty-free licenses under the patents to make, use and sell apparatus which complies with the standard. The terms, conditions and scope of the licenses are set out in notices published in the May 13, 1975 and August 31, 1976 issues of the Official Gazette of the United States Patent and Trademark Office (934 O.G. 452 and 949 O.G. 1717).

11. Alternative Modes of Using the DES.   FIPS PUB 81, DES Modes of Operation, describes four different modes for using the algorithm described in this standard. These four modes are called the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CFB) mode, and the Output Feedback (OFB) mode. 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 pseudorandom outputs which are combined with the plaintext to produce cipher, thereby chaining together the resulting cipher; OFB is identical to CFB except that the previous output of the DES is used as input in OFB while the previous cipher is used as input in CFB. OFB does not chain the cipher.

12. Implementation of this Standard.   This standard became effective July 1977. It was reaffirmed in 1983, 1988, and 1993. It applies to all Federal agencies, contractors of Federal agencies, or other organizations that process information (using a computer or telecommunications system) on behalf of the Federal Government to accomplish a Federal function. Each Federal agency or department may issue internal directives for the use of this standard by their operating units based on their data security requirement determinations. FIPS 46-2 which revises the implementation of the Data Encryption Algorithm to include software, firmware, hardware, or any combination thereof, is effective June 30, 1994. This revised standard may be used in the interim period before the effective date.

NIST provides technical assistance to Federal agencies in implementing data encryption through the issuance of guidelines and through individual reimbursable projects. The National Security Agency assists Federal departments and agencies in communications security for classified applications and in determining specific security requirements. Instructions and regulations for procuring data processing equipment utilizing this standard are included in the Federal Information Resources Management Regulation (FIRMR) Subpart 201-8.111-1.

13. Specifications.   Federal Information Processing Standard (FIPS) 46-2, Data Encryption Standard (DES) (affixed).

14. Cross Index.

    a.   Federal Information Resources Management Regulations (FIRMR) subpart 201.20.303, Standards, and subpart 201.39.1002, Federal Standards.

    b.   FIPS PUB 31, Guidelines to ADP Physical Security and Risk Management.

    c.   FIPS PUB 41, Computer Security Guidelines for Implementing the Privacy Act of 1974.

    d.   FIPS PUB 65, Guideline for Automatic Data Processing Risk Analysis.

    e.   FIPS PUB 73, Guidelines for Security of Computer Applications.

    f.   FIPS PUB 74, Guidelines for Implementing and Using the NBS Data Encryption Standard.

    g.   FIPS PUB 81, DES Modes of Operation.

    h.   FIPS PUB 87, Guidelines for ADP Contingency Planning.

    i.   FIPS PUB 112, Password Usage.

j. FIPS PUB 113, Computer Data Authentication.

k. FIPS PUB 140-1, Security Requirements for Cryptographic Modules.

l. FIPS PUB 171, Key Management Using ANSI X9.17.

m. Other FIPS and Federal Standards are applicable to the implementation and use of this standard.

In particular, the Code for Information Interchange, Its Representations, Subsets, and Extensions (FIPS PUB 1-2) and other related data storage media or data communications standards should be used in conjunction with this standard. A list of currently approved FIPS may be obtained from the National Institute of Standards and Technology, Computer Systems Laboratory, Gaithersburg, MD 20899.

15. Qualifications. The cryptographic algorithm specified in this standard transforms a 64-bit binary value into a unique 64-bit binary value based on a 56-bit variable. If the complete 64-bit input is used (i.e., none of the input bits should be predetermined from block to block) and if the 56-bit variable is randomly chosen, no technique other than trying all possible keys using known input and output for the DES will guarantee finding the chosen key. As there are over 70,000,000,000,000,000 (seventy quadrillion) possible keys of 56 bits, the feasibility of deriving a particular key in this way is extremely unlikely in typical threat environments. Moreover, if the key is changed frequently, the risk of this event is greatly diminished. However, users should be aware that it is theoretically possible to derive the key in fewer trials (with a correspondingly lower probability of success depending on the number of keys tried) and should be cautioned to change the key as often as practical. Users must change the key and provide it a high level of protection in order to minimize the potential risks of its unauthorized computation or acquisition. The feasibility of computing the correct key may change with advances in technology. A more complete description of the strength of this algorithm against various threats is contained in FIPS PUB 74, Guidelines for Implementing and Using the NBS Data Encryption Standard.

