INTERNATIONAL STANDARD

ISO 16609

First edition
2004-03-01

Banking — Requirements for message authentication using symmetric techniques

Banque — Exigences pour authentification des messages utilisant des techniques symétriques

Reference number
ISO 16609:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. 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.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 16609 was prepared by Technical Committee ISO/TC 68, *Banking, securities and other financial services*, Subcommittee SC 6, *Retail financial services*.

This first edition of ISO 16609 cancels and replaces ISO 8730:1990, ISO 8731-1:1987, ISO 8731-2:1992 and ISO 9807:1991, of which it constitutes a technical revision.

Introduction

A MAC (message authentication code) is a data field used to verify the authenticity of a message, generated by the sender of the message and transmitted together with it. The MAC enables an intended recipient to detect if the message has been altered and, if so, whether such an alteration arises by accident or with intent to defraud.

This International Standard has been prepared so that institutions involved in banking activities wishing to implement message authentication can do so in a secure manner and in a way that facilitates interoperability between separate implementations.

The requirements of this International Standard are compatible with those in the editions of ISO 8730 and ISO 9807 it replaces.

Banking — Requirements for message authentication using symmetric techniques

1 Scope

This International Standard specifies procedures, independent of the transmission process, for protecting the integrity of transmitted banking messages and for verifying that a message has originated from an authorized source. It also specifies a method by which block ciphers can be approved for use in the authentication of banking messages. In addition, because of the necessity for both members in a communicating pair to use the same means for data representation, it defines some methods for data representation. A list of block ciphers approved for the calculation of a message authentication code (MAC), as well as the method to be used to approve additional block ciphers, is also provided. The authentication methods it defines are applicable to messages formatted and transmitted both as coded character sets and as binary data.

This International Standard is designed for use with symmetric algorithms where both sender and receiver use the same key. It does not specify methods for establishing the shared key, nor does it provide for encipherment for the protection of messages against unauthorized disclosure. Its application will not protect the user against internal fraud by sender or receiver, or forgery of a MAC by the receiver.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 7746:1998, *Banking — Telex formats for inter-bank messages*

ISO 8583:1993, *Financial transaction card originated messages — Interchange message specifications*

ISO 8601:2000, *Data elements and interchange formats — Information interchange — Representation of dates and times*

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

ISO/IEC 9797-1:1999, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*

ISO/IEC 9797-2:2002, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a hash-function*

ISO/IEC 10116:1997, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

ISO/IEC 10118-3:1998, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO 11568-1:1994, *Banking — Key management (retail) — Part 1: Introduction to key management*

ISO 11568-2:1994, *Banking — Key management (retail) — Part 2: Key management techniques for symmetric ciphers*

ISO 11568-3:1994, *Banking — Key management (retail) — Part 3: Key life cycle for symmetric ciphers*

ISO 13491 (all parts) *Banking — Secure cryptographic devices (retail)*

ANSI X3.92:1981, *American National Standard for Information Systems — Data encryption algorithm*

ANSI X9.52:1998, *American National Standard for Financial Services — Triple data encryption algorithm, modes of operation*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
algorithm
specified mathematical process for computation or set of rules which, if followed, will give a prescribed result

3.2
authentication
process used between a sender and a receiver to ensure data integrity and provide data origin authentication

3.3
authentication algorithm
algorithm used, together with an authentication key and one or more authentication elements, for authentication

3.4
authentication element
message element that is to be protected by authentication

3.5
authentication key
cryptographic key used for authentication

3.6
beneficiary [party]
ultimate party (can be more than one) to be credited or paid as a result of a transfer

3.7
block cipher
algorithm for computing a function which maps a fixed length string of bits and a secret key to another string of bits with the same fixed length

3.8
bias
condition where, during the generation of random or pseudo-random numbers, the occurrence of some numbers is more likely than others

3.9
cryptoperiod
defined period of time during which a specific cryptographic key is authorized for use, or during which time the cryptographic keys in a given system may remain in effect

3.10
cryptanalysis
art and science of breaking ciphertext

3.11
data integrity
property defining data that has not been altered or destroyed in an unauthorized manner

3.12
date MAC computed
DMC
date on which the sender computed the message authentication code (MAC)

NOTE The DMC can be used to synchronize the authentication process through selection of the proper key.

3.13
data origin authentication
corroboration that the source of data received is as claimed

3.14
decipherment
decryption
reversal of a corresponding encipherment

3.15
delimiter
group of characters used to delineate the beginning and end of a data field or fields

3.16
encipherment
encryption
(reversible) transformation of data by a cryptographic algorithm to produce ciphertext (i.e. to hide the information content of the data)

3.17
hexadecimal digit
single character in the range 0 to 9, A to F (upper case), representing a four-bit string

3.18
identifier for authentication key
IDA
field that identifies the key to be used in authenticating the message

3.19
message authentication code
MAC
string of bits that is the output of a MAC algorithm

3.20
MAC algorithm
cryptographic check function
algorithm for computing a function, which maps strings of bits and a secret key to fixed length strings of bits

NOTE 1 It must satisfy the following two properties:

— for any key and any input string, the function can be computed efficiently;

— for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the ith input string may have been chosen after observing the value of the first i to 1 function values.

