INTERNATIONAL STANDARD

ISO 10126-1

First edition
1991-11-01

Banking — Procedures for message encipherment (wholesale) —

Part 1: General principles

Banque — Procédures de chiffrement de messages (service aux entreprises) —
Partie 1: Principes généraux

Reference number
ISO 10126-1:1991(E)

Contents

International Organization for Standardization
Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 10126-1 was prepared by Technical Committee ISO/TC 68, *Banking and related financial services*, Sub-Committee SC 2, *Operations and procedures*.

ISO 10126 consists of the following parts, under the general title *Banking — Procedures for message encipherment (wholesale)*:

— *Part 1: General principles*

— *Part 2: DEA algorithm*

ISO 10126 originates from work done in ANSI, and is a development of (while remaining compatible with) ANSI X9.23 (1988), *Financial Institution: Encryption of Wholesale Financial Messages*.

The four annexes to this part of ISO 10126 are intended to simplify its implementation.

a) Annex A provides the procedure for review of alternative encipherment algorithms.

b) Annex B provides examples of the various filtering techniques described in this part of ISO 10126.

c) Annex C provides a comparison of the effects of the filters described in this part of ISO 10126 and, by use of the expansion factor, shows the relationship between the number of bits transmitted and those originated.

d) Annex D provides examples of the three methods for encipherment and decipherment of encipherment elements within a message which are described in this part of ISO 10126.

Annex A forms an integral part of this part of ISO 10126. Annexes B, C and D are for information only.

Introduction

This part of ISO 10126 specifies a method for the encipherment and decipherment of entire (or parts of) wholesale financial messages by the use of application level encipherment, for the purpose of providing confidentiality.

The level of security provided by this part of ISO 10126 is dependent upon a) the security associated with the algorithm used for encipherment and the implementation of that algorithm within the procedures laid down by this part of ISO 10126 and, b) the operation of a secure system of key management.

Particular algorithms, suitable for use with this part of ISO 10126, are described in ISO 10126-2. A suitable International Standard for key management is described in ISO 8732.

Banking — Procedures for message encipherment (wholesale) —

Part 1:
General principles

1 Scope

The procedures defined are designed to protect, by means of encipherment, financial messages (entire messages or encipherment elements) exchanged through any communications architecture. Such architectures will include store and forward and telex environments, any number of nodes and public or private networks.

Since enciphered text can interfere with communications processes in existing wholesale financial networks, this part of ISO 10126 provides a means to permit the enciphered message to be transmitted through a number of networks without being misinterpreted as communications protocol information [e.g. STX (start of text), EOT (end of text)].

The confidentiality of financial message data, in both structured and unstructured forms, is protected by the use of this part of ISO 10126.

The techniques described do not provide integrity protection (i.e. protection against modification, substitution and replay). Data integrity protection is the subject of ISO 8730 and ISO 8731. Message formats are also beyond the scope of this part of ISO 10126.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10126. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10126 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 646:1983, *Information processing — ISO 7-bit coded character set for information interchange.*

ISO 8730:1990, *Banking — Requirements for message authentication (wholesale).*

ISO 8731-1:1987, *Banking — Approved algorithms for message authentication — Part 1: DEA.*

ISO 8731-2:1987, *Banking — Approved algorithm for message authentication — Part 2: Message authenticator algorithms.*

ISO 8732:1988, *Banking — Key management (wholesale).*

ISO 10126-2:1991, *Banking — Procedures for message encipherment (wholesale) — Part 2: DEA algorithm.*

3 Definitions

For the purposes of this part of ISO 10126, the following definitions apply.

3.1 baudot: A five bit character information coding scheme (excluding optional start bits and stop bits); CCITT Alphabet Number 2.

3.2 block: A data unit with a specified length.

3.3 ciphertext: Enciphered information.

3.4 communicating pair: Two logical parties who have previously agreed to exchange data.

3.5 cryptographic key; key: A parameter used in conjunction with an algorithm for the purpose of encipherment or decipherment.

3.6 data unit: Binary vector of K bits denoted as (B1, B2, ..., BK).

3.7 decipherment: The reversal of a corresponding reversible encipherment.

3.8 enciphered element: Enciphered encipherment element.

3.9 encipherment: The cryptographic transformation of data to produce ciphertext.

3.10 encipherment element: A contiguous group of plaintext characters which is to be enciphered.

3.11 filtering: The process of encoding binary text into a format insensitive to control characters.

3.12 financial message: A communication containing information which has financial implications.

