INTERNATIONAL STANDARD

ISO/IEC 9797-1

Second edition
2011-03-01

Information technology — Security techniques — Message Authentication Codes (MACs) —

Part 1: Mechanisms using a block cipher

Technologies de l'information — Techniques de sécurité — Codes d'authentification de message (MAC) —
Partie 1: Mécanismes utilisant un chiffrement par blocs

Reference number
ISO/IEC 9797-1:2011(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) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national 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 and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 9797-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 9797-1:1999), which has been technically revised. MAC Algorithms 5 and 6 of ISO/IEC 9797-1:1999, which consisted of two single CBC-MAC computations, have been replaced by two other MAC algorithms, which perform single CBC-MAC computations and which offer improved efficiency. Annex A on object identifiers has been added. The security analysis in Annex C has been updated and Annex D on the relationship to previous standards has been added.

ISO/IEC 9797 consists of the following parts, under the general title *Information technology — Security techniques — Message Authentication Codes (MACs)*:

— *Part 1: Mechanisms using a block cipher*

— *Part 2: Mechanisms using a dedicated hash-function*

— *Part 3: Mechanisms using a universal hash-function*

Further parts may follow.

Introduction

In an IT environment, it is often required that one can verify that electronic data has not been altered in an unauthorized manner and that one can provide assurance that a message has been originated by an entity in possession of the secret key. A MAC (Message Authentication Code) algorithm is a commonly used data integrity mechanism that can satisfy these requirements.

This part of ISO/IEC 9797 specifies six MAC algorithms that are based on an n-bit block cipher. They compute a short string as a function of a secret key and a message of variable length.

The strength of the data integrity mechanism and message authentication mechanism is dependent on the length (in bits) k^* 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 mechanism.

The first mechanism specified in this part of ISO/IEC 9797 is commonly known as CBC-MAC (CBC is an abbreviation of Cipher Block Chaining).

The other five mechanisms are variants of CBC-MAC. MAC Algorithms 2, 3, 5 and 6 apply a special transformation at the end of the processing. MAC Algorithm 6 is an optimized variant of MAC Algorithm 2. MAC Algorithm 5 uses the minimum number of encryptions. MAC Algorithm 5 requires only a single block cipher key setup but it needs a longer internal key. MAC Algorithm 4 applies a special transformation at both the beginning and the end of the processing; this algorithm is recommended for use in applications which require that the key length of the MAC algorithm be twice that of the block cipher.

Information technology — Security techniques — Message Authentication Codes (MACs) —

Part 1:
Mechanisms using a block cipher

1 Scope

This part of ISO/IEC 9797 specifies six MAC algorithms that use a secret key and an n-bit block cipher to calculate an m-bit MAC.

This part of ISO/IEC 9797 can be applied to the security services of any security architecture, process, or application.

Key management mechanisms are outside the scope of this part of ISO/IEC 9797.

This part of ISO/IEC 9797 specifies object identifiers that can be used to identify each mechanism in accordance with ISO/IEC 8825-1. Numerical examples and a security analysis of each of the six specified algorithms are provided, and the relationship of this part of ISO/IEC 9797 to previous standards is explained.

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/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

3 Terms and definitions

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

3.1
block
bit string of length n

3.2
block cipher key
key that controls the operation of a block cipher

3.3
ciphertext
data which has been transformed to hide its information content

[ISO/IEC 9798-1:2010]

3.4
data integrity
property that data has not been altered or destroyed in an unauthorized manner

[ISO 7498-2]

3.5
decryption
reversal of a corresponding encryption

[ISO/IEC 9798-1:2010]

3.6
encryption
reversible operation by a cryptographic algorithm converting data into ciphertext so as to hide the information content of the data

[ISO/IEC 9798-1:2010]

3.7
key
sequence of symbols that controls the operation of a cryptographic transformation

NOTE Examples are encryption, decryption, cryptographic check function computation, signature generation, or signature verification.

[ISO/IEC 9798-1:2010]

3.8
MAC algorithm key
key that controls the operation of a MAC algorithm

3.9
Message Authentication Code
MAC
string of bits which is the output of a MAC algorithm

NOTE A MAC is sometimes called a cryptographic check value (see for example ISO 7498-2 [1]).

3.10
Message Authentication Code algorithm
MAC algorithm
algorithm for computing a function which maps strings of bits and a secret key to fixed-length strings of bits, satisfying 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 a set of input strings and corresponding function values, where the value of the ith input string might have been chosen after observing the value of the first i -1 function values (for integers i > 1)

NOTE 1 A MAC algorithm is sometimes called a cryptographic check function (see for example ISO 7498-2 [1]).