When correctly implemented and properly used, this standard will provide a high level of cryptographic protection to computer data. NIST, supported by the technical assistance of Government agencies responsible for communication security, has determined that the algorithm specified in this standard will provide a high level of protection for a time period beyond the normal life cycle of its associated equipment. The protection provided by this algorithm against potential new threats will be reviewed within 5 years to assess its adequacy (See Special Information Section). In addition, both the standard and possible threats reducing the security provided through the use of this standard will undergo continual review by NIST and other cognizant Federal organizations. The new technology available at that time will be evaluated to determine its impact on the standard. In addition, the awareness of any breakthrough in technology or any mathematical weakness of the algorithm will cause NIST to reevaluate this standard and provide necessary revisions.

At the next review (1998), the algorithm specified in this standard will be over twenty years old. NIST will consider alternatives which offer a higher level of security. One of these alternatives may be proposed as a replacement standard at the 1998 review.

16. Comments. Comments and suggestions regarding this standard and its use are welcomed and should be addressed to the National Institute of Standards and Technology, Attn: Director, Computer Systems Laboratory, Gaithersburg, MD 20899.

17. Waiver Procedure. Under certain exceptional circumstances, the heads of Federal departments and agencies may approve waivers to Federal Information Processing Standards (FIPS). The head of such agency may redelegate such authority only to a senior official designated pursuant to section 3506(b) of Title 44, United States Code. Waiver shall be granted only when:

a. Compliance with a standard would adversely affect the accomplishment of the mission of an operator of a Federal computer system; or

b. Compliance with a standard would cause a major adverse financial impact on the operator which is not offset by Government-wide savings.

Agency heads may act upon a written waiver request containing the information detailed above. Agency heads may also act without a written waiver request when they determine that conditions for meeting the standard cannot be met. Agency heads may approve waivers only by a written decision which explains the basis on which the agency head made the required finding(s). A copy of each decision, with procurement sensitive or classified portions clearly identified, shall be sent to: National Institute of Standards and Technology; ATTN: FIPS Waiver Decisions, Technology Building, Room B-154, Gaithersburg, MD 20899.

In addition, notice of each waiver granted and each delegation of authority to approve waivers shall be sent promptly to the Committee on Government Operations of the House of Representatives and the Committee on Governmental Affairs of the Senate and shall be published promptly in the *Federal Register*.

When the determination on a waiver applies to the procurement of equipment and/or services, a notice of the waiver determination must be published in the *Commerce Business Daily* as a part of the notice of solicitation for offers of an acquisition or, if the waiver determination is made after that notice is published, by amendment to such notice.

A copy of the waiver, any supporting documents, the document approving the waiver and any accompanying documents, with such deletions as the agency is authorized and decides to make under 5 United States Code Section 552(b), shall be part of the procurement documentation and retained by the agency.

18. Special Information. In accordance with the Qualifications Section of this standard, reviews of this standard have been conducted every 5 years since its adoption in 1977. The standard was reaffirmed during each of those reviews. This revision to the text of the standard contains changes which allow software implementations of the algorithm and which permit the use of other FIPS approved cryptographic algorithms.

19. Where to Obtain Copies of the Standard. 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 46-2 (FIPSPUB46-2), and identify the title. When microfiche is desired, this should be specified. Prices are published by NTIS in current catalogs and other issuances. Payment may be made by check, money order, deposit account or charged to a credit card accepted by NTIS.

**Federal Information
Processing Standards Publication 46-2**

1993 December 30

Specifications for the

DATA ENCRYPTION STANDARD (DES)

The Data Encryption Standard (DES) shall consist of the following Data Encryption Algorithm to be implemented in special purpose electronic devices. These devices shall be designed in such a way that they may be used in a computer system or network to provide cryptographic protection to binary coded data. The method of implementation will depend on the application and environment. The devices shall be implemented in such a way that they may be tested and validated as accurately performing the transformations specified in the following algorithm.

