



BSI Standards Publication

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

Part 3: Mechanisms using a universal hash-function

National foreword

This British Standard is the UK implementation of ISO/IEC 9797-3:2011+A1:2020. It supersedes BS ISO/IEC 9797-3:2011, which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the ISO/IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by A1 A1.

The UK participation in its preparation was entrusted to Technical Committee IST/33/2, Cryptography and Security Mechanisms.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2020
Published by BSI Standards Limited 2020

ISBN 978 0 539 02649 8

ICS 35.030

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2012.

Amendments/corrigenda issued since publication

Date	Text affected
29 February 2020	Implementation of ISO/IEC amendment A1:2020

First edition
2011-11-15

AMENDMENT 1
2020-02

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

Part 3: Mechanisms using a universal hash-function

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

Partie 3: Mécanismes utilisant une fonction de hachage universelle



Reference number
ISO/IEC 9797-3:2011(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2011



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols and abbreviated terms	2
5 General model	4
6 Mechanisms	5
6.1 Introduction	5
6.2 UMAC	5
6.2.1 Description of UMAC	5
6.2.2 Requirements	5
6.2.3 Notation and auxiliary functions	5
6.2.4 Key preprocessing	8
6.2.5 Message preprocessing	9
6.2.6 Message hashing	9
6.2.7 Layered hash-functions	9
6.2.8 Finalization	11
6.3 Badger	11
6.3.1 Description of Badger	11
6.3.2 Requirements	12
6.3.3 Notation and auxiliary functions	12
6.3.4 Key preprocessing	12
6.3.5 Message preprocessing	13
6.3.6 Message hashing	13
6.3.7 Finalization	14
6.4 Poly1305-AES	15
6.4.1 Description of Poly1305-AES	15
6.4.2 Requirements	15
6.4.3 Key preprocessing	15
6.4.4 Message preprocessing	15
6.4.5 Message hashing	16
6.4.6 Finalization	16
6.5 GMAC	16
6.5.1 Description of GMAC	16
6.5.2 Requirements	16
6.5.3 Notation and auxiliary functions	17
6.5.4 Key preprocessing	17
6.5.5 Message preprocessing	18
6.5.6 Message hashing	18
6.5.7 Finalization	18
Annex A (normative) Object Identifiers	19
Annex B (informative)  Numerical Examples 	21
Annex C (informative) Security Information	24
Bibliography	25

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 9797-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

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*

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 four MAC algorithms using universal hash-functions: UMAC, Badger, Poly1305-AES and GMAC.

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 mechanism and message authentication mechanism is dependent on the length (in bits) and secrecy of the key, on the length (in bits) of a hash-code produced by the hash-function, on the strength of the hash-function, on the length (in bits) of the MAC, and on the specific mechanism.

NOTE A general framework for the provision of integrity services is specified in ISO/IEC 10181-6^[7].

This page deliberately left blank

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

Part 3: Mechanisms using a universal hash-function

1 Scope

This part of ISO/IEC 9797 specifies the following MAC algorithms that use a secret key and a universal hash-function with an n -bit result to calculate an m -bit MAC based on the block ciphers specified in ISO/IEC 18033-3 and the stream ciphers specified in ISO/IEC 18033-4:

- a) UMAC;
- b) Badger;
- c) Poly1305-AES;
- d) GMAC.

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

ISO/IEC 18031, *Information technology — Security techniques — Random bit generation*

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

ISO/IEC 18033-4, *Information technology — Security techniques — Encryption algorithms — Part 4: Stream ciphers*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9797-1 and the following apply.

3.1

empty string

string of symbols of length zero

3.2

key

sequence of symbols that controls the operation of a cryptographic transformation

3.3

nonce

number used once

3.4

prime number

positive integer greater than 1 which has no integer divisors other than 1 and itself

3.5

tag

result of a MAC algorithm, adjoined to a possibly encrypted message to provide integrity protection

3.6

universal hash-function

function mapping strings of bits to fixed-length strings of bits, indexed by a parameter called the key, satisfying the property that for all distinct inputs, the probability over all keys that the outputs collide is small

Note 1 to entry: Universal hash-functions were introduced by Carter and Wegman^[4], and their application in MAC algorithms was first described by Wegman and Carter ^[10].

4 Symbols and abbreviated terms

The following notation is used in this part of ISO/IEC 9797:

$\text{bit}(S, n)$ Returns the integer 1 if the n^{th} bit of the string S is 1, otherwise returns the integer 0 (indices begin at 1).

