Incorporating Amendment No 1

AS 2805.5—1985
UDC 336.717.131.3:681.3.04

Australian Standard®
2805.5—1985

ELECTRONIC FUNDS TRANSFER—REQUIREMENTS FOR INTERFACES

Part 5—DATA ENCRYPTION ALGORITHM

STANDARDS ASSOCIATION OF AUSTRALIA
Incorporated by Royal Charter

D

This Australian standard was prepared by Committee IS/5, Electronic Funds Transfer. It was approved on behalf of the Council of the Standards Association of Australia on 19 March 1985 and published on 17 May 1985.

The following interests are represented on Committee IS/5:

Australian Association of Permanent Building Societies
Australian Bankers Association
Australian Computer Equipment Manufacturers Association
Australian Electrical and Electronics Manufacturers Association
Australian Federation of Credit Union Leagues
Australian Information Industry Association
Australian Institute of Petroleum
Australian Retailers Association
Australian Software Houses Association
Catering Institute of Australia
Life Insurance Federation of Australia
National Card Issuers
National Network Operators
Reserve Bank of Australia
Telecom Australia

This standard was issued in draft form for comment as DR 84122.

AUSTRALIAN STANDARD

ELECTRONIC FUNDS TRANSFER—REQUIREMENTS FOR INTERFACES

Part 5
DATA ENCRYPTION ALGORITHM

AS 2805.5—1985

First published 1985
Amended ..... December 1985
Reprinted incorporating Amdt 1985

PUBLISHED BY THE STANDARDS ASSOCIATION OF AUSTRALIA
STANDARDS HOUSE, 80 ARTHUR ST, NORTH SYDNEY, N.S.W.

ISBN 0 7262 3764 7

PREFACE

This standard was prepared by the Association's Committee on Electronic Funds Transfer. It is one of a series of standards on electronic funds transfer (EFT), requirements for interfaces; the other standards in the series being as follows:

Part 1—Communications Interface and Data Representation

Part 2—Message Structure, Format and Content

Part 3—PIN Management and Security

Part 4—Message Authentication

Part 6—Terminal Key Management and Security*

Part 7—Point of Service Message Content*

Part 8—Financial Institution Message Content*

It should be noted that in this series of standards, the definitions are specific to the Part in which they appear.

In this Part 5, Appendices A and B have been included for the guidance of users; they do not form part of the requirements of this standard.

The algorithm specified in this standard is based on the algorithm specified in American National Standard ANSI X3.92—1981, American National Standard Data Encryption Algorithm, copyright 1981 by the American National Standards Institute.

The two modes of operation specified in this standard are based on two modes of operation given in ANSI X3.106—1983, American National Standard for Information Systems—Data Encryption Algorithm—Modes of Operation, copyright 1983 by the American National Standards Institute.

Material from ANSI X3.92 and ANSI X3.106 has been incorporated herein with the permission of the American National Standards Institute and acknowledgment is made of the assistance received from ANSI.

NOTE: Copies of ANSI X3.92 and ANSI X3.106 may be purchased from ANSI at 1430 Broadway, New York, NY 10018 or from SAA Head Office.

---

*In course of preparation

CONTENTS

STANDARDS ASSOCIATION OF AUSTRALIA

Australian Standard
for
ELECTRONIC FUNDS TRANSFER—REQUIREMENTS FOR INTERFACES

PART 5—DATA ENCRYPTION ALGORITHM

FOREWORD

This standard specifies a data encryption algorithm (DEA) for the cryptographic protection of digital data. The DEA is a complete description of a mathematical algorithm for encrypting and decrypting binary-coded information. Encrypting data converts it to an unintelligible form. Decrypting converts the data back to the original form. The algorithm described in this standard specifies both encrypting and decrypting operations, which are based on a binary number called a key. The key consists of 64 binary digits (0s or 1s), of which 56 bits are used directly by the algorithm and 8 bits may be used for error detection. The algorithm specified in this standard is identical to the algorithm specified in ANSI X3.92 and Federal Information Processing Standard 46 (published by the US National Bureau of Standards).

Binary-coded data may be cryptographically protected using the DEA in conjunction with a key. The key is generated in such a way that each of the 56 bits used directly by the algorithm is random. Each member of a group of authorized users of encrypted data must have the key that was used to encrypt the data in order to decrypt it. This key, held by each member in common, is used to decrypt the data received in encrypted form from other members of the group. The encryption algorithm specified herein is publicly known. Therefore, the cryptographic security of the data depends solely on the security provided for the key used to encrypt and decrypt the data.