DATA ENCRYPTION ALGORITHM

Introduction

The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key.*Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP^{-1}. The key-dependent computation can be simply defined in terms of a function f, called the cipher function, and a function KS, called the key schedule. A description of the computation is given first, along with details as to how the algorithm is used for encipherment. Next, the use of the algorithm for decipherment is described. Finally, a definition of the cipher function f is given in terms of primitive functions which are called the selection functions S_i and the permutation function P. S_i, P and KS of the algorithm are contained in the Appendix.

The following notation is convenient: Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R. Since concatenation is associative $B_1 B_2 \ldots B_8$, for example, denotes the block consisting of the bits of B_1 followed by the bits of $B_2 \ldots$ followed by the bits of B_8.

Enciphering

A sketch of the enciphering computation is given in figure 1.

---

*Blocks are composed of bits numbered from left to right, i.e., the left most bit of a block is bit one.
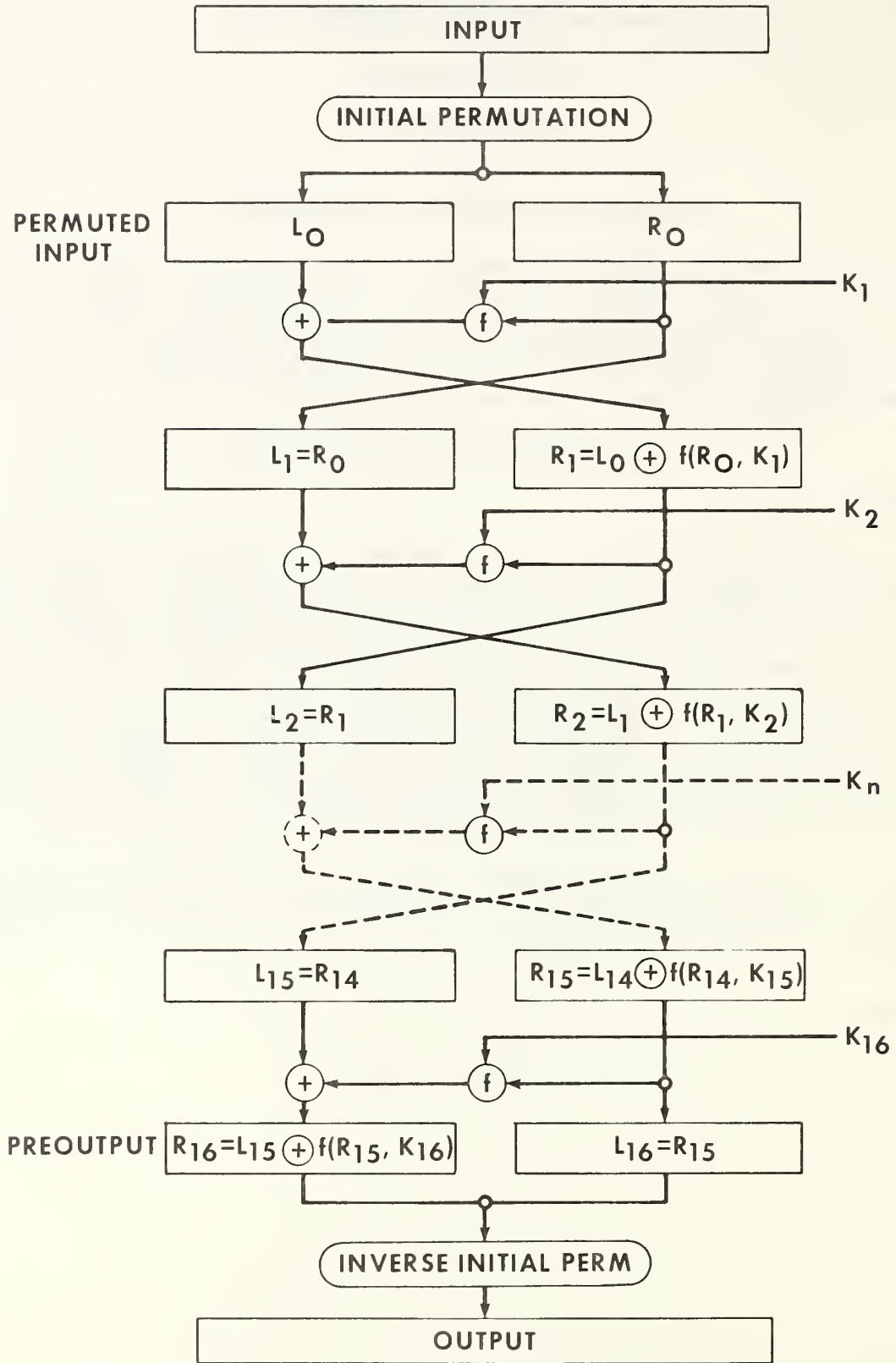
FIGURE 1. *Enciphering computation.*