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

3.11
***n*-bit block cipher**
block cipher with the property that plaintext blocks and ciphertext blocks are n bits in length

[ISO/IEC 10116]

3.12
output transformation
function that is applied at the end of the MAC algorithm, before the truncation operation

3.13
plaintext
unencrypted information

NOTE Adapted from ISO/IEC 9798-1:2010.

4 Symbols and notation

Throughout this part of ISO/IEC 9797 the following symbols and notation are used:

CT_i — n-bit binary representation of the integer i.

D — data string to be input to the MAC algorithm.

D_j — block derived from the data string D after the padding and splitting process.

$d_K(C)$ — decryption of the ciphertext C with the block cipher e using the key K.

$e_K(P)$ — encryption of the plaintext P with the block cipher e using the key K.

F — final iteration.

g — output transformation that maps the block H_q to the block G.

G — block that is the result of the output transformation.

$GF(2^n)$ — finite field with exactly 2^n elements.

$H_0,H_1,\ldots,H_q$ — blocks used in the MAC algorithm to store intermediate results.

k — length (in bits) of the block cipher key.

k^* — length (in bits) of the MAC algorithm key.

K, K', K'' — secret block cipher keys of length (in bits) k.

K_1, K_2 — secret masking keys of length (in bits) n.

L — length block, used in Padding Method 3, equal to the binary representation of the length of the input message, left-padded to form an n-bit block.

L_D — length (in bits) of the data string D.

m — length (in bits) of the MAC.

multx(T) — operation on an n-bit string T defined as $T*x$, where T is treated as an element in the finite field $GF(2^n)$, and is multiplied by the element corresponding to the monomial x in $GF(2^n)$. It can be computed as follows, where T_{n-1} denotes the leftmost bit of T and, as defined below, << denotes a one-bit left shift operation.

$$\text{multx}(T) = \begin{cases} T \ll 1 & \text{if } T_{n-1} = 0 \\ (T \ll 1) \oplus \tilde{P}_n & \text{if } T_{n-1} = 1 \end{cases}$$

n — block length (in bits) of the block cipher.

$p_n(x)$ irreducible polynomials of degree n over $\mathrm{GF}(2)$, that is, polynomials with no non-trivial divisors.

$\tilde{p}_n$ string of bits of length n, consisting of the rightmost n coefficients (corresponding to x^{n-1}, x^{n-2}, …, x, x^0 = 1) of the irreducible polynomial $p_n(x)$. For n=128, $p_n(x) = x^{128} + x^7 + x^2 + x + 1$, and $\tilde{p}_{128}$ = $0^{120}10000111$. For n=64, $p_n(x) = x^{64} + x^4 + x^3 + x + 1$, and $\tilde{p}_{64}$ = $0^{59}11011$.

q number of blocks in the data string D after the padding and splitting process.

S secret string of length (in bits) n.

S_1, S_2 secret strings of length (in bits) $t \cdot n$.

t smallest integer greater than or equal to k/n.

$j \sim X$ string obtained from the string X by taking the leftmost j bits of X.

$X \oplus Y$ exclusive-or of bit-strings X and Y.

$X \| Y$ concatenation of bit-strings X and Y (in that order).

0^n string consisting of n zero bits.

:= symbol denoting the 'set equal to' operation, used in the procedural specifications of MAC algorithms, where it indicates that the value of the string on the left side of the symbol shall be made equal to the value of the expression on the right side of the symbol.

$*$ finite field multiplication. In the polynomial representation, each element of $\mathrm{GF}(2^n)$ is represented by a binary polynomial of degree less than n. More explicitly the bit string $A = a_{n-1} \ldots a_2 a_1 a_0$ is mapped to the binary polynomial $a(x) = a_{n-1} x^{n-1} + \ldots + a_2 x_2 + a_1 x + a_0$. Multiplication in the finite field $\mathrm{GF}(2^n)$, denoted by $A*B$, corresponds to the multiplication of two polynomials $a(x)b(x)$ modulo a binary irreducible polynomial $p_n(x)$ of degree n; that is, $A*B$ is the polynomial of degree at most n-1 obtained by multiplying $a(x)$ and $b(x)$, dividing the result by $p_n(x)$, and then taking the remainder. Here $p_n(x)$ is chosen to be the lexicographically first polynomial from among the irreducible polynomials of degree n that have a minimum number of non-zero coefficients. For n=128, $p_n(x) = x^{128} + x^7 + x^2 + x + 1$.

$X{<<}1$ string obtained from the string X by a left shift of 1 bit; if the length of X is n bits then $X{<<}1$ is the string consisting of the rightmost n bits of $X \| 0$.