NOTE 2 Computational feasibility depends on the user's specific security requirements and environment.

3.21
message element
contiguous group of characters designated for a specific purpose

3.22
message identifier
MID
systems trace audit number (superseded)
field used uniquely to identify a financial message or transaction (e.g. sending bank's transaction reference) within a given context (e.g. DMC).

NOTE In ISO 8583, the MID was referred to as the systems trace audit number (STAN), which it supersedes.

3.23
message text
information conveyed or transmitted between sender and receiver, excluding header and trailer information used for transmission purposes

3.24
nonce
number used once

3.25
receiver
party intended to receive the message

3.26
sender
party responsible for, and authorized to, send a message

3.27
value date
date on which funds are to be at the disposal of the beneficiary

4 Protection

IMPORTANT — Integrity protection applies only to the selected authentication elements. Other parts of the message can be subject to undetected alterations. It is important that users ensure the integrity of the presentation of the data.

4.1 Protection of authentication keys

Authentication keys are secret cryptographic keys that have been previously established by the sender and receiver and which are used by the authentication algorithm. Such keys shall be deterministically or pseudo-randomly generated (see Annex F and Annex G). Any key used for authentication shall be protected against disclosure to unauthorized parties. Use of the authentication keys shall be restricted to the sending and receiving parties (or their authorized agents) and only used for authentication. Keys shall be managed in accordance with ISO 11568 or ISO 8732.

Authentication keys can best be protected if the MAC is computed by a secure cryptographic device, the keys are in plaintext only within such a device and the device is compliant with ISO 13491.

4.2 Authentication elements

The MAC calculation shall include those message elements, as agreed between sender and receiver, which require protection against fraudulent alteration. All such message elements should be included in the MAC calculation.

Subject to bilateral agreement, the MAC calculation may also cover data elements not transmitted in the message (e.g. padding bits or data computable by both parties from information already shared).

The choice of data to be included in the MAC will depend on the specific application. The following elements shall be included in the calculation of the MAC whenever they appear in the message:

a) transaction amount;

b) currency;

c) identifier for authentication key (IDA);

d) identification of parties to be credited and debited;

e) identification of beneficiary party;

f) value date;

g) message identifier;

h) date and time;

i) indication as to the disposition of the transaction.

4.3 Detection of duplication or loss

A mechanism shall be implemented to detect duplication or loss. Without recourse to further message exchanges, the recipient may only detect the replay of a previous transaction if able to identify transactions uniquely, and shall then check that such unique identifying information has not already occurred. Furthermore, in order to detect loss, transactions shall be identifiable as being in a sequence. Both conditions are thus achieved by involving in the MAC computation some elements (i.e., message elements or key elements) that are unique to the transaction and that relate it uniquely to the previous transaction. This shall be achieved in one of the following ways.

a) Include in the MAC calculation a unique transaction reference that does not repeat within the lifetime of the system.

   EXAMPLE The reference will include sender ID, recipient ID, key ID, transaction number and date.

b) Include in the MAC calculation a message identifier (MID), a value that does not repeat before either

   — the change of date, i.e. date MAC computed (DMC), or

   — the expiration of the cryptoperiod of the key used for authentication,

   whichever occurs first: i.e. there shall not be more than one message with the same date and the same message identifier that uses the same key.

   The MID may consist of a unique sending bank's transaction reference number in a fixed format message as a message identifier. A method of protection is described in Annex E. The MID may either contain the DMC or be a separate field.

c) Use a unique key per transaction where either

— the key of one transaction is derived from that of the previous transaction (see, for instance, ISO 11568-2 and Annex G), or

— the key is derived using a unique transaction reference (see Annex G).

d) Combine the above techniques.

5 Procedures for message authentication

5.1 Preliminaries

Implementers shall conduct a risk assessment of the application to determine the data to be protected (see Clause 4), the required key length and MAC algorithm, and shall agree upon the following:

— a block cipher (if MAC algorithm chosen from ISO/IEC 9797-1);

— a hash-function (if MAC algorithm chosen from ISO/IEC 9797-2);

— a padding method (if MAC algorithm chosen from ISO/IEC 9797-1);

— the length in bits of the MAC;

— the key change frequency (this should take into consideration the current state of the art of cryptanalysis);

— a common key derivation method (if required by the MAC algorithm).

Approved block ciphers are given in Annex A.

The correspondents shall also exchange a secret authentication key.

Financial service applications should use a key with a length of less than 112 bits only with caution, and with a full understanding of underlying risk management and assessment (see ISO 13491). The ISO/IEC 9797-1:1999-specified MAC Algorithms 1 and 3 (see Clause 6) are recommended for applications requiring 112-bit MAC algorithm keys.

The sender shall calculate a MAC using the selected data elements. This MAC shall be appended to the text of the transmitted message such that it is identifiable by the receiver. The receiver shall repeat the computation, using the same authentication method as defined in this clause. The message authenticates if the received and computed reference MACs are identical.