8

The 64 bits of the input block to be enciphered are first subjected to the following permutation, called the initial permutation *IP*:

$$IP$$

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

That is the permuted input has bit 58 of the input as its first bit, bit 50 as its second bit, and so on with bit 7 as its last bit. The permuted input block is then the input to a complex key-dependent computation described below. The output of that computation, called the preoutput, is then subjected to the following permutation which is the inverse of the initial permutation:

$$IP^{-1}$$

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

That is, the output of the algorithm has bit 40 of the preoutput block as its first bit, bit 8 as its second bit, and so on, until bit 25 of the preoutput block is the last bit of the output.

The computation which uses the permuted input block as its input to produce the preoutput block consists, but for a final interchange of blocks, of 16 iterations of a calculation that is described below in terms of the cipher function f which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits.

Let the 64 bits of the input block to an iteration consist of a 32 bit block L followed by a 32 bit block R. Using the notation defined in the introduction, the input block is then LR.

Let K be a block of 48 bits chosen from the 64-bit key. Then the output $L'R'$ of an iteration with input LR is defined by:

(1)
$$L' = R$$
$$R' = L \oplus f(R,K)$$

where $\oplus$ denotes bit-by-bit addition modulo 2.

As remarked before, the input of the first iteration of the calculation is the permuted input block. If $L'R'$ is the output of the 16th iteration then $R'L'$ is the preoutput block. At each iteration a different block K of key bits is chosen from the 64-bit key designated by *KEY*.

With more notation we can describe the iterations of the computation in more detail. Let KS be a function which takes an integer n in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block K_n which is a permuted selection of bits from KEY. That is

(2) $$K_n = KS(n, KEY)$$

with K_n determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule because the block K used in the n'th iteration of (1) is the block K_n determined by (2).

As before, let the permuted input block be LR. Finally, let L_0 and R_0 be respectively L and R and let L_n and R_n be respectively L' and R' of (1) when L and R are respectively L_{n-1} and R_{n-1} and K is K_n; that is, when n is in the range from 1 to 16,

(3) $$L_n = R_{n-1}$$
$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

The preoutput block is then $R_{16}L_{16}$.

The key schedule KS of the algorithm is described in detail in the Appendix. The key schedule produces the 16 K_n which are required for the algorithm.

Deciphering

The permutation IP^{-1} applied to the preoutput block is the inverse of the initial permutation IP applied to the input. Further, from (1) it follows that:

(4) $$R = L'$$
$$L = R' \oplus f(L', K)$$

Consequently, to *decipher* it is only necessary to apply the *very same algorithm to an enciphered message block,* taking care that at each iteration of the computation *the same block of key bits* K *is used* during decipherment as was used during the encipherment of the block. Using the notation of the previous section, this can be expressed by the equations:

(5) $$R_{n-1} = L_n$$
$$L_{n-1} = R_n \oplus f(L_n, K_n)$$

where now $R_{16}L_{16}$ is the permuted input block for the deciphering calculation and L_0R_0 is the preoutput block. That is, for the decipherment calculation with $R_{16}L_{16}$ as the permuted input, K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the 16th iteration.

The Cipher Function f

A sketch of the calculation of $f(R, K)$ is given in figure 2.