5 Requirements

Users who wish to employ a MAC algorithm from this part of ISO/IEC 9797 shall select:

— a block cipher e, either one of those specified in ISO/IEC 18033-3 or the DEA block cipher (specified in Annex A of ISO/IEC 18033-3:2005 and ANSI X3.92 [10]). DEA may only be used with MAC Algorithms 3 and 4;

— a padding method from amongst those specified in 6.3;

— a MAC algorithm from amongst those specified in Clause 7;

— the length (in bits) m of the MAC; and

— a common key derivation method if MAC Algorithm 4 is used; a common key derivation method may also be required for MAC Algorithms 2 and 6.

Agreement on these choices amongst the users is essential for the purpose of the operation of the data integrity mechanism.

The length m of the MAC shall be a positive integer less than or equal to the block length n.

If Padding Method 3 is used, the length in bits of the data string D shall be less than 2^n.

If MAC Algorithm 4 is used, the number of blocks in the padded version of the data string shall be greater than or equal to two, i.e., $q \geq 2$.

The selection of a specific block cipher *e*, padding method, MAC algorithm, value for m, and key derivation method (if any) are beyond the scope of this part of ISO/IEC 9797.

NOTE 1 These choices affect the security level of the MAC algorithm. For a detailed discussion, see Annex C.

The same key shall be used for calculating and verifying the MAC. If the data string is also being encrypted, the key used for the calculation of the MAC shall be different from that used for encryption.

NOTE 2 It is considered to be good cryptographic practice to have independent keys for confidentiality and for data integrity.

The security of the MAC algorithms in this part of ISO/IEC 9797 is critically dependent on the procedures and practices followed to manage the keys. Information about key management can be found in ISO 8732 [3], ISO/IEC 11770 [8] and ISO 11568 [9].

Disclosure of intermediate values during the computation of the MAC algorithms may enable forgery and/or key recovery attacks (cf. Annex C).

6 Model for MAC algorithms

6.1 General

The application of the MAC algorithm requires the following seven steps: key derivation (optional), padding, splitting, iterative application of the block cipher, final iteration, output transformation, and truncation. Steps 4 through 7 are illustrated in Figure 1.

NOTE In MAC Algorithm 4, the first stage of the iteration (Step 4) is different from the other stages.