Data can only be decrypted by using exactly the same key used to encrypt it. Unauthorized recipients of the encryption who know the algorithm but do not have the correct key, cannot decrypt it and obtain the original data. However, anyone who does have the key and the algorithm, can easily obtain the original data. A standard algorithm, based on a secure key, thus provides a basis for exchanging encrypted digital data by issuing the key used to encrypt it to those authorized to have the data.

This standard also specifies two ways of using the data encryption algorithm. Each way, or mode of operation, provides different benefits and has different characteristics. The mechanics, or functional operation, of the modes are defined in this standard, but how to implement the modes and when to use them are not specified. These modes of operation permit anyone implementing and using the data encryption algorithm to communicate securely with anyone else using the algorithm, provided that they have selected the same mode of operation and share the same cryptographic key. The use of this algorithm provides for compatibility, even though the two communicating parties use different implementations. Therefore, one person may have implemented the mode of the algorithm in software to minimize initial cost while the other person may have implemented the mode and the algorithm in hardware to maximize security and speed.

The two modes of operation specified in this standard are identical with two of the four specified in ANSI X3.106 and Federal Information Processing Standard 81 (published by the US National Bureau of Standards).

SPECIFICATION

1 SCOPE. This standard specifies a mathematical algorithm for encrypting and decrypting the information contained in specific data fields of card-originated electronic messages relating to financial transactions. It also specifies two modes of operation for using the algorithm.

2 DEFINITIONS. For the purpose of this standard, the following definitions apply:

2.1 Algorithm—a clearly specified mathematical process for computation; a set of rules which, if followed, will give a prescribed result.

2.2 Cipher text—clear text that has been encrypted.

2.3 Clear text—intelligible text or signals that have meaning and that can be read and used.

2.4 Data Encryption Algorithm (DEA)—an algorithm designed to encrypt and decrypt blocks of data.

2.5 Decryption—the transformation of cipher text into clear text.

NOTE: 'Decryption' is sometimes referred to as 'decipherment'.

2.6 Encryption—the transformation of clear text into cipher text for the purpose of security or privacy.

NOTE: 'Encryption' is sometimes referred to as 'encipherment'.

2.7 Encryption algorithm—a set of mathematically expressed rules for rendering information unintelligible by effecting a series of transformations to the normal representation of the information through the use of variable elements controlled by the application of a key.

2.8 Initialization vector—a binary vector used in the initial input block in cipher block chaining mode of operation.

2.9 Key—a 64-bit quantity which is used for transformations between cipher text and clear text.

2.10 Modulo 2 addition—a mathematical operation equivalent to binary addition without carry.

NOTE: 'Modulo 2 addition' is represented by the symbol $\oplus$ and is sometimes referred to as an 'exclusive OR' operation.

3 DATA ENCRYPTION ALGORITHM SPECIFICATIONS.

3.1 Introduction. The data encryption algorithm (DEA) is designed to encrypt and decrypt blocks of data consisting of 64 bits, under control of a 64 bit key. Decrypting must be accomplished by using the same key that was used for encrypting, but with the schedule of addressing the key bits altered so that the decrypting process is the reverse of the encrypting process. A block to be encrypted is subject to an initial permutation *IP*, then to a complex key-dependent computation, and finally to a permutation IP^{-1} that is the inverse of the initial permutation. 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. Descriptions of the computation and the encrypting operation are provided in Clause 3.2. The decrypting operation is described in Clause 3.3. The definition of the encryption function *f* is given in Clause 3.4. The *S* and *KS* functions of the algorithm are described in Clause 3.5.

NOTE: The representation of keys, and special key values, are described in Appendix A.

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, $B_1B_2 \ldots B_8$, 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.

3.2 Encrypting. The encrypting computation is illustrated in Fig. 1. Thus, the 64 bits of the input block to be encrypted are first subjected to the initial permutation *IP*, as given in Table 1. 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 input to a complex, key-dependent computation described by the equations that follow. The output of that computation, called the preoutput, is then subjected to the permutation given in Table 2, which is the inverse of the initial permutation. Thus, 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.

TABLE 1

INITIAL PERMUTATION *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

TABLE 2

INVERSE OF 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

The computation uses the permuted input block as input to produce the preoutput block. It consists (except for a final interchange of blocks) of 16 iterations of a set of operations including calculation of the encryption function *f*, which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits. The function *f* is described in Clause 3.4.

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 Clause 3.1, 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 as follows:

$$L' = R$$

$$R' = L \oplus f(R, K) \qquad \text{(Eq. 1)}$$