R (32 BITS)

E

48 BITS

K (48 BITS)

+

S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8

P
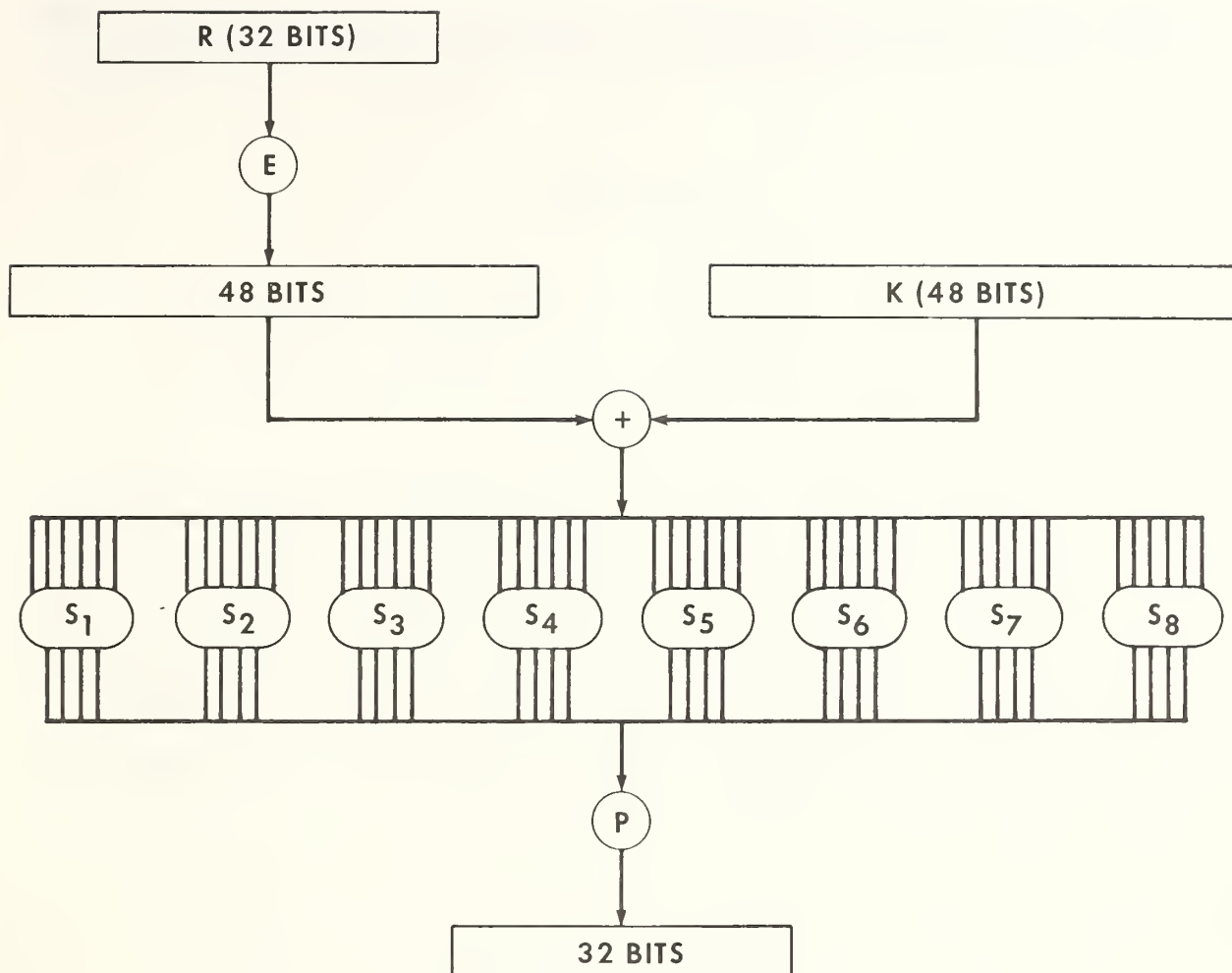
32 BITS

FIGURE 2. *Calculation of f (R, K).*

Let E denote a function which takes a block of 32 bits as input and yields a block of 48 bits as output. Let E be such that the 48 bits of its output, written as 8 blocks of 6 bits each, are obtained by selecting the bits in its inputs in order according to the following table:

E BIT-SELECTION TABLE

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Thus the first three bits of $E(R)$ are the bits in positions 32, 1 and 2 of R while the last 2 bits of $E(R)$ are the bits in positions 32 and 1.

