INTERNATIONAL STANDARD

ISO/IEC 9797-2

Second edition
2011-05-01

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

Reference number
ISO/IEC 9797-2:2011(E)

© ISO/IEC 2011

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

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

Part 2: Mechanisms using a dedicated hash-function

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

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]

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

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 2^6 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]

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

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.
$X \parallel Y$	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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

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. $K_1 = K_1[0] \parallel K_1[1] \parallel K_1[2] \parallel K_1[3]$.

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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

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:

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

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.

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

6.1.4 Step 4 (output transformation)

The modified round-function ϕ' is applied one additional time, with first input the string KT (defined below) and second input the string H' (the result of Step 3) i.e.

$$H'' := \phi'(KT, H').$$

For Dedicated Hash-Functions 1, 2, 3, 4 and 8,

$$KT = K_2 \,||\, (K_2 \oplus T_0) \,||\, (K_2 \oplus T_1) \,||\, (K_2 \oplus T_2).$$

For Dedicated Hash-Functions 5 and 6,

$$KT = K_2 \,||\, (K_2 \oplus T_0) \,||\, (K_2 \oplus T_1) \,||\, (K_2 \oplus T_2) \,||\, K_2 \,||\, (K_2 \oplus T_0) \,||\, (K_2 \oplus T_1) \,||\, (K_2 \oplus T_2).$$

Here T_0, T_1, and T_2 are 128-bit strings defined in Clause 6.3 for each dedicated hash-function.

NOTE The output transformation corresponds to processing an additional data block derived from K_2 *after* padding and appending of the length field.

6.1.5 Step 5 (truncation)

The MAC of m bits is derived by taking the leftmost m bits of the string H'', i.e.

$$\text{MAC} := m \sim H''.$$

6.2 Efficiency

If the padded data string (where the padding algorithm depends on the selected hash-function) contains q blocks, then MAC Algorithm 1 requires $q + 7$ applications of the round-function.

This can be reduced to $q + 1$ applications of the round-function by pre-computing the values K_0, K_1 and K_2, and by replacing the initial value IV by IV' in the application of the hash-function. It is recommended to make this modification to the code of the hash-function together with the mandatory modification required for Step 2.

For long input strings, MAC Algorithm 1 has a performance which is comparable to that of the hash-function used.

6.3 Computation of the constants

The constants described in this clause are used in MAC Algorithms 1 and 3. MAC Algorithm 3 is specified in Clause 8.

The strings T_i and U_i $(0 \leqslant i \leqslant 2)$ are fixed elements in the description of the MAC algorithm. They are computed (only once) using the hash-function; they are different for each of the seven hash-functions.

The 128-bit constants T_i and 768-bit constants U_i are defined as follows. The definition of T_i involves the 496-bit constant R = "ab…yzAB…YZ01…89" and 16-bit constants S_0, S_1, S_2, where S_i is the 16-bit string formed by repeating twice the 8-bit representation of i (e.g., the hexadecimal representation of S_1 is `3131`). In both cases ASCII coding is used; this is equivalent to coding using ISO/IEC 646:1991.

$$\textit{for } i := 0 \textit{ to } 2 \quad T_i := 128 \sim \overline{h}\,(S_i \,||\, R)$$

$$\textit{for } i := 0 \textit{ to } 2 \quad U_i := T_i \,||\, T_{i+1} \,||\, T_{i+2} \,||\, T_i \,||\, T_{i+1} \,||\, T_{i+2}$$

where the subscripts in T_i are taken modulo 3.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

In Dedicated Hash-Functions 1, 2, 3, 4, 5, 6 and 8, for all constants C_i, C'_i and all words $K_1[i]$ the most significant bit corresponds to the left-most bit. The constants C_i and C'_i are presented using a hexadecimal representation.

6.3.1 Dedicated Hash-Function 1 (RIPEMD-160)

The 128-bit constant strings T_i for Dedicated Hash-Function 1 are defined as follows (in hexadecimal representation):

T_0 = 1CC7086A046AFA22353AE88F3D3DACEB
T_1 = E3FA02710E491D851151CC34E4718D41
T_2 = 93987557C07B8102BA592949EB638F37

Two sequences of constant words $C_0, C_1, \ldots, C_{79}$ and $C'_0, C'_1, \ldots, C'_{79}$ are used in the round-function of Dedicated Hash-Function 1. They are defined as follows:

$C_i = K_1[0]\ \Psi$ 00000000, $(0 \le i \le 15)$,
$C_i = K_1[1]\ \Psi$ 5A827999, $(16 \le i \le 31)$,
$C_i = K_1[2]\ \Psi$ 6ED9EBA1, $(32 \le i \le 47)$,
$C_i = K_1[3]\ \Psi$ 8F1BBCDC, $(48 \le i \le 63)$,
$C_i = K_1[0]\ \Psi$ A953FD4E, $(64 \le i \le 79)$,

$C'_i = K_1[1]\ \Psi$ 50A28BE6, $(0 \le i \le 15)$,
$C'_i = K_1[2]\ \Psi$ 5C4DD124, $(16 \le i \le 31)$,
$C'_i = K_1[3]\ \Psi$ 6D703EF3, $(32 \le i \le 47)$,
$C'_i = K_1[0]\ \Psi$ 7A6D76E9, $(48 \le i \le 63)$,
$C'_i = K_1[1]\ \Psi$ 00000000, $(64 \le i \le 79)$.

6.3.2 Dedicated Hash-Function 2 (RIPEMD-128)

The 128-bit constant strings T_i for Dedicated Hash-Function 2 are defined as follows (in hexadecimal representation):

T_0 = FD7EC18964C36D53FC18C31B72112AAC
T_1 = 2538B78EC0E273949EE4C4457A77525C
T_2 = F5C93ED85BD65F609A7EB182A85BA181

Two sequences of constant words $C_0, C_1, \ldots, C_{63}$ and $C'_0, C'_1, \ldots, C'_{63}$ are used in the round-function of Dedicated Hash-Function 2. They are defined as follows:

$C_i = K_1[0]\ \Psi$ 00000000, $(0 \le i \le 15)$,
$C_i = K_1[1]\ \Psi$ 5A827999, $(16 \le i \le 31)$,
$C_i = K_1[2]\ \Psi$ 6ED9EBA1, $(32 \le i \le 47)$,
$C_i = K_1[3]\ \Psi$ 8F1BBCDC, $(48 \le i \le 63)$,

$C'_i = K_1[0]\ \Psi$ 50A28BE6, $(0 \le i \le 15)$,
$C'_i = K_1[1]\ \Psi$ 5C4DD124, $(16 \le i \le 31)$,
$C'_i = K_1[2]\ \Psi$ 6D703EF3, $(32 \le i \le 47)$,
$C'_i = K_1[3]\ \Psi$ 00000000, $(48 \le i \le 63)$.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

6.3.3 Dedicated Hash-Function 3 (SHA-1)

The 128-bit constant strings T_i for Dedicated Hash-Function 3 are defined as follows (in hexadecimal representation):

T_0 = `1D4CA39FA40417E2AE5A77B49067BBCC`
T_1 = `9318AFEF5D5A5B46EFCA6BEC0E138940`
T_2 = `4544209656E14F97005DAC76868E97A3`

A sequence of constant words $C_0, C_1, \ldots, C_{79}$ is used in the round-function of Dedicated Hash-Function 3. It is defined as follows:

$C_i = K_1[0]\ \Psi$ `5A827999`, $(0 \le i \le 19)$,
$C_i = K_1[1]\ \Psi$ `6ED9EBA1`, $(20 \le i \le 39)$,
$C_i = K_1[2]\ \Psi$ `8F1BBCDC`, $(40 \le i \le 59)$,
$C_i = K_1[3]\ \Psi$ `CA62C1D6`, $(60 \le i \le 79)$.