Implementers should also consider the performance characteristics given in 6.3.

5.2 Message format

The sender shall format and code the message by a method agreed with the recipient.

5.3 Key generation

Subject to agreement with the receiving party, the sender of a message may generate a new key with which to compute the MAC. The derivation of such a key may involve transaction and message-dependent data. Annex F and Annex G provide some examples of key generation and derivation.

5.4 MAC Generation

The sender of a message shall generate a MAC by processing in an agreed order (e.g. the sequence in which they appear in the message) those authentication elements of the transmitted message that are to be protected by an approved authentication algorithm (see Clause 6). The algorithm shall be activated by means of an authentication key, which is a secret between the two correspondents. This process creates the MAC, which shall then be included with the original message text.

5.5 MAC placement

The MAC shall be either

a) placed in the message, in an additional field specified for the transport of the MAC, or

b) appended to the data portion of the message, if there is no specified MAC field.

Where the field allocated has a length, for transport, greater than the MAC length, the MAC shall be positioned by left justifying it within the field.

5.6 MAC checking

On receipt of the message, the receiver shall compute a reference MAC using the authentication elements, an identical authentication key and an identical algorithm. Authenticity of the content of the authentication elements and the message source shall be considered to have been confirmed when the receiver's computed reference MAC agrees with that received with the message text.

A received MAC (and its delimiters) shall not be included in the algorithm computation.

The process of generating the MAC is sensitive to the sequence in which the authentication elements are processed (i.e., a change in the sequence of authentication elements after the MAC is generated will result in a failure to authenticate).

6 Approved MAC algorithms

6.1 Overview of ISO/IEC 9797-1

6.1.1 Algorithms 1 to 6

The MAC algorithm shall be one of those specified in ISO/IEC 9797. The present clause offers an interpretation of the characteristics of those algorithms and a mapping between them and the algorithms of the superseded ISO 8731 and ISO 9807.

ISO/IEC 9797-1 specifies six MAC algorithms that use a secret key and an n-bit block cipher to calculate an m-bit MAC, and which are based upon the cipher block chaining (CBC) mode of operation of a block cipher.

NOTE 1 The security analysis given in ISO/IEC 9797-1:1999, Annex B, provides implementation recommendations for protecting against forgery and key recovery attacks.

NOTE 2 The modes of operation of an n-bit block cipher are standardized in ISO/IEC 10116.

— MAC Algorithm 1 is a simple CBC-MAC using a single key.

— MAC Algorithm 2 is a variant on Algorithm 1, with an additional final transformation using a second key.

— MAC Algorithm 3 is a variant on Algorithm 1, ending with two additional transformations, the penultimate transformation uses a second key and the final transformation uses the first key.

— MAC Algorithm 4 is a variant on Algorithm 2, with an additional initial transformation using the second key.

— MAC Algorithm 5 uses two parallel instances of Algorithm 1, and combines the two results with a bit-wise exclusive-or operation, while using a single-length MAC algorithm key.

— MAC Algorithm 6 uses two parallel instances of Algorithm 4, and combines the two results with a bit-wise exclusive-or operation, while doubling the MAC algorithm key length.

The strength of the MAC mechanism is dependent on the length (in bits) and secrecy of the key, on the block length (in bits) n and strength of the block cipher, on the length (in bits) m of the MAC, and on the specific algorithm.

6.1.2 Relationship to previous standards

This subclause provides a summary of the relationship between the algorithms specified in ISO/IEC 9797 and other, previous standards which are now superseded.

Table 1 — ISO 9797-1 — Relationship to previous standards

Standard	ISO 9797 algorithm	Block cipher	Block size n	Padding method	MAC size m
IS 8731-1 [ANSI X9.9]	1	DEA (ANSI X3.92: 1981)	64	1	32
IS 9807 [ANSI X9.19]	1 or 3	DEA (ANSI X3.92: 1981)	64	1	32

6.1.3 Minimum key lengths

MAC mechanisms should use keys providing at least 112 bits (in case of a shorter key length, see the related recommendation of 5.1).

6.1.4 Recommended mechanisms

This subclause provides a summary of the recommended implementations of ISO/IEC 9797-1 mechanisms.

Although ISO/IEC 9797-1 specifies six MAC algorithms, this International Standard recommends two of these for the financial services industry:

Algorithm 1 using T-DEA

Algorithm 3 using DEA

Table 2 — Recommended mechanisms

ISO 9797 algorithm	Block cipher	Block size n	Key length	MAC size m
1	T-DEA (ANSI X9.52:1998)	64	112	$32 \leqslant m \leqslant 64$
3	DEA (ANSI X3.92:1981)	64	112	$32 \leqslant m \leqslant 64$

The security analysis in ISO 9797-1:1999, Annex B, provides implementation recommendations for protecting against forgery and key recovery attacks.

If Algorithm 1 is used, then steps should be taken to prevent xor forgery attacks as described in ISO 9797-1:1999, Annex B. An adequate precaution is to use Padding Method 3.