Each of the unique selection functions $S_1, S_2, \ldots, S_8$, takes a 6-bit block as input and yields a 4-bit block as output and is illustrated by using a table containing the recommended S_1:

$$\underline{S_1}$$

Column Number

Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

If S_1 is the function defined in this table and B is a block of 6 bits, then $S_1(B)$ is determined as follows: The first and last bits of B represent in base 2 a number in the range 0 to 3. Let that number be i. The middle 4 bits of B represent in base 2 a number in the range 0 to 15. Let that number be j. Look up in the table the number in the i'th row and j'th column. It is a number in the range 0 to 15 and is uniquely represented by a 4 bit block. That block is the output $S_1(B)$ of S_1 for the input B. For example, for input 011011 the row is 01, that is row 1, and the column is determined by 1101, that is column 13. In row 1 column 13 appears 5 so that the output is 0101. Selection functions $S_1, S_2, \ldots, S_8$ of the algorithm appear in the Appendix.

The permutation function P yields a 32-bit output from a 32-bit input by permuting the bits of the input block. Such a function is defined by the following table:

$$\underline{P}$$

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

The output $P(L)$ for the function P defined by this table is obtained from the input L by taking the 16th bit of L as the first bit of $P(L)$, the 7th bit as the second bit of $P(L)$, and so on until the 25th bit of L is taken as the 32nd bit of $P(L)$. The permutation function P of the algorithm is repeated in the Appendix.

Now let $S_1, \ldots, S_8$ be eight distinct selection functions, let P be the permutation function and let E be the function defined above.

To define $f(R, K)$ we first define $B_1, \ldots, B_8$ to be blocks of 6 bits each for which

(6)
$$B_1 B_2 \ldots B_8 = K \oplus E(R)$$

The block $f(R, K)$ is then defined to be

(7)
$$P(S_1(B_1) S_2(B_2) \ldots S_8(B_8))$$

Thus $K \oplus E(R)$ is first divided into the 8 blocks as indicated in (6). Then each B_i is taken as an input to S_i and the 8 blocks $S_1(B_1)$, $S_2(B_2)$, ..., $S_8(B_8)$ of 4 bits each are consolidated into a single block of 32 bits which forms the input to P. The output (7) is then the output of the function f for the inputs R and K.

APPENDIX

PRIMITIVE FUNCTIONS FOR THE DATA ENCRYPTION ALGORITHM

The choice of the primitive functions KS, S_1, ..., S_8 and P is critical to the strength of an encipherment resulting from the algorithm. Specified below is the recommended set of functions, describing S_1, ..., S_8 and P in the same way they are described in the algorithm. For the interpretation of the tables describing these functions, see the discussion in the body of the algorithm.

The primitive functions S_1, ..., S_8, are:

$$S_1$$

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

$$S_2$$

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

$$S_3$$

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

$$S_4$$

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

$$S_5$$

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

$$S_6$$

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

$$S_7$$

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

$$S_8$$

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

The primitive function P is:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Recall that K_n, for $1 \leq n \leq 16$, is the block of 48 bits in (2) of the algorithm. Hence, to describe KS, it is sufficient to describe the calculation of K_n from KEY for $n = 1, 2, \ldots, 16$. That calculation is illustrated in figure 3. To complete the definition of KS it is therefore sufficient to describe the two permuted choices, as well as the schedule of left shifts. One bit in each 8-bit byte of the KEY may be utilized for error detection in key generation, distribution and storage. Bits 8, 16, . . ., 64 are for use in assuring that each byte is of odd parity.

Permuted choice 1 is determined by the following table:

$$PC-1$$

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

The table has been divided into two parts, with the first part determining how the bits of C_0 are chosen, and the second part determining how the bits of D_0 are chosen. The bits of KEY are numbered 1 through 64. The bits of C_0 are respectively bits 57, 49, 41, . . ., 44 and 36 of KEY, with the bits of D_0 being bits 63, 55, 47, . . ., 12 and 4 of KEY.