3.13 Initialization Vector (IV): A number used as a starting point for encipherment of a data sequence. It increases security by introducing additional cryptographic variance, and also facilitates the synchronization of cryptographic equipment.

3.14 Initial Text Sequence (ITS): An n-bit binary vector which may be prefixed to a message.

3.15 key: See cryptographic key.

3.16 logical party: One or more physical parties forming one member of a communicating pair.

3.17 message: A communication containing one or more transactions or related information.

3.18 most significant bit(s): The leftmost bit(s) of a binary vector.

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

3.20 padding: One or more bits appended to a message in order to cause the message to contain an exact multiple of the number of bits required by the filtering process or the cryptographic algorithm.

3.21 plaintext: Unenciphered information.

3.22 receiver: The logical party that is authorized to decipher a received message.

3.23 sender: The logical party that is responsible for originating an enciphered message.

4 Application

4.1 Protection provided

The confidentiality of financial message data is protected by the use of this part of ISO 10126. This part of ISO 10126 provides a method of protection for both structured and unstructured message data. Confidentiality protection is provided between two logical parties. The logical parties to a communication are the sender and receiver.

4.2 General operation

4.2.1 Processing

The sender of a financial message shall generate ciphertext by applying the encipherment process described in clause 5 either to the entire plaintext message, or to encipherment elements within the message, using a secure system of key management. The message is then forwarded to the receiver. The receiver shall decipher the ciphertext by applying the decipherment procedures described in clause 5.

Note that the cryptographic process is sensitive to the sequence in which the plaintext is enciphered, and that sequence must be the same at the time of encipherment and decipherment.

If decipherment of an entire message or encipherment elements is required at a later date, an audit journal shall include sufficient information to retrieve the key, the message and any other information used in the cryptographic process.

When a financial message is both authenticated and enciphered, the message shall be authenticated prior to encipherment, and different keys shall be used for encipherment and authentication.

4.2.2 Communications methods

This part of ISO 10126 may be used to encipher entire financial messages or encipherment elements within messages transmitted over any communications media through any number of nodes in public or private networks.

5 Encipherment and decipherment of entire messages and encipherment elements

5.1 General

Entire messages and encipherment elements shall be enciphered and deciphered in accordance with ISO 10126-2. There are four methods of encipherment allowed by this part of ISO 10126, i.e. entire

message encipherment and three methods for enciphering encipherment elements. Several messages using different keys, different modes of operation, or methods of encipherment (see 5.3) and filters (see clause 6) may be combined within a single transmission.

5.2 Encipherment and decipherment of entire messages

When an entire message is to be enciphered, all plaintext other than header and trailer information (e.g. information added by a network for transmission purposes) shall be enciphered. The Initial Text Sequence (if present), the entire message including the MAC (if present), and the padding field (if present) shall be enciphered as a unit. After encipherment, ciphertext may be filtered in accordance with clause 6. When an entire message is to be deciphered, all ciphertext is deciphered as a unit.

5.3 Encipherment and decipherment of encipherment elements

Three methods for encipherment and decipherment of encipherment elements within a message are permitted by this part of ISO 10126.

Method 1: Encipherment elements are independently enciphered into enciphered elements. Enciphered elements are independently deciphered into encipherment elements (see 5.3.2).

Method 2: Encipherment elements are concatenated and enciphered as a single data string. The resulting ciphertext string is transmitted in the message as a single string. The ciphertext string is deciphered and the encipherment elements are inserted into the message text (see 5.3.3.1).

Method 3: Encipherment elements are concatenated and enciphered as a single data string. The resulting ciphertext string is divided into ciphertext substrings for transmission in the message. The ciphertext substrings are concatenated and deciphered as a single string. The resulting encipherment elements are inserted into the message text (see 5.3.3.2).

After encipherment, ciphertext may be filtered in accordance with clause 6. See annex D for examples of these three methods. The same key, the same algorithm and mode of operation, the same method of encipherment, and the same filter (if implemented) shall be used for each encipherment element within a particular message.

Encipherment of encipherment elements does not afford the same protection as entire message encipherment, e.g. the context of enciphered elements may give an unauthorized party clues as to the underlying plaintext.

5.3.1 Field delimiters

Explicit or implicit delimiters are required to demarcate individual encipherment elements, enciphered elements, ciphertext strings, and ciphertext substrings. Delimiters for individual encipherment elements allow the application to determine the encipherment elements to be enciphered. Delimiters for enciphered elements, ciphertext strings, and ciphertext substrings allow the application to determine the ciphertext which needs to be deciphered.