If Algorithm 3 is used, then the number of MACs generated using the same key should be restricted. In order not to restrict the lifetime of the MAC-generating device, the use of session keys is recommended (see Annex G).

Trivial Forgery: If Padding Method 1 is used, then an adversary can typically add or delete a number of trailing '0' bits of the data string without changing the MAC. This implies that Padding Method 1 should only be used in environments where the length of the data string is known to the parties beforehand, or where data strings with a different number of trailing '0' bits have the same semantics.

6.2 Overview of ISO/IEC 9797-2

ISO/IEC 9797-2 specifies three MAC algorithms that use a secret key and a hash-function (or its round-function) with an n-bit result to calculate an m-bit MAC. The hash-functions are chosen from those specified in ISO/IEC 10118-3 (commonly known as SHA-1, RIPEMD-160 and RIPEMD-128).

The strength of the message authentication mechanism is dependent on the length (in bits) and secrecy of the key, on the length (in bits) n of the hash-function and its strength, on the length (in bits) m of the MAC, and on the specific algorithm.

— Hash Algorithm 1, the first algorithm specified in ISO/IEC 9797-2, is commonly known as MDx-MAC. It calls the complete hash-function once, but makes a small modification to the round-function by adding a key to the additive constants in the round-function.

— Hash Algorithm 2, the second algorithm specified in ISO/IEC 9797-2, is commonly known as HMAC. It calls the complete hash-function twice.

— Hash Algorithm 3, the third algorithm specified in ISO/IEC 9797-2, is a variant of MDx-MAC that takes as input only short strings (at most 256 bits). It offers a higher performance for applications that work with short input strings only.

6.3 Implementation recommendations

One simple criterion for choosing between mechanisms based on ISO/IEC 9797-1 algorithms and mechanisms based on ISO/IEC 9797-2 algorithms is the availability of an implementation of the block cipher or hash-function. Other criteria will determine the precise choice of parameters. For instance, when DEA is used as the block cipher then the following distinctions can be made:

— ISO/IEC 9797-1 mechanisms will often be slower than ISO/IEC 9797-2 mechanisms, especially in software;

— ISO/IEC 9797-1 mechanisms require less memory than ISO/IEC 9797-2 mechanisms;

— ISO/IEC 9797-2 mechanisms can provide longer MACs (up to 160 bits);

— the secret keys used with Algorithms 1 and 2 of ISO/IEC 9797-1 are shorter than those used with ISO/IEC 9797-2 algorithms. (Algorithm 1 with a single 56-bit DEA does not comply).

Tables 3 and 4 indicate the relative performance characteristics, respectively, of ISO/IEC 9797-1 algorithms using DEA or T-DEA as the underlying block cipher and ISO/IEC 9797-2 mechanisms using either SHA-1/RIPEMD-160 or RIPEDMD-128 as the underlying hash-function.

If Algorithm 1 is used with T-DEA as the underlying block cipher (instead of DEA) then the number of DEA computations is tripled.

A security comparison of all the MAC algorithms is provided in Annexes B of ISO/IEC 9797-1 and ISO/IEC 9797-1

Table 3 — ISO 9797-1 — Relative performance using the DEA

ISO 9797-1 algorithm	Block cipher	MAC size	Key length	Number of round functions/block cipher evaluations for message size: [a]		
				8 bytes	64 bytes	1 kB
1	DEA	⩽ 64	56	1 to 2	8 to 9	128 to 129
1	T-DEA	⩽ 64	112 or 168	1 to 2	8 to 9	128 to 129
2	DEA	⩽ 64	112	2 to 3	9 to 10	129 to 130
3	DEA	⩽ 64	112	3 to 4	10 to 11	130 to 131
4	DEA	⩽ 64	112	4 to 5	10 to 11	130 to 131
5	DEA	⩽ 64	56	2 to 4	16 to 18	256 to 258
6	DEA	⩽ 64	112	8 to 10	20 to 22	260 to 262

[a] The range of values in these three columns depends on the padding method used.

Table 4 — ISO 9797-2 — Relative performance

ISO 9797-2 algorithm	Hash-function	MAC size	Key Length	Number of round function evaluations for unpadded message size:		
				8 bytes	64 bytes	1kB
1	SHA-1, or RIPEMD-160	⩽ 160	⩽ 128	8	9	24
1	RIPEMD-128	⩽ 128	⩽ 128	8	9	24
2	SHA-1, or RIPEMD-160	⩽ 160	160...512	4	5	20
2	RIPEMD-128	⩽ 128	128...512	4	5	20
3	SHA-1, or RIPEMD-160	⩽ 80	⩽ 128	7	n/a	n/a
3	RIPEMD-128	⩽ 64	⩽ 128	7	n/a	n/a

NOTE 1 Algorithms 1 and 3 pre-computation can save 6 hash computations for a fixed key.

NOTE 2 Algorithm 3 the length of message is restricted to at most 32 bytes.

Annex A
(normative)

Approved block ciphers for message authentication

A.1 Introduction

ISO/IEC 9797-1 specifies six MAC algorithms based on block ciphers. It does not specify the block cipher itself. The purpose of this annex is to specify, either directly or through reference, the block ciphers approved by ISO/IEC 9797-1. It also defines the procedures by which block ciphers should be added to this list.