6.3.4 Dedicated Hash-Function 4 (SHA-256)

The 128-bit constant strings T_i for Dedicated Hash-Function 4 are computed as follows (where $\overline{h}$ is the simplified version of Dedicated Hash-Function 4 defined in Clause 6.1.1):

$T_0 = 128 \sim \overline{h}\ (S_0 || R)$,
$T_1 = 128 \sim \overline{h}\ (S_1 || R)$,
$T_2 = 128 \sim \overline{h}\ (S_2 || R)$.

A sequence of constant words $C_0, C_1, \ldots, C_{63}$ is used in the round-function of Dedicated Hash-Function 4. It is defined as follows:

$C_i = K_1[i \bmod 8]\ \Psi\ C''_i$ $(0 \le i \le 63)$,

where the sequence $C''_0, C''_1, \ldots, C''_{63}$ in a hexadecimal representation (where the most significant bit corresponds to the left-most bit) is defined as follows, where the words are listed in the order $C''_0, C''_1, \ldots, C''_{63}$.

```
428a2f98 71374491 b5c0fbcf e9b5dba5 3956c25b 59f111f1 923f82a4 ab1c5ed5
d807aa98 12835b01 243185be 550c7dc3 72be5d74 80deb1fe 9bdc06a7 c19bf174
e49b69c1 efbe4786 0fc19dc6 240ca1cc 2de92c6f 4a7484aa 5cb0a9dc 76f988da
983e5152 a831c66d b00327c8 bf597fc7 c6e00bf3 d5a79147 06ca6351 14292967
27b70a85 2e1b2138 4d2c6dfc 53380d13 650a7354 766a0abb 81c2c92e 92722c85
a2bfe8a1 a81a664b c24b8b70 c76c51a3 d192e819 d6990624 f40e3585 106aa070
19a4c116 1e376c08 2748774c 34b0bcb5 391c0cb3 4ed8aa4a 5b9cca4f 682e6ff3
748f82ee 78a5636f 84c87814 8cc70208 90befffa a4506ceb bef9a3f7 c67178f2
```

NOTE These values are the first thirty-two bits of the fractional parts of the cube roots of the first sixty-four primes. They are the constant sequence used in SHA-256.

6.3.5 Dedicated Hash-Function 5 (SHA-512)

The 128-bit constant strings T_i for Dedicated Hash-Function 5 are computed as follows (where $\overline{h}$ is the simplified version of Dedicated Hash-Function 5 defined in Clause 6.1.1):

$T_0 = 128 \sim \overline{h}\ (S_0 || R)$,
$T_1 = 128 \sim \overline{h}\ (S_1 || R)$,
$T_2 = 128 \sim \overline{h}\ (S_2 || R)$.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

A sequence of constant words C_0, C_1, …, C_{79} is used in the round-function of Dedicated Hash-Function 5. It is defined as follows:

$$C_i = K_1[i \bmod 4] \Psi C''_i \quad (0 \le i \le 79),$$

where the sequence C''_0, C''_1, …, C''_{79} in a hexadecimal representation (where the most significant bit corresponds to the left-most bit) is defined as follows, where the words are listed in the order C''_0, C''_1, …, C''_{79}.

```
428a2f98d728ae22 7137449123ef65cd b5c0fbcfec4d3b2f e9b5dba58189dbbc
3956c25bf348b538 59f111f1b605d019 923f82a4af194f9b ab1c5ed5da6d8118
d807aa98a3030242 12835b0145706fbe 243185be4ee4b28c 550c7dc3d5ffb4e2
72be5d74f27b896f 80deb1fe3b1696b1 9bdc06a725c71235 c19bf174cf692694
e49b69c19ef14ad2 efbe4786384f25e3 0fc19dc68b8cd5b5 240ca1cc77ac9c65
2de92c6f592b0275 4a7484aa6ea6e483 5cb0a9dcbd41fbd4 76f988da831153b5
983e5152ee66dfab a831c66d2db43210 b00327c898fb213f bf597fc7beef0ee4
c6e00bf33da88fc2 d5a79147930aa725 06ca6351e003826f 142929670a0e6e70
27b70a8546d22ffc 2e1b21385c26c926 4d2c6dfc5ac42aed 53380d139d95b3df
650a73548baf63de 766a0abb3c77b2a8 81c2c92e47edaee6 92722c851482353b
a2bfe8a14cf10364 a81a664bbc423001 c24b8b70d0f89791 c76c51a30654be30
d192e819d6ef5218 d69906245565a910 f40e35855771202a 106aa07032bbd1b8
19a4c116b8d2d0c8 1e376c085141ab53 2748774cdf8eeb99 34b0bcb5e19b48a8
391c0cb3c5c95a63 4ed8aa4ae3418acb 5b9cca4f7763e373 682e6ff3d6b2b8a3
748f82ee5defb2fc 78a5636f43172f60 84c87814a1f0ab72 8cc702081a6439ec
90befffa23631e28 a4506cebde82bde9 bef9a3f7b2c67915 c67178f2e372532b
ca273eceea26619c d186b8c721c0c207 eada7dd6cde0eb1e f57d4f7fee6ed178
06f067aa72176fba 0a637dc5a2c898a6 113f9804bef90dae 1b710b35131c471b
28db77f523047d84 32caab7b40c72493 3c9ebe0a15c9bebc 431d67c49c100d4c
4cc5d4becb3e42b6 597f299cfc657e2a 5fcb6fab3ad6faec 6c44198c4a475817
```

NOTE These values are the first sixty-four bits of the fractional parts of the cube roots of the first eighty primes. They are the constant sequence used in SHA-512.