With C_0 and D_0 defined, we now define how the blocks C_n and D_n are obtained from the blocks C_{n-1} and D_{n-1}, respectively, for $n = 1, 2, \ldots, 16$. That is accomplished by adhering to the following schedule of left shifts of the individual blocks:
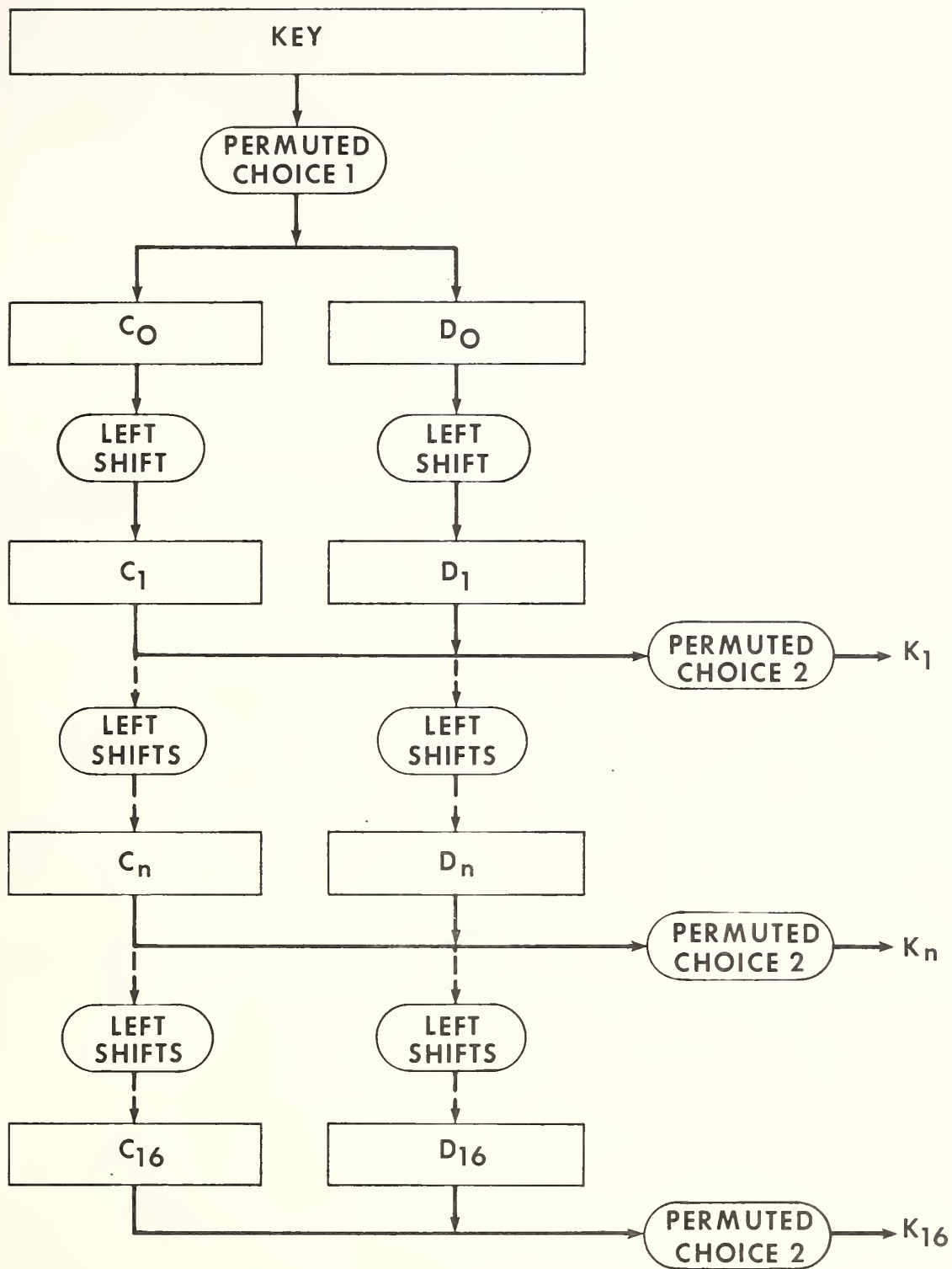
16

