INTERNATIONAL STANDARD

ISO/IEC 9797-2

Second edition
2011-05-01

Corrected version
2011-06-15

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

Part 2: Mechanisms using a dedicated hash-function

Technologies de l'information — Techniques de sécurité — Codes d'authentification de message (MAC) —

Partie 2: Mécanismes utilisant une fonction de hachage dédiée

Reference number
ISO/IEC 9797-2:2011(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.

ISO/IEC 9797-2 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 9797-2:2002), which has been technically revised by including MAC algorithms based on Dedicated Hash-Functions 4 – 7 of ISO/IEC 10118-3:2004 and Dedicated Hash-Function 8 of ISO/IEC 10118-3/Amd.1:2006.

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.

This corrected version of ISO/IEC 9797-2:2011 incorporates corrections to subclauses 3.14, 6.3, 6.3.5 and 6.3.6.

Introduction

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning MAC Algorithm 1 (MDx-MAC) given in Clause 6.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he is willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from:

Entrust Technologies, Technology Licensing Dept., 1000 Innovation Drive, Ottawa, Ontario, Canada K2K 3E7.

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

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

Part 2:
Mechanisms using a dedicated hash-function

1 Scope

This part of ISO/IEC 9797 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. These mechanisms can be used as data integrity mechanisms to verify that data has not been altered in an unauthorized manner. They can also be used as message authentication mechanisms to provide assurance that a message has been originated by an entity in possession of the secret key. The strength of the data integrity and message authentication mechanisms is dependent on the entropy and secrecy of the key, on the length (in bits) *n* of a hash-code produced by the hash-function, on the strength of the hash-function, on the length (in bits) *m* of the MAC, and on the specific mechanism.

The three mechanisms specified in this part of ISO/IEC 9797 are based on the dedicated hash-functions specified in ISO/IEC 10118-3. The first mechanism is commonly known as MDx-MAC. It calls the hash-function once, but it makes a small modification to the round-function in the hash-function by adding a key to the additive constants in the round-function. The second mechanism is commonly known as HMAC. It calls the hash-function twice. The third mechanism is a variant of MDx-MAC that takes as input only short strings (at most 256 bits). It offers higher performance for applications that work with short input data strings only.

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

NOTE A general framework for the provision of integrity services is specified in ISO/IEC 10181-6 [5].

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 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO/IEC 10118-3:2004/Amd.1:2006, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions — Amendment 1: Dedicated Hash-Function 8 (SHA-224)*

3 Terms and definitions

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

3.1
block
bit-string of length L_1, i.e. the length of the first input to the round-function

[ISO/IEC 10118-3]

3.2
collision-resistant hash-function
hash-function satisfying the following property:

- it is computationally infeasible to find any two distinct inputs which map to the same output

[ISO/IEC 10118-1]

3.3
entropy
total amount of information yielded by a set of bits, representative of the work effort required for an adversary to be able to reproduce the same set of bits

[ISO/IEC 18032]

3.4
input data string
string of bits which is the input to a hash-function

3.5
hash-code
string of bits which is the output of a hash-function

[ISO/IEC 10118-1]

3.6
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

- for a given output, it is computationally infeasible to find an input which maps to this output;
- for a given input, it is computationally infeasible to find a second input which maps to the same output

[ISO/IEC 10118-1]

3.7
initializing value
value used in defining the starting point of a hash-function

[ISO/IEC 10118-1]

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

[ISO/IEC 9797-1]

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]).

[ISO/IEC 9797-1]

3.10
Message Authentication Code (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 the set of input strings and corresponding function values, where the value of the *i*th input string may have been chosen after observing the value of the first *i*-1 function values (for integer $i > 1$)

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.

[ISO/IEC 9797-1]

3.11
output transformation
function that is applied at the end of the MAC algorithm, before the truncation operation

[ISO/IEC 9797-1]

3.12
padding
appending extra bits to a data string

[ISO/IEC 10118-1]

3.13
round-function
function that transforms two binary strings of lengths L_1 and L_2 to a binary string of length L_2

NOTE 1 It is used iteratively as part of a hash-function, where it combines a data string of length L_1 with the previous output of length L_2.

[ISO/IEC 10118-1]

NOTE 2 This function is also referred to as compression function in a certain hash-function text.

3.14
security strength
a number associated with the amount of work (i.e. the number of operations) that is required to break a cryptographic algorithm or system

NOTE Security strength is specified in bits, and is a specific value from the set {80, 112, 128, 192, 256}. A security strength of *b* bits means that of the order of 2^b operations are required to break the system.

3.15
word
string of 32 bits used in Dedicated Hash-Functions 1, 2, 3, 4 and 8, or a string of 64 bits used in Dedicated Hash-Functions 5 and 6 of ISO/IEC 10118-3

[ISO/IEC 10118-3]

4 Symbols and notation

This part of ISO/IEC 9797 makes use of the following symbols and notation defined in ISO/IEC 9797-1 [3]:

D the input data string, i.e. the data string to be input to the MAC algorithm.

m the length (in bits) of the MAC.

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

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

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

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

:= a 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.

For the purposes of this part of ISO/IEC 9797, the following symbols and notation apply:

$\overline{D}$ padded data string.

h hash-function.

h' the hash-function h with modified constants and modified IV.

$\overline{h}$ simplified hash-function h without the padding and length appending, and without truncating the round-function output (L_2 bits) to its left-most L_H bits.

NOTE 1 $\overline{h}$ shall only be applied to input strings with a length that is a positive integer multiple of L_1.

NOTE 2 The output of $\overline{h}$ should be L_2 bits rather than L_H bits; in particular, in Dedicated Hash-Functions 6 and 8 defined in ISO/IEC 10118-3, L_H is always smaller than L_2.

H', H'' strings of L_2 bits which are used in the MAC algorithm computation to store an intermediate result.

IV', IV_1, IV_2 initializing values.

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

K secret MAC algorithm key.

$K', K_0, K_1, K_2, \overline{K}, \overline{K}_1, \overline{K}_2$ secret MAC algorithm derived keys.

KT the first input string of the function ϕ' used in the output transformation step of MAC Algorithm 1.

$\tilde{L}$ the bit string encoding the message length in MAC Algorithm 3.

OPAD, *IPAD* constant strings used in MAC Algorithm 2.

R, S_0, S_1, S_2 constant strings used in the computation of the constants for MAC Algorithm 1 and MAC Algorithm 3.

T_0, T_1, T_2 constant strings used in the key derivation for MAC Algorithm 1 and MAC Algorithm 3.

U_0, U_1, U_2 constant strings used in the key derivation for MAC Algorithm 1 and MAC Algorithm 3.

ϕ' round-function with modified constants.

$K_1[i]$ the ith word of the string K_1, i.e. $K_1 = K_1[0] \,||\, K_1[1] \,||\, K_1[2] \,||\, K_1[3]$.

This part of ISO/IEC 9797 makes use of the following symbols and notation defined in ISO/IEC 10118-1:

H hash-code.

IV initializing value.

L_X length (in bits) of a bit-string X.

This part of ISO/IEC 9797 makes use of the following symbols and notation defined in ISO/IEC 10118-3:

C_i, C'_i constant words used in the round-functions.

L_1 the length (in bits) of the first of the two input strings to the round-function ϕ.

L_2 the length (in bits) of the second of the two input strings to the round-function ϕ, of the output string from the round-function ϕ, and of IV.

w the length (in bits) of a word; w is 32 when using Dedicated Hash-Functions 1, 2, 3, 4 and 8 of ISO/IEC 10118-3, and w is 64 when using Dedicated Hash-Functions 5 and 6 of ISO/IEC 10118-3.

ϕ a round-function, i.e. if X and Y are bit-strings of lengths L_1 and L_2 respectively, then $\phi(X,Y)$ is the string obtained by applying ϕ to X and Y.

Ψ the modulo 2^w addition operation, where w is the number of bits in a word. So, if A and B are words, then $A \Psi B$ is the word obtained by treating A and B as the binary representations of integers and computing their sum modulo 2^w, and the result is constrained to lie between 0 and 2^w-1 inclusive. The value of w is 32 in Dedicated Hash-Functions 1, 2, 3, 4 and 8, and 64 in Dedicated Hash-Functions 5 and 6.

5 Requirements

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

- a MAC algorithm from amongst those specified in Clauses 6, 7, and 8;
- a dedicated hash-function from the functions specified in ISO/IEC 10118-3; and
- the length (in bits) m of the MAC.

NOTE 1 The use of MAC Algorithms 1 and 3 with Dedicated Hash-Function 7 of ISO/IEC 10118-3 is not specified in this part of ISO/IEC 9797.

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

The key K used in a MAC algorithm shall have entropy that meets or exceeds the security strength to be provided by the MAC algorithm.

NOTE 2 In every case, the MAC algorithm key K shall be chosen such that every possible key is approximately equally likely to be selected.

For MAC Algorithms 1 and 2, the length m of the MAC shall be a positive integer less than or equal to the length of the hash-code L_H. For MAC Algorithm 3, the length m of the MAC shall be a positive integer less than or equal to half the length of the hash-code, i.e., $m \leq L_H/2$.

For MAC Algorithms 1 and 2, the length in bits of the input data string D shall be at most $2^{64} - 1$ when using Dedicated Hash-Functions 1, 2, 3, 4 and 8, and at most $2^{128} - 1$ when using Dedicated Hash-Functions 5 and 6. For MAC Algorithm 2, it shall be at most $2^{256} - 1$ when using Dedicated Hash-Function 7. For MAC Algorithm 3, it shall be at most 256.