6.3.6 Dedicated Hash-Function 6 (SHA-384)

The 128-bit constant strings T_i for Dedicated Hash-Function 6 are computed as follows (where $\overline{h}$ is the simplified version of Dedicated Hash-Function 6 defined in Clause 6.1.1):

$$\begin{aligned} T_0 &= 128 \sim \overline{h}\,(S_0 \| R), \\ T_1 &= 128 \sim \overline{h}\,(S_1 \| R), \\ T_2 &= 128 \sim \overline{h}\,(S_2 \| R). \end{aligned}$$

A sequence of constant words C_0, C_1, …, C_{79} is used in the round-function of Dedicated Hash-Function 6. It is defined as follows:

$$C_i = K_1[i \bmod 4] \Psi C''_i \quad (0 \le i \le 79),$$

where the sequence C''_0, C''_1, …, C''_{79} is the same as that for Dedicated Hash-Function 5 of Clause 6.3.5.

6.3.7 Dedicated Hash-Function 8 (SHA-224)

The 128-bit constant strings T_i for Dedicated Hash-Function 8 are computed as follows (where $\overline{h}$ is the simplified version of Dedicated Hash-Function 8 defined in Clause 6.1.1):

$$\begin{aligned} T_0 &= 128 \sim \overline{h}\,(S_0 \| R), \\ T_1 &= 128 \sim \overline{h}\,(S_1 \| R), \\ T_2 &= 128 \sim \overline{h}\,(S_2 \| R). \end{aligned}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

A sequence of constant words C_0, C_1, …, C_{63} is used in the round-function of Dedicated Hash-Function 6. It is defined as follows:

$$C_i = K_1[i \bmod 8] \, \Psi \, C''_i \quad (0 \le i \le 63),$$

where the sequence C''_0, C''_1, …, C''_{63} is the same as that for Dedicated Hash-Function 4 of Clause 6.3.4.

7 MAC Algorithm 2

NOTE 1 This clause contains a description of HMAC [7].

MAC Algorithm 2 requires two applications of a hash-function to compute a MAC value.

The hash-function shall be selected from ISO/IEC 10118-3, with the requirement that L_1 is a positive integer multiple of 8 and that $L_2 \le L_1$.

NOTE 2 Dedicated Hash-Functions 1 – 7 in ISO/IEC 10118-3:2004 and Dedicated Hash-Function 8 in ISO/IEC 10118-3/Amd1:2006 satisfy these conditions.

The key size k in bits shall be at least L_2, where L_2 is the size of the hash-code in bits, and at most L_1 bits, where L_1 is the size of the data input of the round-function in bits, i.e., $L_2 \le k \le L_1$.

7.1 Description of MAC Algorithm 2

MAC algorithm 2 requires the following four steps: key expansion, hashing operation, output transformation, and truncation.

7.1.1 Step 1 (key expansion)

Append (L_1 - k) zero bits to the right of the key K; the resulting string of length L_1 is denoted by $\overline{K}$.

The key $\overline{K}$ is expanded to create two subkeys $\overline{K}_1$ and $\overline{K}_2$:

- Define the string *IPAD* as the concatenation of $L_1/8$ copies of the hexadecimal value '36' (or the binary value '00110110'). Then compute the value $\overline{K}_1$ as the exclusive-or of $\overline{K}$ and the string *IPAD*, i.e.

  $$\overline{K}_1 := \overline{K} \oplus IPAD.$$

- Define the string *OPAD* as the concatenation of $L_1/8$ copies of the hexadecimal value '5C' (or the binary value '01011100'). Then compute the value $\overline{K}_2$ as the exclusive-or of $\overline{K}$ and the string *OPAD*, i.e.

  $$\overline{K}_2 := \overline{K} \oplus OPAD.$$

7.1.2 Step 2 (hashing operation)

The string input to the hash-function is equal to the concatenation of $\overline{K}_1$ and D, i.e.

$$H' := h(\overline{K}_1 \,||\, D).$$

7.1.3 Step 3 (output transformation)

The string input to the hash-function is equal to the concatenation of $\overline{K}_2$ and H', i.e.

$$H'' := h(\overline{K}_2 \,||\, H').$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

7.1.4 Step 4 (truncation)

The MAC of m bits is derived by taking the leftmost m bits of the string H'', i.e.

$$\text{MAC} := m \sim H''.$$

7.2 Efficiency

If the padded data string (where the padding algorithm is specific to the chosen hash-function) contains q blocks, then MAC Algorithm 2 requires $q + 3$ applications of the round-function.

This can be reduced to $q + 1$ applications of the round-function by modifying the code for the hash-function. One can pre-compute the values $IV_1 := \phi(\overline{K_1}, IV)$ and $IV_2 := \phi(\overline{K_2}, IV)$ and replace the initial value IV by IV_1 in the first application of the hash-function, and by IV_2 in the output transformation (the second application of the hash-function). This also requires a modification of the padding method: indeed, the actual input to the hash-function is now L_1 bits shorter; this means that the value L_1 has to be added to the value L_D.

For long input strings, MAC Algorithm 2 has a performance comparable to that of the hash-function used.

8 MAC Algorithm 3

NOTE This clause contains a variant of MAC Algorithm 1 that is optimized for short inputs (at most 256 bits).

MAC Algorithm 3 requires seven applications of the simplified round-function to compute a MAC value, but this can be reduced to a single application of this round-function by performing certain pre-computations.

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/Amd1:2006.

The key size k in bits shall be at most 128 bits, and the MAC length m in bits shall be at most $L_H/2$.

8.1 Description of MAC Algorithm 3

MAC algorithm 3 requires the following five steps: key expansion, modification of the constants of the round-function, padding, application of the round-function, and truncation.

8.1.1 Step 1 (key expansion)

If K is shorter than 128 bits, concatenate K to itself a sufficient number of times and select the leftmost 128 bits 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:

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

Here U_0, U_1 and U_2 are 768-bit constants that are defined in Clause 6.3, and $\overline{h}$ denotes the 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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

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 $\geq L_H$.

When using Dedicated Hash-Functions 1, 2, 3, 5 and 6, the derived key K_1 is split into four words denoted $K_1[i]$ $(0 \leq i \leq 3)$, 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 $K_1[i]$ $(0 \leq i \leq 7)$, 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 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 each dedicated hash-function in ISO/IEC 10118-3.

8.1.2 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 resulting round-function is denoted by ϕ'.

8.1.3 Step 3 (padding)

The padding bits that are added to the original data string 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.

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 $\overline{D}$ of length 256 bits.

NOTE If the input data string is empty, the padded data string $\overline{D}$ consists of 256 '0' bits.

8.1.4 Step 4 (application of the round-function)

The bit string $\widetilde{L}$ is computed as the binary representation of the length (in bits) L_D of the data string D, left-padded with as few '0' bits as necessary to obtain a 128-bit string. The right-most bit of the bit string $\widetilde{L}$ corresponds to the least significant bit of the binary representation of L_D.

The string which is input to the round-function ϕ' (with modified constants) is equal to the concatenation of K_2, $\overline{D}$, and the exclusive-or of K_2 and $\widetilde{L}$.