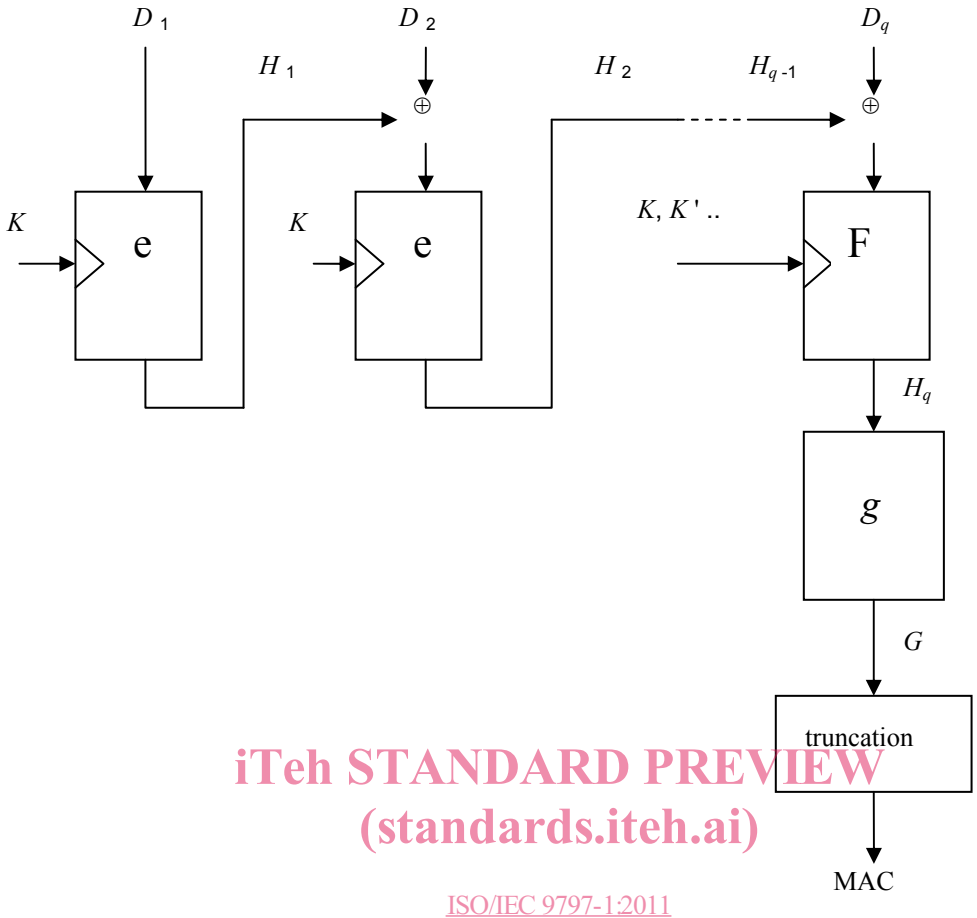

Figure 1 — Application of steps 4, 5, 6 and 7 of the MAC algorithm

6.2 Step 1 (key derivation)

6.2.1 General

MAC Algorithm 5 uses a key derivation algorithm which derives two masking keys from a block cipher key. MAC Algorithms 2, 4 and 6 may need a key derivation algorithm, which derives two block cipher keys from a block cipher key.

This part of ISO/IEC 9797 specifies two key derivation algorithms.

6.2.2 Key Derivation Method 1

This key derivation method computes two block cipher keys K' and K'', each of length (in bits) k, from a block cipher key K.

This key derivation method uses the Counter Method (CTR) defined in ISO/IEC 10116 [7]. It consists of the following operations:

— Define the integer t as the smallest integer greater than or equal to k/n.

— Define the counter CT_i, $1 \le i \le 2t$ as the string consisting of the binary representation of the integer i left-padded with as few (possibly none) '0' bits as necessary to obtain an n-bit block.

— Compute the string S_1 of length (in bits) tn equal to $e_K(CT_1)||e_K(CT_2)||\ldots||e_K(CT_t)$ and set $K' := k \sim S_1$.

— Compute the string S_2 of length (in bits) tn equal to $e_K(CT_{t+1})||e_K(CT_{t+2})||\ldots||e_K(CT_{2t})$ and set $K'' := k \sim S_2$.

6.2.3 Key Derivation Method 2

This key derivation method computes two masking keys K_1 and K_2 of length in bits n from a block cipher key.

It consists of the following operations:

— First the secret string S of length in bits n is computed as follows: $S := e_K(0^n)$.

— Next the masking key K_1 is obtained from S : $K_1 :=$ multx(S).

— Finally the masking key K_2 is derived from K_1 : $K_2 :=$ multx(K_1).

6.3 Step 2 (padding)

6.3.1 General

This step involves prefixing and/or postfixing the data string D with additional 'padding' bits such that the padded version of the data string will always be a multiple of n bits in length. The padding bits that are added to the original data string, according to the chosen padding method, are only used for calculating the MAC. Consequently, these padding bits (if any) need not be stored or transmitted with the data. The verifier shall know whether or not the padding bits have been stored or transmitted, and which padding method is in use.

This part of ISO/IEC 9797 specifies four padding methods. Padding methods 1, 2 and 3 can be chosen for MAC Algorithms 1, 2, 3, 4, and 6 specified in this part of ISO/IEC 9797. Padding method 4 shall only be used with MAC Algorithm 5.

6.3.2 Padding Method 1

The data string D to be input to the MAC algorithm shall be right-padded with as few (possibly none) '0' bits as necessary to obtain a data string whose length (in bits) is a positive integer multiple of n.

NOTE 1 MAC algorithms using Padding Method 1 may be subject to trivial forgery attacks. See informative Annex C for further details.

NOTE 2 If the data string is empty, Padding Method 1 specifies that it is right-padded with n `0' bits.

6.3.3 Padding Method 2

The data string D to be input to the MAC algorithm shall be right-padded with a single '1' bit. The resulting string shall then be right-padded with as few (possibly none) '0' bits as necessary to obtain a data string whose length (in bits) is a positive integer multiple of n.

NOTE If the data string is empty, Padding Method 2 specifies that it is right-padded with a single '1' bit followed by n - 1 '0' bits.

6.3.4 Padding Method 3

The data string D to be input to the MAC algorithm shall be right-padded with as few (possibly none) '0' bits as necessary to obtain a data string whose length (in bits) is a positive integer multiple of n. The resulting string shall then be left-padded with a block L. The block L consists of the binary representation of the length (in bits) L_D of the unpadded data string D, left-padded with as few (possibly none) '0' bits as necessary to obtain an n-bit block. The right-most bit of the block L corresponds to the least significant bit of the binary representation of L_D.

NOTE 1 Padding Method 3 is not suitable for use in situations where the length of the data string is not available prior to the start of the MAC calculation.

NOTE 2 If the data string is empty, Padding Method 3 specifies that it is right-padded with n '0' bits and left-padded with a block L consisting of n '0' bits.