Each encipherment element may be either implicitly or explicitly delimited within a plaintext message. After encipherment a message may contain enciphered elements, or a ciphertext string, or ciphertext substrings. Each enciphered encipherment element or ciphertext substring may be either implicitly or explicitly delimited within a message. A ciphertext string within a message may be either explicitly or implicitly delimited.

Subclauses 5.3.2 and 5.3.3 describe the three methods of processing encipherment elements with the use of explicit delimiters. Implementations using implicit delimiters shall process encipherment elements, enciphered elements, ciphertext strings and ciphertext substrings in a manner equivalent to one of the three methods.

5.3.1.1 Implicit delimiters

An encipherment element, enciphered element, ciphertext string, or ciphertext substring is implicitly delimited, for the purpose of this part of ISO 10126, if its position in the message is fixed or unambiguously identified by format rules, and both sender and receiver have previously agreed to its use.

5.3.1.2 Explicit delimiters

In the absence of implicit delimiters, explicit delimiters shall be used to identify the beginning and the ending of encipherment elements, enciphered elements, a ciphertext string, or ciphertext substrings. Explicit delimiters are reserved. The following explicit delimiters are established for

a) Encipherment elements: QE- and -EQ

   For example: QE-encipherment element-EQ

b) Enciphered elements: QC- and -CQ

   For example: QC-enciphered element-CQ

c) Ciphertext strings or substrings: QC- and -CQ

   For example: QC-ciphertext string-CQ

   or: QC-ciphertext substring-CQ

5.3.2 Independent encipherment and decipherment of encipherment elements (method 1 of 5.3)

The encipherment element delimiters, QE- and -EQ, shall not be included in the encipherment process and shall be replaced with the enciphered element delimiters, QC- and -CQ, respectively. Filtering may be performed on enciphered elements as described in clause 6, before adding the QC- and -CQ delimiters.

Each enciphered element shall be placed in the message where the corresponding encipherment element had been. Enciphered elements may be longer than their corresponding encipherment elements for the following reasons:

a) the mode of encipherment may have required plaintext padding before encipherment;

b) filtering which causes data expansion may have been used.

The receiver defilters (if necessary) and independently deciphers all enciphered elements delimited by the QC- and -CQ delimiters. Each plaintext encipherment element shall be inserted in place of the corresponding enciphered element. The QC- and -CQ delimiters shall be replaced by the QE- and -EQ delimiters, respectively.

5.3.3 Encipherment and decipherment of concatenated encipherment elements

Encipherment elements may be assembled in sequence and enciphered as a single data string. After encipherment the ciphertext may be filtered in accordance with clause 6.

5.3.3.1 Ciphertext strings (method 2 of 5.3)

The ciphertext formed from enciphering both the concatenated encipherment elements and their -EQ delimiters may be placed within a message as a single ciphertext string. The ciphertext string shall be delimited by the QC- and -CQ delimiters. The QE- delimiters remain in the plaintext portion of the message as place markers.

The receiver deciphers the ciphertext delimited by the QC- and -CQ delimiters as a single string to yield an assembly of concatenated encipherment elements, each with a trailing -EQ delimiter. Each encipherment element and its -EQ delimiter shall be placed in sequence in the plaintext at the point following its corresponding QE- delimiter. For example, the first encipherment element and its -EQ delimiter is placed after the first QE- delimiter in the plaintext message and the second encipherment element and its -EQ delimiter is placed after the second QE- delimiter in the plaintext message. The QC- and -CQ delimiters along with the ciphertext are discarded.

5.3.3.2 Ciphertext substrings (method 3 of 5.3)

The ciphertext formed from enciphering concatenated encipherment elements without any delimiters may be divided into ciphertext substrings which are inserted in place of the encipherment elements in the message. When ciphertext is inserted in place of an encipherment element, each bit of the encipherment element is replaced by a single ciphertext bit, selected in sequence, except for the last encipherment element which contains all remaining ciphertext bits. Each ciphertext substring shall be delimited by the QC- and -CQ delimiters which replace in the message the QE- and -EQ delimiters, respectively.