For Dedicated Hash-Functions 1, 2, 3, 4 and 8,

$$H' := \phi'(K_2 \,||\, \overline{D} \,||\, (K_2 \oplus \widetilde{L}), IV').$$

For Dedicated Hash-Functions 5 and 6,

$$H' = \phi'(K_2 \,||\, \overline{D} \,||\, (K_2 \oplus \widetilde{L}) \,||\, K_2 \,||\, \overline{D} \,||\, (K_2 \oplus \widetilde{L}), IV').$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

8.1.5 Step 5 (truncation)

The MAC of *m* bits is derived by taking the leftmost *m* bits of the string *H*′, i.e.

MAC := $m \sim H'$.

8.2 Efficiency

MAC Algorithm 3 requires seven applications of the round-function.

This can be reduced to a single application of the round-function by pre-computing the values K_0, K_1, and K_2.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

Annex A
(normative)

ASN.1 Module

This annex specifies the ASN.1 module related to the MAC mechanisms specified in ISO/IEC 9797-2.

```
MechanismsUsingADedicatedHashFunction {
   iso(1) standard(0) message-authentication-codes(9797) part(2)
      asn1-module(0) mechanisms-using-a-dedicated-hash-function(0) version2(2)}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
    OID, ALGORITHM, HashFunctions, HashFunctionAlgs
        FROM DedicatedHashFunctions {iso(1) standard(0)
                hash-functions(10118) part(3)
                asn1-module(1) dedicated-hash-functions(0)};

-- OID assignments
-- ===============

is9797-2 OID ::= {iso standard message-authentication-codes(9797) part(2)}

id-mac-1 OID ::= {is9797-2 macAlgorithm-1(1)}
id-mac-2 OID ::= {is9797-2 macAlgorithm-2(2)}
id-mac-3 OID ::= {is9797-2 macAlgorithm-3(3)}

-- MAC algorithm identifier type and the set of recognized MAC algorithms
-- ======================================================================
AlgorithmIdentifier {ALGORITHM:IOSet} ::= SEQUENCE {
    algorithm ALGORITHM.&id({IOSet}),
    parameters ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
}

MessageAuthenticationCode ::= AlgorithmIdentifier{{MacAlgorithms}}

MacAlgorithms ALGORITHM ::= {
    {OID id-mac-1 PARMS MacParameters} |
    {OID id-mac-2 PARMS MacParameters} |
    {OID id-mac-3 PARMS MacParameters} ,
    ... -- additional algorithms expected --
}

-- MAC parameter type definitions
-- ==============================

-- The optional parameters may be agreed upon by other means

MacParameters ::= SEQUENCE {
    dhfAlgo HashFunctions OPTIONAL,
    m INTEGER (1..MAX)
}

END -- MechanismsUsingADedicatedHashFunction --
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

Annex B
(informative)

Examples

B.1 General

This annex gives examples for the computation of MAC Algorithms 1, 2 and 3. Examples of MAC Algorithms 1 and 3 use Dedicated Hash-Functions 1 – 6 and 8. Examples of MAC Algorithm 2 use Dedicated Hash-Functions 1 – 8. Dedicated Hash-Functions 1 – 7 are specified in ISO/IEC 10118-3:2004 and Dedicated Hash-Function 8 is specified in ISO/IEC 10118-3/Amd1:2006. Nine examples of hash-code calculation are given for MAC Algorithms 1 and 2. The input strings numbered 1 to 9 are contained in Table B.1. Only the first 5 examples in the table are given for MAC Algorithm 3.

Throughout this annex we refer to ASCII coding of data strings; this is equivalent to coding using ISO/IEC 646.

Table B.1 – Input strings for the test values

no.	input string
1	"" (empty string)
2	"a"
3	"abc"
4	"message digest"
5	"abcdefghijklmnopqrstuvwxyz"
6	"abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq"
7	"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
8	"1234567890" repeated 8 times to give 80 characters
9	"a" repeated 1,000,000 times to give 1 million characters

The two key values used are the following 128-bit string:

key 1 = `00112233445566778899AABBCCDDEEFF` and
key 2 = `0123456789ABCDEFFEDCBA9876543210`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.2 MAC Algorithm 1

For the examples in this section, the value $m = L_2/2$ has been selected, that is, m = 80 for Dedicated Hash-Function 1 and 3 and m = 64 for Dedicated Hash-Function 2.

B.2.1 Dedicated Hash-Function 1 (RIPEMD-160)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	B7F4508111EB8C3B5229C6AED406DE9ECA640133
2	BC78F55933BCEB1EE85A906F9E18374F23E310F9
3	6300DC20E97A5AA29DB9C7D607D23D126FA36863
4	3A2AC89B78EEAB8759F5112BCAD4CD405EEB5D35
5	16DC174925BBC27E0C93D426C346846F97F8BC69
6	E062210BA5C9C94737BF3A6E85B3B5664FBD1D4E
7	9B462D5CBDAE1485FFE10BC001EF9E3AF6D128B5
8	88E73A01A1DE36C92D6F9E41F7278D407B4A4CCD
9	E7B128E4A1842B750F1E61A486C867C4887A4B21

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	B45D6CA84CFB9020E0D5ABA2A7609D3D81F3F57F
2	8844375992037D1BCD0D118EE548D70C3F19CBBB
3	917C59B8AC7FC19DC25BEF82766412FA16BBC6A7
4	E0737CC7976D8F424390CB8798D623D751AFE15A
5	D57FAE83687071 8EFA4BD4A5F2F322A179A8735E
6	42B20D4C8FD5E8672760CF83C0478D7BF8021404
7	42B20D4C8FD5E8672760CF83C0478D7BF8021404
8	10441DF4F68CE8815818DC0FB370ABF87BCA4464
9	E06AD21D2AF04DD4217AB03B1A578F036997D01A

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.2.2 Dedicated Hash-Function 2 (RIPEMD-128)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	64-bit MAC result: leftmost 64 bits of
1	A47A64E9EDE0741B3FDDE33E5C1C6D78
2	51355051852FDC79FB228EAC905633AD
3	D83940DAFFBD4CBBE6BA30A6F9E63F5F
4	1A7CFE2BB26E973E213C1CB96FA4C2EF
5	798AEAC6046B31907C197BD68E59D376
6	0B8E1D4A571F32657189E22A1F2F4A53
7	B814730F482300C6E474FD255A66D680
8	9060A30758EBE3368D939AC168F1A9FD
9	20763FDEDF01E56FF5756954302C7DE0

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	64-bit MAC result: leftmost 64 bits of
1	35FA3AC39F50F2A4E3FFC7AF5776B4EB
2	A89E25E6796747B630A2A00B802EA53E
3	66339027A36608EBD932DD551616E7B2
4	1F8779BAD84B503739311211A2761EAD3
5	31BF5B5B7ABAC2567DC0E02F1C3A25D7
6	B5B8BA3B8EA895FBC83CB7588FBD2656
7	8D27BBEC257C848D5CF375EB5EDA4CC7
8	B40B5BF6727DE90B26F770850F059C89
9	76C7BC831B0BCE593DFD44E8E054A373

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.2.3 Dedicated Hash-Function 3 (SHA-1)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	C8A8B3C75E6CE7C6C4F79CC19853CCD54ABCB079
2	8DD9AE643BF10BBB7B978EF13EE6C0F480618FB0
3	A738B26A8BD318184E76707A99CAE14C670B9711
4	1EBFE413E55D6B288A2BD01D294A21FD8D4B20BF
5	0CE7BF40A73D977AB4999CF3A9BD1C5BEDC442E9
6	12A6823CC181294F95109073A6AA0C8961B14386
7	9369EE4A043AF1CA6E078D0B8A9CE5C1545440BA
8	B00D37D70A84B762FC0A8A9BC1B15F0E517B5EDF
9	DDDF44613E8559D12C150D022D5FE33F9E0FBACE

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	C3A5ECD1E715C7272CFE78BC278086587B040422
2	D5D50FFA7EFDF1B17E96E2EC14DBC4412F7B771F
3	01BFDD568008D412158F5B0C90AE2730DCFB77FB
4	9982E0EE91DB89AE7E7618AD1D649BA43406DBDD
5	ACD04E1004FCE53DECA9EE7AB95DAF97B7C44AA8
6	FADF62DCE789E86E60756AA819EF62C3E5C25E94
7	46DB9A49FB4976D007B14B1574843D019CA99445
8	4EF5BED3E816C530B23F491583C038596BB76FDB
9	BAC6BE6BE6153FECE2891F9DA03824DD4D535D19

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

B.2.4 Dedicated Hash-Function 4 (SHA-256)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	76D91CA2337FF25EF66DF2AE7172626C5544428822B9E1B9C94121D384489C09
2	479DA7965233CB2133C6B949AD82CBC5B29F528DB90FF04A1496323D77D15FFD
3	BE6E923798F594BC529C87DF5A42333EE18BE88FED984B0EFE092BF31D570FAE
4	664AA91CFF68C786E943FEB0E6BB465213FFC57AF5C8F973827DF67956FC21D6
5	D7A7A1D1007CB2D3DC578BB4FDA4A5B1B2EBF7B27C9CC43BDF7A382851DF91AB
6	BC9853B4E7AE574B3DCF4728BF44FC27A3C04C5AB90EA189DD175B6F5ECB4335
7	66F3238A2C2B6A3AB86089B9DF33BA6420F7E66F5DE6856D79CCA908DFE57BFE
8	B1A59E0905F8EE9ACF5C77E67883C8C3CA10DA965BE31F75A47AD85015CD478B
9	15FC09FABB62AADEE831B9988E2DE2F41A3C685D28E4C06720ED6E8493CD060A

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	F0400A79DD136B4E83EB507E23F98C1A54E7DCA33A38C75902008F90B003C37A
2	407FA9A8170113C1C0B06B8FDC32AB5914343D0CBEBEE5B1C84008182794CD8C
3	7E02C2AB8B8115B446E80C70CE4A0126E83E0208420E39965272D6497EE16B86
4	BEFC08E950BF7CD6B6BA0026A910328ACF45551DB93180099D0893C8415314F1
5	00019D3C8D5D563A9A24462029171C6D4CF29BCF8CC6D60BF76584A6B4F696ED
6	383F7B8050D7B08DEAEEB3B5B0449666981527796 8D5A2ACDEF04D37C596E2E7
7	FFEE6E137909EF2140A87619B51CF6C7FBAFC6EF5B8388C8ACF0F7DE5AA8E7BB
8	511855ADB1B5FE79C1C04B565CD40359B5DFA474AC52BE7F4CB2753285B90D0C
9	8F6D5B1C7CC360DC4E4320755684B24726B8C4312A12B329ADC8C2550C3FEB08

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.2.5 Dedicated Hash-Function 5 (SHA-512)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	A4D628912D3CDA87D92F4597B7385E9BEB6161A2C12D412E3EAA7B4FD1003ABE D6EBE9C7418D60905267A84F0A2B22865D8E21F1E48D4E105F6C3A653D5C63E2
2	DBBD9316AAE10399C742C212365A529B7EE5F9BBDB96BA9A14A078010AF81806 AF4635AADA77595FE21B7B5C552038AAE38CF32C4D4E480855560E98AC25297F
3	C9598B319F5DA537044289553FB7B0FAC95B51569BD08DB4A45995CB75A344FE 88ABF5001694497ED71B5CAB3C5D4212E937E50712CCFAFF5C8B3D4C23EABDF5
4	B22A1CD30D3AF351931E746542BEFAD2985BD6838831BDBCFEE3B9DA48AA8996 76C4ADF4278D79D45A6C6E4AEA613B5F3FDE4E6F4FE06854B9736B9355EE0A6B
5	EC807431CE07B43F95E001B562525B0F49EC6BD5B91055C030D79D5E462008A8 B9D862ABD3E8517D59FE3F3E60424EFC1327D67D53A04F4871076999619D4327
6	176422D10271BD8A22AFA4164F62439DEFAF0B4901AF4F8FD366C79055280635 175B8D920574AC85B493FBA1EFFDD46233C54BAEC783FF3030BADF6FB37413AD
7	22E8A8624F3982CA3A7B18635E4029FB6CD3B771EAFF7BAD5A00C064C1099B99 7BCF7FA529D5864BEA94DA7EB5367D8C27763B7303FAD4F517D598AC7453A60A