A.2 Approved block cipher: DEA

DEA is published as ANSI X3.92-1981. It is a 64-bit block cipher using a key with 56 effective bits.

A.3 Approved block cipher: T-DEA

T-DEA is published in ANSI X9.52. It is a 64-bit block cipher using a key with 112 or 168 effective bits.

A.4 Procedure for review of alternative block ciphers

A.4.1 Origination

An alternative authentication algorithm proposed for incorporation in this ISO 16609 shall be submitted by, or with the approval of, a national standards body, to the Secretariat of ISO/TC 68.

A.4.2 Justification of proposal

The originator shall justify a proposal by describing

a) the purpose the proposal is designed to serve,

b) how this purpose is better achieved by the proposal than algorithms already specified by this International Standard,

c) additional merits not described elsewhere, and

d) experience in use of the new algorithm.

A.4.3 Documentation

The proposed algorithm shall be completely documented when submitted for consideration. The documentation shall include

a) a full description of the algorithm proposed,

b) a clear acknowledgement that the algorithm satisfies, or is compatible with, all the requirements contained in this International Standard,

c) a logic flow diagram showing the processing steps used to compute the MAC,

d) a definition and explanation of any new terms, factors, or variables introduced,

e) authentication key requirements, usage, and handling,

f) a step-by-step computation example illustrating the computation of the MAC using the standard example message (see Annex C), and

g) detailed information on any prior testing to which the proposed algorithm has been subjected, particularly concerning its security, reliability and stability, and including an outline of the testing procedures used, the results of the tests, and the identity of the agency or group performing the tests and certifying the results — i.e. sufficient information shall be provided to enable an independent agency to conduct the same tests and to compare the results achieved.

A.4.4 Public disclosure

Any algorithm submitted for consideration shall be free from security classification. If copyright patent application has been made on the algorithm, it shall be assessed in accordance with IEC/ISO procedures. All documentation of the algorithm shall be considered public information available to any individual, organization or agency for review and testing.

A.4.5 Examination of proposals

Each new proposal shall be examined by ISO and a report on it prepared within 180 days of receipt (see A.4.6). The report shall state whether the proposal is adequately documented, if it has been properly tested and certified already, and if the proposed algorithm satisfies the conditions and requirements of the International Standard. The examination may also recommend submission of the proposal for public review (A.4.6).

A.4.6 Public review

When the report recommends that public review is necessary, proposals considered suitable for acceptance shall be forwarded (with the consent of the originator) to selected agencies and institutions with an international reputation in this field. These agencies and institutions will be requested to examine and report on the proposals within 90 days of receipt.

NOTE This period of public review could extend the 180 days allowed for preparation of the report on the proposal (see A.4.5).

A.4.7 Appeal procedure

Originators whose proposals are rejected (see A.4.5) may ask the Secretariat of ISO/TC 68 to have the proposals subjected to public review (see A.4.6), if this has not already been done. If, following submission of the public review reports, rejection is still recommended, the originator may request the TC 68 Secretariat to circulate the proposal, together with copies of all relevant reports on it, for ballot by the P-members of the technical committee whose ruling in the matter by a simple majority of those voting shall be final.

A.4.8 Incorporation of new authentication algorithms

New algorithms for authentication recommended for acceptance, together with relevant reports on them, shall be circulated for letter ballot as proposed amendments to ISO 16609.

A.4.9 Maintenance

An algorithm approved by the method described in this International Standard shall be reviewed by ISO/TC 68 at intervals of not greater than five years.

Annex B
(informative)

Message authentication for coded character sets

B.1 Format options

This annex offers five options for the coding of data to be authenticated:

— binary data (see B.3);

— coded characters (see B.4) entire message text, no editing;

— coded characters (see B.5) extracted message elements, no editing;

— coded characters (see B.6) entire message text, editing;

— coded characters (see B.7) extracted message elements, editing.

Option 1 is designed for the authentication of a binary string of data.

Options 2 and 3 are designed for the authentication of data in coded character sets whenever the transmission medium provides character set transparency [e.g. systems and networks designed in accordance with the open systems interconnection (OSI) model].

Options 4 and 5 are designed for the authentication of data in a restricted coded character set for use whenever the transmission medium is not transparent to the character set being used (e.g. baudot, telex, and store and forward services such as those provided by many international record carriers).

Choice of the format option is the responsibility of the correspondents and will be subject to bilateral agreement.

As noted in ISO 9797-1:1999, Annex B, it is important to protect against xor forgery attacks when using Algorithm 1 with Padding Method 1 (or 2). This might be achieved, for instance, by the recipient knowing the length of the message or the number of delimited fields in the message.

B.2 Code character sets (as used in Options 2 to 5)

B.2.1 Defined message element formats

B.2.1.1 General

The field formats for DMC, IDA, MAC and MID are represented in the form specified in this International Standard. Formats of other message elements are not specified.

