INTERNATIONAL STANDARD

ISO/IEC 10118-1

Second edition
2000-06-15

Information technology — Security techniques — Hash-functions —

Part 1: General

Technologies de l'information — Techniques de sécurité — Fonctions de brouillage —

Partie 1: Généralités

Reference number
ISO/IEC 10118-1:2000(E)



© ISO/IEC 2000

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO 2000

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 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

© ISO/IEC 2000 – All rights reserved

Not for Resale

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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 part of ISO/IEC 10118 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 10118-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 10118-1:1994), which has been technically revised to add a general model for hash-functions. Note, however, that implementations which comply with ISO/IEC 10118-1:1994 will be compliant with this edition of ISO/IEC 10118-1.

ISO/IEC 10118 consists of the following parts, under the general title *Information technology — Security techniques — Hash-functions*:

— *Part 1: General*

— *Part 2: Hash-functions using an n-bit block cipher algorithm*

— *Part 3: Dedicated hash-functions*

— *Part 4: Hash-functions using modular arithmetic*

Annex A forms a normative part of this part of ISO/IEC 10118.

© ISO/IEC 2000 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Information technology — Security techniques — Hash-functions —

Part 1: General

1 Scope

ISO/IEC 10118 specifies hash-functions and is therefore applicable to the provision of authentication, integrity and non-repudiation services. Hash-functions map arbitrary strings of bits to a fixed-length strings of bits, using a specified algorithm. They can be used for

- reducing a message to a short imprint for input to a digital signature mechanism, and
- committing the user to a given string of bits without revealing this string.

NOTE - The hash-functions specified in this part of ISO/IEC 10118 do not involve the use of secret keys. However, these hash-functions may be used, in conjunction with secret keys, to build message authentication codes. Message Authentication Codes (MACs) provide data origin authentication as well as message integrity. For the calculation of a MAC the user is referred to ISO/IEC 9797.

This part of ISO/IEC 10118 contains definitions, symbols, abbreviations and requirements, that are common to all the other parts of ISO/IEC 10118.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10118. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 10118 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9797 (all parts), *Information technology – Security techniques – Message Authentication Codes (MACs).*

3 Terms and definitions

For the purposes of this part of ISO/IEC 10118, the following terms and definitions apply.

3.1
big-endian
a method of storage of multi-byte numbers with the most significant bytes at the lowest memory addresses

3.2
collision-resistant hash-function
a hash-function satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

NOTE – computational feasibility depends on the specific security requirements and environment.

3.3
data string (data)
a string of bits which is the input to a hash-function

© ISO/IEC 2000 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.4
hash-code
the string of bits which is the output of a hash-function

NOTE – The literature on this subject contains a variety of terms that have the same or similar meaning as hash-code. Modification Detection Code, Manipulation Detection Code, digest, hash-result, hash-value and imprint are some examples.

3.5
hash-function
a function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

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

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

NOTE – Computational feasibility depends on the specific security requirements and environment.

3.6
hash-function identifier
a byte identifying a specific hash-function

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

3.8
output transformation
a transformation or mapping of the output of the iteration stage to obtain the hash-code

3.9
padding
appending extra bits to a data string

3.10
round-function
a function $f(.,.)$ that transforms two binary strings of lengths L_1 and L_2 to a binary string of length L_2 - 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

4 Symbols (and abbreviated terms)

4.1 General Symbols

Throughout ISO/IEC 10118, the following symbols and abbreviations are used:

B_i - A byte

D - Data

D_i - A block derived from the data-string after the padding process

h - Hash-function

H - Hash-code

H_i - A string of L_2 bits which is used in the hashing operation to store an intermediate result

IV - Initializing value

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

--`,,```,,,,````-`-`,,`,,`,`,,`---
© ISO/IEC 2000 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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

L_x - Length (in bits) of a string of bits X

f - round-function (phi)

T – An output transformation function which may be a truncation or some other mapping

$X \| Y$ - Concatenation of strings of bits X and Y in the indicated order

$X \oplus Y$ - Exclusive–or of strings of bits X and Y (where $L_X = L_Y$)

4.2 Symbols specific to this part

For the purpose of this part of ISO/IEC 10118, the following symbols and notations apply :

q - The number of blocks in the data string after the padding and splitting process

4.3 Coding conventions

In contexts where the terms “most significant bit/byte” and “least significant bit/byte” have a meaning, (e.g., where strings of bits/bytes are treated as numerical values) then the leftmost bits/bytes of a block shall be the most significant.

5 Requirements

The use of a hash-function requires that the parties involved shall operate upon precisely the same bit-string, even though the representation of the data may be different in each entity’s environment. This may require one or more of the entities to convert the data into an agreed bit-string representation prior to applying a hash-function.

Some of the hash-functions specified in ISO/IEC 10118 require padding, so that the data string is of the required length. Several padding methods are presented in Annex A of this part of ISO/IEC 10118; additional padding methods may be specified in each part of ISO/IEC 10118 where padding is needed.

6 General Model for hash-functions

The hash-functions specified in this standard require the use of a round-function f. In subsequent parts of ISO/IEC 10118, several alternatives for the function f are specified.

The hash-functions which are specified in subsequent parts of ISO/IEC 10118, provide hash-codes of length L_H, where L_H is less than or equal to the value of L_2 for the round-function f being used.

6.1 Hashing Operation

Let f be a round-function and IV be an initializing value of length L_2. For the hash-functions specified in subsequent parts of ISO/IEC 10118, the value of the IV shall be fixed for a given round-function f. The hash-code H of the data D shall be calculated in four steps.

6.1.1 Step 1 (padding)

The data string D is padded in order to ensure that its length is an integer multiple of L_1. See Annex A for more information.