8	17A2D95A4935C88234EEBFBA29140B57ACECA329E513AA7BD7110283759FA6E6 D9457D4B58C7DE765A495703B04AA476E5412DDE52E799C841EAF37925A37CB3
9	F807A843A14B94B977259556B656A7A401C3D026B5FDC993BC1E6A2E0E5AC7EC 7CAA611C9EB3F4609E7699F43A305BDC4818B823B219BE7AB5C54A4AC01F5E4B

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	14D4BDDF705DEFA756AE8154FA09792D1D86EF52A88E8552CA4DB56420F6BD08 0D3114F626B90B59EA1D0C7E4187678E22C263FE94D074D70477252CDDA86C04
2	C340C2B6F8C606AD499747EC6239DF94D5365047061EB4A26789EB83177B002A 2F8C1F5B866A23389CAE742A057C4ECBDC7F7C20266BC99625A974DC345B0EE0
3	AE0343C78F41BB04026B8BC36C7D09BDB7E6C8450FE340A7223274C5F61DBA14 5C776D694C3B05E8BEBB1A494607A00E363E21B3BCC7ADB5FAE52F20D4F2B210
4	F98F58C4C7498F17A70FB5A86CA0D2AE86DA99E318CF0B8A801639A6DC7B3DB7 C57B975EA6347B55D68C6E8B34C185CB06972370B15EADBB4F35C8277AFB7D79
5	7B073E2F13EB203BF937567D6A42F4F80A7BBD47E120226B2B1171C8F62BCC53 CC0DA4F0DD78C5534C1370E3DAEA81DD2EA6DF7EDD7BEE7334065A6A2B0A0FB8
6	641BBAA2591C17B1D73435DE640A22FAC1A2C38D11BC025F6991ACF667096011 D6E48F27826F06BB006425DC4EBBE9EE7CDE3CF1C3A9592C674CCEEDA0F10BE4
7	B0025F9B04BDD64D15AA61D0A5CF8CE5C1FC0A55830CF81FEAB1A3854D5D3E41 F11191891 3E9638292B9BF752C6B6F0626A322FC89C28E03C80816F5115C7753
8	5E9A62193081 23921A527502526DA57957E0C2C00601CA5224769DD925FC43F4 7FEAC4163B0D62CBCC7E4537859792DF8E4CCFEA4E8E3D387014A514F42B6CA3
9	CDB9961A62447DAAEA3046E59517DFBAA8C7E51C7B45A4561918B8B5CFD3EF89 96B36921490049A70AA19CFAF88F017BDC03FC1CE419BAB718642186C17502FC

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

B.2.6 Dedicated Hash-Function 6 (SHA-384)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	3148BD96B4CAB64EE95C6444EAE0053E783AC28949FE05D0 52A1FCCE7DB8A619A1D0BC858FAF013E53BD2A24AAEA03E7
2	735E4ED128DDF2EEC1FDF0370BF32517DF63E836FC0CBBB9 A1C5CEFEC32CFB5586E2AA3A85ABD08FC6EB8EB9E777CB52
3	15B7AF6E28E6AC436DB405B706078ED88F0F9D292C4C1A4A 5EF1FAF0BA315683439D0DFE325283C1C83DE846C23DA890
4	F49C22A8E48D0B4C145EEFDDE51027248A2BCD341FE43504 4DC980C38333D5F197CE08BF26354C545690C805D6AB1E9F
5	9DB498C5971A668857AD56C724F82104C2CC78D90701A29A E97D9D269180FC64E35E058CE8898927001D20BBFF3B472A
6	3D782A279E1D683623EE34D9935D43D9627CF5D045DBEB7D FFF8C29CB3916A4F0FF00C012CE125956655873A3D883812
7	3A9AFFFEBF68E5660606797A5BAFDBC2B47DE1DEDA40AC48 0F535583D9544A63210ACE4DA24F997786C2F80367FA5284
8	0309E1B60798568E00522AC145ABB39AA19C5066549E07AA 5D6ADDEED34B4F2F3DF166C9C3915F44F92D79A049E7B753
9	343CC7791B07C4D26E5735224B302C495D9CB2A92EA27BDB 2A96487C9CB3EC09D75C9E0179F73180E24D78000D45D8FD

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	FE10F454AD0EAA717123E77AF83760DA2B5F27AAB2BBE4D3 5C4AE1352D54CA36554EA206475F1BC5D24060CAF17E7432
2	BCE4739807A7BD64F709605DBE1A2B572D963CFF5CAF557F 194F18CB924A6B90B2A54E3844701A77D459A64AB3142A1C
3	86CD24861B303510FA374D2AA5E2CA78F329138D5B9C2E04 3C0FFC5E0AB37342B7859AE8B7CD8D6F3B1833F81628CA9E
4	B03FC36C1FA2C72EF1EC1C9B8E9E322EA23C1A0F6C5AC5E9 AB2BA60D560D7156385D8D2DB0B117C71340E053DE4A08EA
5	3B923573C9598E5F78EED06EADFCD6DAE2F8C762A8BBDFDE 8F0986934F5DD4A212B3493DD190E723A17B51D4F4F351AA
6	0188E2DDA587AD67E91DD38C247051E1446CEDA39EDF33B0 DD9BFC4310BAFE41887638216955F6E31E683420D858D16B
7	269010C8844FECCF00F3EC33E600DD6F1BA56DA91F1F1257 575BD5C5EBE6931F212CC8FDF81EA70C0CDCFBF20D06E84C
8	92CF0571CD8AD78C1CD43E847837A849EC8B92CEAAEED5E7 541E23B14C4FF713A40C94D6A34E731543E1D3EC52B69BEA
9	95FA93E72585717CA74701D46EF9E25A9E2FD10FE015CD8F 04427323315A60B074E1A1134F18B2154644658C24EB3D36

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.2.7 Dedicated Hash-Function 8 (SHA-224)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	065B86EDEE58E12C4D83C7AD1500EB4CD313DC3AA2147A12F39B7AFE
2	73D5627C9DBDE736841766FE543B7954D59AEE1F5C6823BCE7E77351
3	A4F4EA69DF69D9705D71305817B38AFE1EF6ECF724C3F6743B26A9D2
4	E8E14D49D6C2A88D4A717A276693FF3D03444AC43FE02C99B6919A23
5	38CB47CB00B57B858A0ABB864F05B8E83EE4A250A5794740796FAA05
6	64ACE1CD852195AC765764DBA813749BDECB15498C30FB6CA73E0F71
7	570E89F76E8435B945CF47AF5B054262C654636AEB955FD951076B91
8	CCC8BDEED3869F5D8E2013EDFCE22A36185C5403103F04E586F987C5
9	E0BEA67230DA03039540FA70CB0FBD69464E9DEE3C3FF80CD5D76646

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	FD5B8A7CD44B62730BF4DE82D7E2D70135E29FAEB1E66933D54BFF68
2	E70AB63A4D92C3B2A6975BB24B9380B5320D7E510107D27CC97F0086
3	36C19C58F5F13B43544424A085EB5E822AF4BD7F32B7AD190B5CA3BB
4	DC53898B38F96269AAF0890A3A4D7E00B03B931C3AF7C80C8BBCF6EC
5	446B9988D8C5B35A20DD6B71B5F1E3E048144CD082C801449B5497EA
6	48E72C0EDC3E52370EF2DAE859A5462D60C735DA3F0B7494AFB0D5AB
7	3E048D2F615BC681D1B2FD59A37F7D8972AE7C8096603B859F771223