The receiver assembles all ciphertext substrings without their corresponding delimiters, in sequence, into a ciphertext string. The ciphertext string is then deciphered and divided into encipherment elements which are inserted in place of the ciphertext substrings in the message. For example, if the first ciphertext substring of the message is 16 bits, then those bits are replaced by the first 16 bits of the deciphered ciphertext string. If the second ciphertext substring of the message is 40 bits, then those bits are replaced by the next 40 bits of the deciphered ciphertext string. This process continues until the last ciphertext substring is reached. All the bits of the last ciphertext substring are replaced by the remaining bits of the deciphered ciphertext string. Finally the QC- and -CQ delimiters shall be replaced in the message by the QE- and -EQ delimiters, respectively.

6 Transparent transmission of enciphered data

6.1 Introduction

The result of the encipherment process is a pseudo-random binary string. Thus

a) Communications systems implementing ISO 646 character coding may interpret or recognize some of this pseudo-random binary output as control functions. These control functions may have a detrimental effect on transmission service operation (e.g. spurious XON/XOFF or End of File characters).

b) Communications systems implementing baudot character coding may interpret or recognize some of the ciphertext as character strings which are used by transmission services as control strings. Detrimental results may be expected from the interpretation of baudot control-specific character strings (e.g. spurious BT or EOT strings).

Consequently, transformation (filtering) of binary output is necessary for communications systems which are sensitive to control functions or control character strings. Examples of the various filtering techniques are described in annex B. Annex C provides a comparison of the effects of the various filters described in this part of ISO 10126.

6.2 No filtering

Where the transmission service is transparent to control characters and strings, no transmission filter need be used. However, if explicit delimiters as described in 5.3.1.2 are used, it is important to avoid the occurrence of these delimiters in the ciphertext (see 6.9).

6.3 Hexadecimal filtering

Filtering shall be accomplished by converting each four bits of ciphertext into a hexadecimal character, and then transmitting it in the character code used by the network. Example 1 of annex B describes a hexadecimal filter.

6.4 ISO 646 filter

Networks using the International Reference Version of the character code in accordance with ISO 646 and which do not translate lower-case letters to upper-case may use the 94 character filter described in this clause. Example 2 of annex B describes an ISO 646 (IRV) filter.

The binary stream is divided into groups of 13-bit data units, each data unit construed as an unsigned binary integer. The last data unit shall be padded to 13 bits if necessary (see 6.6). The two characters corresponding to the 13-bit number are found using table 1. Decoding is done by reversing the process.

A single table suffices for this filter (table 1 shows the first 95 values in that table). The first column consists of numbers from 0 to 8 191 in order. Column 2 converts this original input to the first 8 192 possible pairs of non-blank non-hyphen printable characters in ISO 646, table 11, International Reference Version, in lexicographical order.

The first character in this filter form character pair may be represented differently under ISO 646 as modified for national use. However, this part of ISO 10126 uses these elements only as a vehicle for transforming 13-bit vectors into 14-bit vectors, and does not require printing of the characters. Therefore no conflicts arise from the use of varying national representations.

Table 1 — ISO 646 filter (showing the sequence of permitted characters)

Original input	Filter form		Original input	Filter form		Original input	Filter form	
0	!	!	14	!	0	29	!	?
1	!	"	15	!	1	30	!	@
2	!	#	16	!	2	31	!	A
3	!	¤	17	!	3	32	!	B
4	!	%	18	!	4	33	!	C
5	!	&	19	!	5	34	!	D
6	!	'	20	!	6	35	!	E
7	!	(	21	!	7	36	!	F
8	!	)	22	!	8	37	!	G
9	!	*	23	!	9	38	!	H
10	!	+	24	!	:	39	!	I
11	!	,	25	!	;	40	!	J
12	!	-	26	!	<	41	!	K
13	!	/	27	!	=	42	!	L
			28	!	>	43	!	M

Original input	Filter form		Original input	Filter form		Original input	Filter form	
44	!	N	62	!	'	80	!	r
45	!	O	63	!	a	81	!	s
46	!	P	64	!	b	82	!	t
47	!	Q	65	!	c	83	!	u
48	!	R	66	!	d	84	!	v
49	!	S	67	!	e	85	!	w
50	!	T	68	!	f	86	!	x
51	!	U	69	!	g	87	!	y
52	!	V	70	!	h	88	!	z
53	!	W	71	!	i	89	!	{
54	!	X	72	!	j	90	!	|
55	!	Y	73	!	k	91	!	}
56	!	Z	74	!	l	92	!	~
57	!	[	75	!	m	93	"	!
58	!	\	76	!	n	94	"	"
59	!	]	77	!	o	95	"	#
60	!	^	78	!	p	↓		
61	!	_	79	!	q	8 191	z	(