The selection of a specific MAC algorithm, dedicated hash-function, and value for m is beyond the scope of this part of ISO/IEC 9797.

NOTE 3 These choices affect the security level of the MAC algorithm. For a detailed discussion, see Annex C.

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

NOTE 4 It is considered to be good cryptographic practice to have independent keys for confidentiality and for data integrity.

6 MAC Algorithm 1

NOTE 1 This clause contains a description of MDx-MAC [9] with Dedicated Hash-Functions 1 – 6 and 8. Table 1 shows the commonly known names of MDx-MAC with individual dedicated hash-functions.

Table 1 –The MDx-MAC algorithm with different Dedicated Hash-Functions

Dedicated Hash-Function:	The MDx-MAC algorithm is also known as
Dedicated Hash-Function 1	RIPEMD-160-MAC
Dedicated Hash-Function 2	RIPEMD-128-MAC
Dedicated Hash-Function 3	SHA-1-MAC
Dedicated Hash-Function 4	SHA-256-MAC
Dedicated Hash-Function 5	SHA-512-MAC
Dedicated Hash-Function 6	SHA-384-MAC
Dedicated Hash-Function 8	SHA-224-MAC

NOTE 2 The use of MAC Algorithm 1 with Dedicated Hash-Function 7 of ISO/IEC 10118-3 is not specified in this part of ISO/IEC 9797.

MAC Algorithm 1 requires one application of the hash-function to compute a MAC value, but requires that the constants in the corresponding round-function are modified.

The hash-function shall be selected from Dedicated Hash-Functions 1 – 6 from ISO/IEC 10118-3:2004, and Dedicated Hash-Function 8 from ISO/IEC 10118-3:2004/Amd.1:2006.

The bit length of the key k shall be at most 128 bits.

6.1 Description of MAC Algorithm 1

MAC algorithm 1 involves the following five steps: key expansion, modification of the constants and the *IV*, hashing operation, output transformation, and truncation.

6.1.1 Step 1 (key expansion)

If K is shorter than 128 bits, concatenate K to itself $\lceil 128/K \rceil$ times and select the leftmost 128 bits of the result to form the 128-bit key K' (if the length (in bits) of K is equal to 128, $K' := K$):

$K' := 128 \sim (K \,||\, K \,||\, \ldots \,||\, K)$.

Compute the subkeys K_0, K_1, and K_2 as follows:

$K_0 := \overline{h}\,(K' \,||\, U_0 \,||\, K')$
$K_1 := 128 \sim \overline{h}\,(K' \,||\, U_1 \,||\, K')$, when using Dedicated Hash-Functions 1, 2 and 3
$K_1 := 256 \sim \overline{h}\,(K' \,||\, U_1 \,||\, K')$, when using Dedicated Hash-Functions 4, 5, 6 and 8
$K_2 := 128 \sim \overline{h}\,(K' \,||\, U_2 \,||\, K')$.

Here U_0, U_1, and U_2 are 768-bit constants that are defined in Clause 6.3, and $\overline{h}$ denotes a simplified hash-function h, i.e., without the padding and length appending, and without truncating the round-function output (L_2 bits) to its left-most L_H bits.

NOTE 1 Padding and length appending are omitted because in this case the length of the input string is either L_1 bits or $2L_1$ bits.

NOTE 2 Truncation is omitted because in this case the length of K_0 is always L_2 bits, which is at least L_H.

When using Dedicated Hash-Functions 1, 2, 3, 5 and 6, the derived key K_1 is split into four words denoted by $K_1[i]$ $(0 \le i \le 3)$, i.e.

$K_1 = K_1[0] \,||\, K_1[1] \,||\, K_1[2] \,||\, K_1[3]$.

When using Dedicated Hash-Functions 4 and 8, the derived key K_1 is split into eight words denoted by $K_1[i]$ $(0 \le i \le 7)$, i.e.

$K_1 = K_1[0] \,||\, K_1[1] \,||\, K_1[2] \,||\, K_1[3] || \, K_1[4] \,||\, K_1[5] \,||\, K_1[6] \,||\, K_1[7]$.

For the conversion of a string into words, a byte ordering convention is required. The byte ordering convention for this conversion is that which is defined for the selected dedicated hash-function in ISO/IEC 10118-3.

6.1.2 Step 2 (modification of the constants and the IV)

When using Dedicated Hash-Functions 1, 2, 3, 4, 5, 6 and 8, the additive constants used in the round-function are modified by the addition mod 2^w of a word of K_1, e.g.,

$C_0 := C_0 \,\Psi\, K_1[0]$.

Clause 6.3 indicates which word of K_1 is added to each constant.

The initial value *IV* of the hash function is replaced by $IV' := K_0$. The function resulting from the changes in this step is denoted by h', and its round-function is denoted by ϕ'.

6.1.3 Step 3 (hashing operation)

The string which is input to the modified hash-function h' is equal to the input data string D, i.e.

$H' := h'(D)$.