$\text{bitlength}(S)$ Length of a string S in bits.

$\text{bitstr2uint}(S)$ The non-negative integer whose binary representation is the string S . More formally, if S is t bits long then $\text{bitstr2uint}(S) = 2^{t-1} * \text{bit}(S, 1) + 2^{t-2} * \text{bit}(S, 2) + \dots + 2^1 * \text{bit}(S, t-1) + \text{bit}(S, t)$.

NOTE Bit strings are treated big-endian, i.e. the first bit is the most significant.

blocklen Block length of the underlying block cipher in octets.

ceil Rounding-up operation, i.e. if x is a floating-point number, then $\text{ceil}(x)$ is the smallest integer n with $n \geq x$.

$\text{Enc}(K, X)$ Encryption of a plaintext block X under a key K using a block cipher Enc .

floor Rounding-down operation, i.e. if x is a floating-point number, then $\text{floor}(x)$ is the largest integer n with $n \leq x$.

H Hash value.

K Master key.

K_E Encryption key.

K_H Hash key.

keylen Block cipher key length in octets.

$\log_2$ Binary logarithm function.

M Message.

MAC Message authentication code.

max Largest value amongst those given as argument.

N Nonce.

$\text{octetlength}(S)$ Length of a string S in octets (where S is assumed to have bitlength a multiple of 8).

$\text{octetstr2uint}(S)$ The non-negative integer defined as $S[0] + 2^8 * S[1] + 2^{16} * S[2] + \dots + 2^{8n-8} * S[n-1]$, where $n = \text{octetlength}(S)$.

NOTE Octet strings are treated little-endian, i.e. the first octet is the least significant.

$\text{prime}(n)$ Largest prime number smaller than 2^n , for any positive integer n .

NOTE The prime numbers used in this part of ISO/IEC 9797 are listed in [Table 1](#).

Table 1 — Prime numbers

n	$\text{prime}(n)$	$\text{prime}(n)$ in hexadecimal format					
32	$2^{32} - 5$					0x FFFFFFFB	
36	$2^{36} - 5$				0x 0000000F	FFFFFFFB	
64	$2^{64} - 59$				0x FFFFFFFF	FFFFFFC5	
128	$2^{128} - 159$		0x FFFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFF61
130	$2^{130} - 5$	0x 00000003	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFFB

$S[i]$ The i -th octet of the string S (indices begin at 0).

NOTE The specification for UMAC in [6.2](#) uses a starting index of 1 rather than 0.

$S[i..j]$ The substring of S consisting of octets i through j .

taglen Length of the tag, in octets.

$\text{uint2bitstr}(x, n)$ The n -octet string S such that $\text{bitstr2uint}(S) = x$.

$\text{uint2octetstr}(x, n)$ The n -octet string S such that $x = \text{octetstr2uint}(S)$.

$X|_s$ Left-truncation of the block of bits X : if X has bit-length greater than or equal to s , then $X|_s$ is the s -bit block consisting of the left-most s bits of X .

$X|^s$ Right-truncation of the block of bits X : if X has bit-length greater than or equal to s , then $X|^s$ is the s -bit block consisting of the right-most s bits of X .

$X \gg 1$ Right shift of a block of bits X by one position: the leftmost bit of $Y = X \gg 1$ will always be set to zero.

$|X|$ The length of X in bits.

$\text{zeropad}(S, n)$ For positive integer n , the string S is padded with zero-bits to the nearest positive multiple of n octets. Formally, $\text{zeropad}(S, n) = S || T$, where T is the shortest string of zero-bits (possibly empty) so that $S || T$ is non-empty and n divides $\text{octetlength}(S || T)$.

$\oplus$ Bit-wise exclusive-OR operation on bit-strings. If A, B are strings of the same length then $A \oplus B$ is the string equal to the bit-wise logical exclusive-OR of A and B .

$\wedge$ Bit-wise logical AND operation on bit-strings. If A, B are strings of the same length then $A \wedge B$ is the string equal to the bit-wise logical AND of A and B .

$+_{32}$ Addition of two 32-bit strings, resulting in a 32-bit string. More formally, $S +_{32} T = \text{uint2bitstr}(\text{bitstr2uint}(S) + \text{bitstr2uint}(T) \bmod 2^{32}, 4)$.