The field formats are verified as part of the authentication process. If an authentication option that employs editing is used, then the field formats will be verified prior to editing. If a formatting error occurs, the message will fail to authenticate. The following field formats are defined.

B.2.1.2 DMC

The date on which the sending institution originates the message is expressed in accordance with ISO 8601 as century, year, month, day (preferably compacted, i.e., CCYYMMDD)

EXAMPLE 19851101 for 1 November 1985.

B.2.1.3 IDA

This field is the identifier of the key for authentication, according to the requirements for key identifiers specified in ISO 8732;

B.2.1.4 MAC

The MAC is expressed as hexadecimal digits of up to four groups of four characters, each group separated by a space (hhhhbhhhhbhhhhbhhhh).

EXAMPLE 5A6Fb09C3bCD86b1FA4.

B.2.1.5 MID

The message identifier is expressed as one to sixteen printable characters (AAAAAAAAAAAAAAAA). Permitted characters are 0 to 9, A to Z (upper case), space (b), comma (,), full stop (.), solidus (/) asterisk (*) and hyphen (-).

EXAMPLE FN-BC/2.5.

B.2.2 Implicit field delimiters

Implicit delimitation of an authentication element may be achieved if its position in the message is fixed or unambiguously identified by standardized format rules. Field names, numbers, or identifying field tags, where specified by the wire service as implicit delimiters, are processed for authentication.

B.2.3 Explicit field delimiters

B.2.3.1 General

Explicit delimiters may be used to identify the beginning and end of message elements, including the MAC. They may be used in all coded character set options. The following explicit delimiters are specified.

B.2.3.2 DMC

QD- and -DQ

EXAMPLE QD-YYMMDD-DQ.

B.2.3.3 IDA

QK- and -KQ

EXAMPLE QK-1 357BANKATOBANKB-KQ.

B.2.3.4 MAC

QM- and -MQ

EXAMPLE QM-hhhhbhhhh-MQ.

B.2.3.5 MID

QX- and -XQ

EXAMPLE QX-aaaaaaaaaaaa-XQ.

B.2.3.6 Other message elements

QT- and –TQ.

EXAMPLE QT-text-TQ

The “text” delimited In QT-text-TQ may be of any length allowed by the wire service.

B.2.4 Use of delimiters

Beginning and ending delimiters, when present, occur in complementary pairs without intervening explicit delimiters.

NOTE If this condition is not satisfied, the message will fail to authenticate.

The message may contain any number of delimited “text” fields; however, the DMC, MID, IDA, and MAC fields do not appear more than once each in a message.

The hyphen (-) appears in all explicit delimiters.

B.2.5 Character representation

All characters of authentication elements input to the algorithm are represented as 8-bit characters comprising the 7-bit code according to ISO 646 (excluding national character assignments) preceded by a zero (e.g. 0, b7, b6, ... b1). Where this necessitates a code translation, the translation is for internal computational purposes only. If the message is transformed into a different character set, the inverse transformation must be applied before beginning the authentication process.

B.2.6 Header and trailer information

Header and trailer message information added (e.g., by a network) for transmission purposes is omitted — i.e. it is not part of the message text nor is it included in the algorithm calculation.

B.3 Option 1: Binary data

The authentication algorithm is applied to the entire message text, or to parts of the message text, by agreement between sender and receiver.

B.4 Option 2: Coded characters; entire message; no editing

Where message processing is automated and the precise content of the body of the message does not change between sender and receiver, the algorithm can be applied to the entire message.

The MAC is computed over the entire message text (see example in Annex C).

B.5 Option 3: Coded characters; extracted message elements; no editing

Where authentication of the entire message is impractical, the authentication algorithm is applied only to the selected message elements.

A MAC is computed on the extracted elements, taken in the order in which they appear (see example in Annex C).

Message elements to be authenticated are extracted according to the following rules.

a) Delete all characters other than the message elements and their corresponding delimiters.

b) Insert a single space after each implicitly delimited message element.

B.6 Option 4: Coded characters; entire message; editing

The MAC is computed on the message text following editing according to the following rules (see example in Annex C), which apply, in the sequence shown, on all message elements — implicitly and explicitly delimited — before processing by the authentication algorithm.

a) Each carriage return and each line feed is replaced by a single space.

b) Lower case alphabetic characters (a to z) are translated to upper case (A to Z).

c) Any characters other than the letters A to Z, digits 0 to 9, space, comma (,), full stop (.), solidus (/), asterisk (*), open and closed parentheses, and hyphen (-) are deleted; thus end-of-text, and other formatting and control characters, are deleted.

d) All leading spaces are deleted.

e) Each sequence of consecutive spaces (internal and trailing) are replaced by a single space.

B.7 Option 5: Coded characters; extracted message elements; editing

This option is used in the same way as Option 3.

Message elements are extracted following the rules given for Option 3.

The editing rules specified for Option 4 are applied.

B.8 "Failed" MAC

B.8.1 Inability to generate MAC

When the MAC is automatically generated, i.e. by automatic extraction of authentication elements, the process can fail because of rule violations (e.g. because of nested delimiters). In that event, where human readability is required (e.g. paper, screen, or microfiche) as a minimum, the failure is indicated by eight spaces (if available) written in two groups of four, separated by a character that is not a hexadecimal digit, preferably an asterisk [e.g. where spaces are not available, zeros will be substituted (i.e. 0000*0000)].

B.8.2 Received MAC does not authenticate

When a received MAC does not equal the reference MAC generated during the authentication process, where human readability is required, failure to authenticate is indicated by the insertion of a non-hexadecimal

printable character in place of the space in the received MAC. Where available in the character set, an asterisk is used, for example, 5A6F*09C3.

B.9 Authentication keys

Authentication keys are secret cryptographic keys that have been previously exchanged by the sender and receiver and which are used by the authentication algorithm. Such keys are randomly or pseudo-randomly generated (see Annex F). Keys used for message authentication are not to be used for any other purpose. Any key used for authentication is to be protected against disclosure to unauthorized parties.

Annex C
(informative)

Examples of message authentication for coded characters sets

C.1 Overview of MAC examples

C.1.1 General

This annex gives examples of message authentication for coded character sets using DEA and T-DEA. The examples illustrate use of the ISO/IEC 9797 MAC algorithms, as indicated in Table C.1.

NOTE All computations included in this annex have been performed at the individual block level using ECB and XOR operations, and the results confirmed using CBC on the complete set of data blocks.

Table C.1 — Overview of MAC calculation examples

Example	Message elements	ISO 9797-1 algorithm	Padding method	Block cipher	Block size n	Effective key bits	MAC size m
1	All	1	1	T-DEA	64	112	$32 \leqslant m \leqslant 64$
2	Selected	1	1	T-DEA	64	112	$32 \leqslant m \leqslant 64$
3	All	3	1	DEA	64	112	$32 \leqslant m \leqslant 64$
As noted in Annex B of ISO 9797-1, it is important to protect against xor-forgery attacks when using Algorithm 1 with Padding Method 1 (or 2). This might be achieved, for instance, by the recipient knowing the length of the message or the number of delimited fields in the message.							

The examples use a transaction-oriented message that could be generated by an ATM and includes an encrypted PIN block.

Examples 1 and 3 use the entire message for the MAC computation. Only the message text (whole body), not the protocol-related fields such as header, are used. Example 2 illustrates a MAC computation using only selected fields of the message.

As specified in 5.1, the authentication algorithms use the CBC mode of operation.

The notation for keys and data blocks used in the examples is consistent with ISO/IEC 9797-1.

C.1.2 Assumed pre-defined agreement

The authentication elements are expressed as the hexadecimal representation of ASCII characters (two hexadecimal digits per character). The hexadecimal representation of the resulting MAC may be converted into ASCII characters for transmission — each hexadecimal digit of the MAC being transmitted as an ASCII character from the set 0 to 9, A to F.

NOTE In other environments, the pre-defined agreement might specify a different representation for the authentication elements, and also for the transmitted MAC — for example, in a bit-oriented protocol, binary representation might be used in both cases, reducing both MAC computation time and MAC transmission time.

C.1.3 Sample input message

All three examples are based on the following input message text (ASCII) where the ¶ symbol is used to denote a field separator:

11¶918273645¶¶58143276¶¶;1234567890123456=991210000?¶00012500¶9786534124876923¶

A brief description of the fields of the sample message follows.

Table C.2 — Sample input message

Field Name	Description	Value
Message Type	Code by which the terminal indicates the type of message being sent.	11
Terminal ID	Number by which the terminal is identified to the network.	918273645
Time Variant No.	Number (or value) which changes with each message or transaction.	58143276
Track 2 Data	Information encoded on Track 2 of a consumer access card. The content of this field is expanded below.	;1234567890123456=991210000?
Transaction Data	Field in which the terminal informs the network of the type and value of transaction requested.	00012500
Encrypted PIN Block	Field in which the consumer-entered PIN is transmitted to the network in encrypted form.	9786534124876923

Table C.3 — Example of Track 2 Data

Start sentinel (SS)	;
Primary account number	1234567890123456
Field separator	=
Expiry date	9912
Discretionary data	10000
End sentinel (ES)	?

C.2 MAC computation Example 1

Example 1 uses the entire message text (which may be regarded as a single authentication element) for MAC computation. For this example, the pre-defined agreement specified inclusion of separators and of all card-encoded (Track 2) data from start sentinel (SS) to end sentinel (ES).

Cryptographic Key (Hex): K = 0123 4567 89AB CDEF FEDC BA98 7654 3210

First Data Block (Hex): 31311C3931383237 (this is the hex representation of ASCII 11¶91827)

All data is represented as hexadecimal.