8	E2CC9DFD0A2945F8F2C3003ADD8AAA493BB0C72BFAA82B7CA8B1F289
9	552A67693AB02EC7D0AF18075DA9875B8B5D1DB89F6CFC73EA7151DE

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

B.3 MAC Algorithm 2

For the examples in this section, the value *m* = 80 has been selected.

B.3.1 Dedicated Hash-Function 1 (RIPEMD-160)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	9EBEA41FBC24CD80BF2ECFD5B8C8CC8181D3FCAE
2	75CB722C50024C0E8A7A0DBA7D5C36B86D9D1DD5
3	5B48C1749DDED71EDFE0ADE2B944E808E4A65820
4	F9033064567F541235C3944EE95CB476055985D1
5	B37885405B71E025AF0CB574021A562A62733628
6	5C6429B982C8054B5B3348A0D7D2CE24D7032BC1
7	B0A4A451D0926855E52428E16D1FEAA241C4DD9B
8	1CCEEC5122F08A76EBCD8E3DE88610D942D8A5F6
9	45D61908BFF6039E6DE3C037FDCE6191F19F6410

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	2FDE5DAF7050D14E6D7ACD2254D17FA3A8CBFCDD
2	239C4020610429A8662BF81A2CAAEA47F8EA0A44
3	89EFFB9F5A6BCEAE3C65D0C9803F3464E5E9E349
4	F5FC87FD5702F5D4E7BB634DA4CB4B41CD505B6C
5	5686C00F69E6C868732C67402AA107CEAB513439
6	525EC4893A221EFD9B6DD351059B40C05B4CE2D3
7	B975ED3893FC8D535376EF49211E2E6B1BB30B90
8	BC201FFA581357C271DAE25104167F3DCC97BADC
9	95A875A1D64D55E677D8E4455E1445E7E940F758

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.3.2 Dedicated Hash-Function 2 (RIPEMD-128)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	64-bit MAC result: leftmost 64 bits of
1	AD9DB2C1E22AF9AB5CA9DBE5A86F67DC
2	3BF448C762DE00BCFA0310B11C0BDE4C
3	F34EC0945F02B70B8603F89E1CE4C78C
4	E8503A8AEC2289D82AA0D8D445A06BDD
5	EE880B735CE3126065DE1699CC136199
6	794DAF2E3BDEEA2538638A5CED154434
7	3A06EEF165B23625247800BE23E232B6
8	9A4F0159C0952DA43A8D466D46B0AF58
9	19B1B3AF333B894DD86D09427116D0AD

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	64-bit MAC result: leftmost 64 bits of
1	8931EEEE56A6B257FD1AB5418183D826
2	DBBCF169EA7419D5BA7BD8EB3673FF2D
3	2C4CD07D3162D6A0E338004D6B6FBC9A
4	75BFB25888F4BB77C77AE83AD0817447
5	B1B5DC0FCB7258758855DD1840FCDCE4
6	670D0F7A697B18F1A8AB7D2A2A00DBC1
7	54E315FDB34A61C0475392E5C7852998
8	AD04354D8AA2A623E72E3594EE3535C0
9	6F9B1C0FC06753618D6DB4B007733795

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

B.3.3 Dedicated Hash-Function 3 (SHA-1)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	86C2962E58B3498A2608935AF7726311F2BFB538
2	0497FF21DAE3251DA0ED2F47F5A3B74ABA6B2560
3	6EE2A25F943E3F3EC05225FBB86BA73E2E5D51D2
4	CD4C0D1328DC4A8DC2801001B129AEFC6E0CF9CE
5	89ECE303FAD1E4313950CC3B008CB239B5B85844
6	9DF741057D075D3C4E1533E38A5FF469647194B4
7	188A58390A6EF9827035B81CDF1B5069211F0EE5
8	98A98D6A81FD361030856D2C19742AD8DBC468E7
9	D2986310BA18A78786534882F9C6BCBF06CCE9E3

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	2739B6BE63F539EB70FE250346F6382A2DFA345F
2	A0C2711A6B1DA4CD8F85EF1E6FF7BF70B412B477
3	18F570E864FF903D2773D53C2E114E1A62152953
4	A80845A89BA15E941A2457084BC431F3E47759E1
5	14143EA1057B02D20C0157216190A006E30F3D41
6	DAB4B41BA639B4715889406FE18E0C037017E063
7	AEAEA5415B4F266CB15CBEB844E56AEC2DABAD6D
8	3DBA11471EB4FCCF21BAEB0BFF7E20150132C6CF
9	3BB917B8BD8560E89FF9054FBE096CBACA109D5F

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.3.4 Dedicated Hash-Function 4 (SHA-256)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	E8A06537F096CCF1A3C425A56CEA054072C4A8DB67BD28CFB02FBEAF84B35F6C
2	DDABFDF46CE93311868B7275E05730AD3E23192A575CC291AE3785289B94A2F3
3	02581EA39A6CF2D752793FD782CFB9CF965BE72B32B322C9551D03510645FB31
4	1F12288F42F42661349E5DB741CE19F3B8C3A8149FD4B8981237FA200FEB104F
5	EA4A04E76EEC57D6906098AFA7AE0264072C09F0DB34269B117C68C3ED989C5E
6	6EB683218305A862A1C1EFBA04A2A62DC4EC27886D3C79AFF7C493C2D6DFB080
7	6DC64AC5C5F197EB5463474AA6B329DA9D5B3C6A3324B147469E06F21EB53C41
8	8F4B417527DA9533408D95951ED6504525C9683B45637B246CE25C99ACC64698
9	5E2E0579A26517B06D2933CF62DEA20347A0A8DF9D7C3D200FA5E894ED9C5EDF

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	DCC3C81236AAD92043D1478DF7926E78205F7BBD0C3001854BF9087261ACCE47
2	AEE154DCF83568248DC228C8C3513E9BEEA268B4979FF17CDE5BE484F4919DDA
3	C3B53B9897D72197B240F08715E5C830886FE2F2EFC2E5A8ACD9D5405098863B
4	60CD78CAED2CC9BD3F5BDA6AAA81596B55556660B19A2DF2FF6C48F89C52CD7E
5	6283D8BA031EE52E2D7EBA96287025F161A5219EF1FB59CEBE6133007B35A146
6	3BB625768D0900710F0EF7E854990BBBA35AA9B7BD4B0133656D290992A9BF79
7	98FF69D0048FF552843CB8D5DC686EB2FEC3600D664A464F7B88F7289CC41A78
8	5893F4AD6CEBB85BB90CD4107BF85EEEBAB621C6EEB4EC487780A45DED09F5B2
9	781BFEC8396C6268E5413D76EDAE0C90E6592B624BB4E0FB6137F4DF33FB91D1

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved

B.3.5 Dedicated Hash-Function 5 (SHA-512)

key 1: `00112233445566778899AABBCCDDEEFF`	
no.	80-bit MAC result: leftmost 80 bits of