FIGURE 3. *Key schedule calculation.*

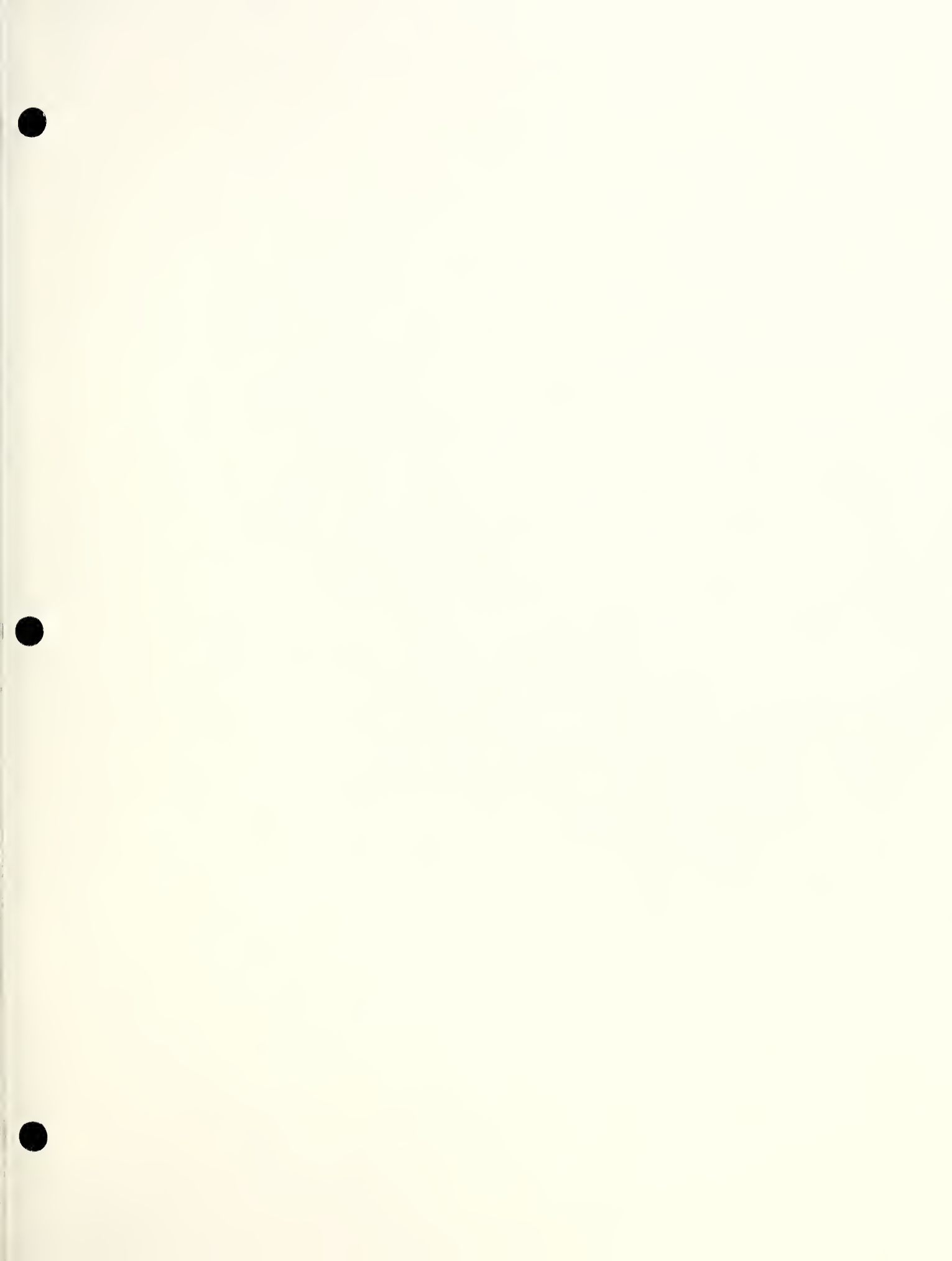| Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

For example, C_3 and D_3 are obtained from C_2 and D_2, respectively, by two left shifts, and C_{16} and D_{16} are obtained from C_{15} and D_{15}, respectively, by one left shift. In all cases, by a single left shift is meant a rotation of the bits one place to the left, so that after one left shift the bits in the 28 positions are the bits that were previously in positions 2, 3, . . ., 28, 1.

Permuted choice 2 is determined by the following table:

$$PC-2$$

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Therefore, the first bit of K_n is the 14th bit of C_nD_n, the second bit the 17th, and so on with the 47th bit the 29th, and the 48th bit the 32nd.