Iteration (x)	DATA BLOCKS (D_x)	T-DEA INPUT BLOCK (D_x xor H_{x-1})	IV / T-DEA OUTPUT BLOCK (H_x)
0			0000000000000000
1	31311C3931383237	31311C3931383237	827E153B886163D2
2	333634351C1C3538	B148210E947D56EA	00A37ACBAD184184
3	3134333237361C1C	319749F99A2E5D98	1AE4BE256716410E
4	3B31323334353637	21D58C1653237739	2F195D24CD861FA4
5	3839303132333435	17206D15FFB52B91	35B8FF7899281997
6	363D393931323130	0385C641A81A28A7	E156D31014362301
7	3030303F1C303030	D166E32F08061331	49FE9E6E54743E43
8	31323530301C3937	78CCAB5E64680774	4B7E8111049919F3
9	3836353334313234	7348B42230A82BC7	E355B6FF76CFFF03
10	3837363932331C00	DB6280C644FCE303	F7B47FFBD1720C55

NOTE 1 The first data block D1 is input directly to the T-DEA. Alternatively, H0 can be set to 0000000000000000 — by setting the IV for the CBC mode of operation.

NOTE 2 The final data block includes 7 characters from the message and a padding byte of 00.

NOTE 3 The 32-bit MAC is F7B47FFB. A longer MAC can be extracted from the final output block using additional bits.

C.3 MAC computation Example 2

Example 2 outlines a MAC computation using only the following selected message elements:

Time Variant No. (or value)

Account Number (PAN) from Track 2 of the consumer card

Transaction Data

Encrypted PIN Block

For this example, the pre-defined agreement specified inclusion of separators and start sentinels in the authentication elements. The authentication calculation is based on the following input message text (ASCII) where the ¶ symbol is used to denote a field separator:

58143276¶;1234567890123456=¶00012500¶9786534124876923¶

Cryptographic Key (Hex): K = 0123 4567 89AB CDEF FEDC BA98 7654 3210

First Data Block (Hex): 3538313433323736 (this is the hex representation of ASCII 58143276)

All data is represented as hexadecimal.

Iteration (x)	DATA BLOCKS (D_x)	T-DEA INPUT BLOCK (D_x xor H_{x-1})	IV / T-DEA OUTPUT BLOCK (H_x)
0			0000000000000000
1	3538313433323736	3538313433323736	46813E6FA5BFB3B0
2	1C3B313233343536	5ABA0F5D968B8686	E3A6673630EF0C1E
3	3738393031323334	D49E5E0601DD3F2A	21644229B112881E
4	35363D1C30303031	14527F358122B82F	84FBC45C0F95DF19
5	323530301C393738	B6CEF46C13ACE821	3F9C8473CDF66468
6	3635333431323438	09A9B747FCC45050	1DBF9E759DF842CD
7	37363932331C0000	2A89A747AEE442CD	6B64A37C973A1548

NOTE 1 The first data block D_1 is input directly to the T-DEA. Alternatively, H_0 can be set to 0000000000000000 by setting the IV for the CBC mode of operation.

NOTE 2 The final data block includes 6 characters from the message and 2 padding bytes of 00.

NOTE 3 The 32-bit MAC is 6B64A37C. A longer MAC can be extracted from the final output block using additional bits.

C.4 MAC computation Example 3

While Example 1 uses the entire message text (which may be regarded as a single authentication element) for MAC computation, for this example, the pre-defined agreement specified inclusion of separators and of all card encoded (Track 2) data from start sentinel (SS) to end sentinel (ES).

Cryptographic Keys (Hex): K = 0123 4567 89AB CDEF
K' = FEDC BA98 7654 3210

First Data Block (Hex): 31311C3931383237 (this is the hex representation of ASCII 11¶91827)

All data is represented as hexadecimal.

Iteration (x)	DATA BLOCKS (D_x)	DEA INPUT BLOCK (D_x xor H_{x-1})	IV / DEA OUTPUT BLOCK (H_x)
0			0000000000000000
1	31311C3931383237	31311C3931383237	356C20A9E60304D9
2	333634351C1C3538	065A149CFA1F31E1	BE3EDA28E5A358EA
3	3134333237361C1C	8F0AE91AD29544F6	D451B35100C56A84
4	3B31323334353637	EF60816234F05CB3	BCF794DAA6BB0FFE
5	3839303132333435	84CEA4EB94883BCB	3622C2A8A5F73F94
6	363D393931323130	001FFB9194C50EA4	EA776E4F7064C650
7	3030303F1C303030	DA475E706C54F660	2ABFE53C0CA6C57D
8	31323530301C3937	1B8DD00C3CBAFC4A	0EBF212FA1E0EBB2
9	3836353334313234	3689141C95D1D986	65603056F90CA687
10	3837363932331C00	5D57066FCB3FBA87	C156F1B8CDBFB451
		C156F1B8CDBFB451	CCCD3C0841F6C7AB
		CCCD3C0841F6C7AB	C209CCB78EE1B606

NOTE 1 The first data block D_1 is input directly to the DEA. Alternatively, H_0 can be set to 0000000000000000 by setting the IV for the CBC mode of operation.

NOTE 2 The final data block includes 7 characters from the message and a padding byte of 00.

NOTE 3 The final iteration includes the additional decrypt with K' and encrypt with K.

NOTE 4 The 32-bit MAC is C209CCB7. A longer MAC can be extracted from the final output block using additional bits.