1	EEB2E7DC1EA7C75966552A7F45C9F30E3FB9A7EA1362BA6D7324DD7DEF461F88 B2B5E433CD7CA25A08554605B0B020C3A865434BABFC140DE5D55C8A94C7FDC6
2	8A507C281F9155A086C97E3BDB14B658F6C901B19486741393898 53F55B453C4 83DF9AB807269B286AFD6FBDB6A59E3CF190BB61951D8A89BB0F3D611DC012DC
3	F5B41F81B56D9CEF4BFFFBDD659470CA9DE7348A23DAC136790B028986D13E7D 74DC59759FAEA253D5342ABD56CF6F5859145AD54BCA62E0C45245B7E4FA5C53
4	2EEC2E512FFDAC27444A563B40BFE9EAF594D0C83947A374AC82797DA811466E 8DF8380836446B4B392E4E815B8E84695DB8253230635C18CFEB19CBE1CF012B
5	DCD40B28C684428F83D1E7D457A906DFCAB55C2298B7A242C4F208F205E948A1 BA081A39C6DF60E2279DE4C3D6F82A4490ACCD6679AAF7FEA90DE4BAE06F2C32
6	E78F92E31D7410DD16EC830B477FE703B79925811758F0D3A11F3E4F48DA0C26 87A797EB2E3D7E20026936A87E6903F9D8C93EF3E8FECF2CB2A42A720F301821
7	49BCFE57FA600FDF68562B91E686B02DE81DE25CF4466C707298538980880BFD 339264B48F2BD712127A1C66D97D1B367DDD8656B996CBD8D5B7EDD561328CD5
8	C4979A98F32B6DCAC7718B6089694DBFC6E6ACDF82724F1EFFB277593B716389 ECEA6F4C40EA524C84A1324C6AFABB78C0FE0AA008C1EA3427AF179540FDFBF3
9	376DD55BA616E59FCD62492675776085 63C168CBF82CC6A89B83BC9224641B28 CC944C6DFEEDE8CCF7FC6F61FF0D322B3183449677330B6EBD83BDD7B57FB846

key 2: `0123456789ABCDEFFEDCBA9876543210`	
no.	80-bit MAC result: leftmost 80 bits of
1	42A0B3DC6AF1D40CF4D58E2E35832C5824AA77E6B685B3E3BBBE69A82C726DD8 B46C861BBE0DADFA3592074261 87A1675A054CE82905F5A2FCE5495D6BBB6957
2	DAED898D6A86AFD1C622C96D33521175690365330EA0CEEEDB85292F8BBADFD8 67A7C1327356DFD75FA0C38B099BD229C39CF7CF07F479308F52A5C29CD284DA
3	41A98A3AD2B5BF46C000DEB754D93C2D41C4EFBE272163346E8D78A1FA222BD8 046551C4FFE81E8BA0A0DBD746A25066DBFAE79B4040D964D8F2DE181E71212D
4	1296CD85141D1D3BAA6C52ED6A1569925112AAB7820883B3369D278A5711C62C F483045B5F417284 1BC917BF92D45EAFC448D975FCAF58D95E8E9AFD127BB7A6
5	C0941AAEA51B8539592403C1CCBB98736F470F5F07C2FBB2374CEFDDE6BF0A34 EAA164B430B59E6921422D6E0C5BEA969FB9F6A7381AC9A8E0107D5BBDD11E3A
6	B87B9328C0041DC4492C5F0AA613EBDC9E1D01156E4E0628705A27B3DCB6EEEC 20E862DA7971DC2B999B6C6952ADB7D8615E68384289076C4B752D0DF393F2E3
7	CB102567BDB4AAB8833419ED3BB95F1875488439B5416FDD466B79CC73BD26E0 9690A3E94CF611D7532128224E225C671B50FC2BED4934516A4955931774B30E
8	41B7CD308733F10CCCC0AE5CAD8AF9E0740317A0EE874489872EC640CC0CB16F 101A12E446F55585555E7AB5128B8D370C006FFA151C7FA35EE10144E1FDD16B
9	2FFE4FEF445D76A2A41E5EDB715170D02ACAC44A580144C17ED65434A876CC99 568E39E97CE78E5325EE376B113C6D7C5247D96AA0DC1D91A0932C81B58D956F

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

B.3.6 Dedicated Hash-Function 6 (SHA-384)

key 1: 00112233445566778899AABBCCDDEEFF	
no.	80-bit MAC result: leftmost 80 bits of
1	7BEE4557700A1D8ECE045E8A6FC980F456D2FF4F5AA77BEA 3147F54DC77E8F6A2FA016E9BAA4E105D53B4631CE088CBE
2	33B764BFC7E4A95BB432BD7766C3EB0F0AB686CAC8FD40CD A53A0B0D623D48373F36B321412C4D01E4AAF8BA53C77751
3	67BF47CD4B410564245D335985B5DD404D085E2DB88F2A35 B0782C7FA4AEF3407D489D66EA8914E74752CD191396313 9
4	1522B5D65022C1CEBF2425EC914320CCDBE198251CFEA79F 499179A2025185B5CE6241E84A6FF0F9C83820FA83597E62
5	98E3EA52031575D96489C7DB7AE3B4A4AE7D80E789958CE7 99350E2E07D7B852FC8BB3EFF3F2954A2C158BC3C70E8ED1
6	34B80B9F2775DABB019819277302ADAEEC0CA6F0963B2979 314A44724B6318D9213DDFFA2E04B175E1E7D6778FC8A8A5
7	5BF44F677E77FBEBE31516D80CED014DC99E7F51AC4CC41F 6401292990E3668319B137C2F1626C67BB92A1CED7BE15AA
8	7A8610621AB18CA0C87AD25A984C333D3B4BE12E85DEB8E3 88E4656133115FD4710DF4B81D0A526E56553C25E6279131
9	63490CBECD7350ACD6D9D5F485D323440A271555CC3C1E51 F245E0FE4D8DFE6340C6146D2EEB46DFF90A0D1970A30C52

key 2: 0123456789ABCDEFFEDCBA9876543210	
no.	80-bit MAC result: leftmost 80 bits of
1	6760086DFFB66B3AA619569BF567D6ACEFD52E2F3D90CC40 103346CB1EE0A493E159785876F8C310B44BF05B64E6B2FF
2	6575DC9AF2E0A9D32D619465C95FED1FF5B1D3E65A2EAD93 405BD2DA82896EE3F9D336B374E5D015594EC44872EBF8C2
3	A6685B72C7545F84405CF20CF17CB67B246FA914C59F335E 7F95B5B11963072777FA3B635AABF86D0D75B83D6365211D
4	0D52B84209956EFD9F39DE27821D328CB0B3FCDEFCF64B99 ED2B65C4DE7A753BCA2361CBB26043649FDFCD8C757E700C
5	9A50FF272D08AB3AD03911B4FB042E0CB8080C18F5938F0C 93340DA508722DBB799C72EA1274B67AD30EAE22E86213B3
6	21BC7FA9F9E23536084CB65EE28727DDD378A1FD6D316D29 BFC3C8A39851EDE817A392D460A628E79A018989249A0CC0
7	F291C145D2F9A10C76C5E40CD4C4027E2688799C95FE4C45 3042DED3EEE4C4331CDC5F571B8642C615DA2A1A854C6EA4
8	23A9161DE7B21284446B49D5038F0D2823A0F05619B243F3 D0E114E3AA9AC905C506A9546E9EEB41F1DD1ABCE8F43B71
9	D056C9491A84401387A18E6953C7157E86C3AD4D3E2B0971 5E91B486EE89C7FE17CE40A10C78EF819433F006F7779443

IECNORM.COM : Click to view the full PDF of ISO/IEC 9797-2:2011

© ISO/IEC 2011 – All rights reserved