NOTE : Sometimes it is more efficient to have the splitting occur before the padding. The padding is then done on the last block, where $L_i < L_2$.

© ISO/IEC 2000 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.1.2 Step 2 (splitting)

The padded version of the data string D is split into L_1-bit blocks D_1 , D_2 , . . . , D_q, where D_1 represents the first L_1 bits of the padded version of D, D_2 represents the next L_1 bits, and so on. The padding and splitting processes are illustrated in Figure 1.

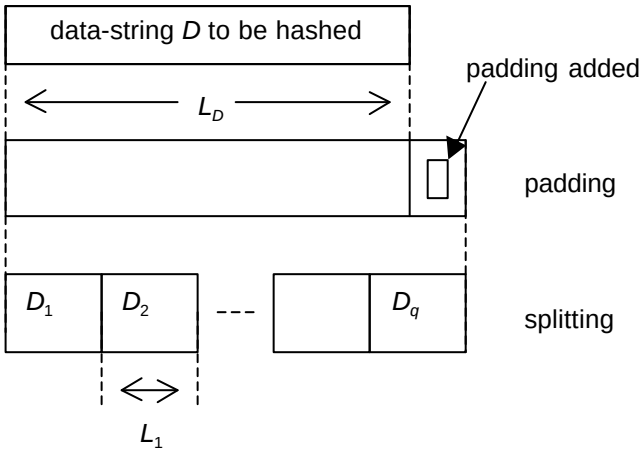

Figure 1 — The padding and splitting processes

6.1.3 Step 3 (iteration)

Let D_1, D_2, . . . D_q be the L_1-bit blocks of the data after padding and splitting. Let H_0 be a bit-string equal to IV. The L_2-bit strings H_1, H_2, . . . , H_q are calculated iteratively in the following way.

for i from 1 to q:

$$H_i = f(D_i, H_{i-1}) ;$$

The iteration process is illustrated in Figure 2.

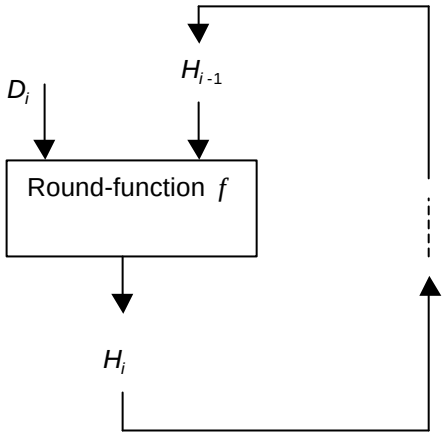

Figure 2 — The iteration process

6.1.4 Step 4 (output transformation)

The hash-code H is derived by performing a transformation T on H_q, the output of step 3, to obtain the L_H bits of the final hash-code. For example, the transformation T may be a truncation operation.

--`,,```,,,,````-`-`,,`,,`,`,,`---
© ISO/IEC 2000 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.2 Use of the general model

In subsequent parts of ISO/IEC 10118, examples of hash-functions based on the general model are specified. Specification of an individual hash-function will in each case require the following to be defined:

- parameters L_1, L_2;
- the padding method;
- the initializing value IV;
- the round function f;
- the output transformation T.

Practical use of a hash-function defined using the general model will also require the choice of the parameter L_H.

© ISO/IEC 2000 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex A
(normative)

Padding Methods

The calculation of a hash-code, as specified in other parts of ISO/IEC 10118, may require the selection of a padding method. The padding method will always output a padded data string whose length (in bits) is an exact integer multiple of L_1. Three methods are presented in this annex.

NOTE - Hash-functions using Padding Method 1 may be subject to trivial forgery attacks, where an adversary can add or delete a number of trailing '0' bits of the data string without changing the hash-code. This means that Padding Method 1 shall only be used in environments where the length of the data string D is known to the parties beforehand, or where data strings with a different number of trailing '0' bits have the same semantics. Theoretical results (see, for example, [1]) also indicate that, in environments not satisfying the above condition, Padding Method 3 may offer certain advantages over Padding Method 2.

The padding bits (if any) need not be stored or transmitted with the data. The verifier shall know whether or not the padding bits have been stored or transmitted, and which padding method is in use.

A.1 Method 1

The data for which the hash-code is to be calculated are appended with as few (possibly no) '0' bits as are necessary to obtain the required length.

A.2 Method 2

The data for which the hash-code is to be calculated are appended with a single '1' bit. The resulting data are then appended with as few (possibly none) '0' bits as are necessary to obtain the required length.

Note – Method 2 always requires the addition of at least one padding bit**.**

A.3 Method 3

This padding method requires the selection of a parameter r (where $r \leq L_1$), e.g. $r = 64$, and a method for encoding the bit length of the data D, i.e. L_D, as a bit string of length r. The choice for r will limit the length of D, in that $L_D < 2^r$.

The data D for which the hash-code is to be calculated is padded using the following procedure.

1. D is concatenated with a single '1' bit.

2. The result of the previous step is concatenated with between zero and L_1-1 '0' bits, such that the length of the resultant string is congruent to L_1-r modulo L_1. The result will be a bit string whose length will be r bits short of an integer multiple of L_1 bits (in the case $r = L_1$, the result will be a bit string whose length is an exact multiple of L_1 bits).

3. Append an r-bit encoding of L_D, using the selected encoding method, yielding the padded version of D.

© ISO/IEC 2000 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Bibliography

[1] Lai, X. and Massey, J. L., Hash Functions Based on Block Ciphers, Advances in Cryptology, EUROCRYPT'92 Proceedings, LNCS 658, Springer-Verlag, 1993.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2000 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ICS 35.040.00

Price based on 7 pages

© ISO/IEC 2000 – All rights reserved

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale