INTERNATIONAL STANDARD

ISO/IEC 10118-2

First edition
1994-10-15

Information technology — Security techniques — Hash-functions —

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

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

Partie 2: Fonctions de brouillage utilisant un algorithme de chiffrement par blocs de n *bits*

Reference number
ISO/IEC 10118-2:1994(E)

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. 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.

International Standard ISO/IEC 10118-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

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*

Annexes A, B and C of this part of ISO/IEC 10118 are for information only.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

INTERNATIONAL STANDARD © ISO/IEC **ISO/IEC 10118-2 : 1994 (E)**

Information technology - Security techniques - Hash-functions

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

1 Scope

This part of ISO/IEC 10118 specifies hash-functions which make use of an *n*-bit block cipher algorithm. They are therefore suitable for an environment in which such an algorithm is already implemented.

Two types of hash-functions are specified. The first provides hash-codes of length smaller than or equal to *n*, where *n* is the block-length of the algorithm used. The second provides hash-codes of length less than or equal to 2*n*.

2 Normative reference

The following standard contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10118. At the time of publication, the edition indicated was valid. All standards are subject to revision and parties to agreements based on this part of ISO/IEC 10118 are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 10118-1 : 1994, *Information technology - Security techniques - Hash-functions - Part 1 : General.*

3 Definitions

For the purposes of this part of ISO/IEC 10118, the definitions given in ISO/IEC 10118-1 and the following definition apply:

***n*-bit block cipher algorithm**: A block cipher algorithm with the property that plaintext blocks and ciphertext blocks are *n* bits in length. [ISO/IEC 10116]

4 Symbols and notation

For the purposes of this part of ISO/IEC 10118, the symbols and abbreviations given in ISO/IEC 10118-1 and the following symbols and abbreviations apply:

e	*n*-bit block cipher algorithm (see ISO/IEC 10116)
K	Key for the algorithm *e* (see ISO/IEC 10116)
eK	Operation of encipherment using the algorithm *e* and the key *K* (see ISO/IEC 10116)
u or u'	Transformation of one *n*-bit block into a key for the algorithm *e*
$T_{[left]}$	- when *n* is even, the string composed of the *n*/2 leftmost bits of the block *T* - when *n* is odd, the string composed of the (*n*+1)/2 leftmost bits of the block *T*
$T_{[right]}$	- when *n* is even, the string composed of the *n*/2 rightmost bits of the block *T* - when *n* is odd, the string composed of the (*n*-1)/2 rightmost bits of the block *T*

5 Requirements

Users who wish to use a hash-function from this part of ISO/IEC 10118 shall select

- an *n*-bit block cipher algorithm *e*;
- one (two) transformation(s) *u* (and *u'*);
- one (two) initializing value(s) *IV* (and *IV'*);
- a padding method;
- the length of *H* (L_H).

An example of such a selection is presented in annex A. The *n*-bit block cipher algorithm to be used is not specified in this part of ISO/IEC 10118 and may be selected from the Register of Cryptographic Algorithms, as defined in ISO/IEC 9979, or from another source. Nonetheless, it should be taken into consideration that a cryptographic property of the algorithm used may introduce some weakness into the resulting hash-function.

The two types of hash-functions make use of either one transformation, called *u*, or two transformations, called *u* and *u'*, which are not specified in this part of ISO/IEC 10118, as they depend on the algorithm used. If this algorithm has been selected from the Register of Cryptographic Algorithms, and the transformations *u* and *u'* are specified, users are encouraged to use them.

6 Hash-functions providing a single length hash-code

6.1 General

The hash-functions which are specified in this clause provide hash-codes of length L_H, where L_H is less than or equal to n.

One transformation denoted by u is used, the purpose of which is to transform an output block into a suitable L_K-bit key for the algorithm e. The specification of u is beyond the scope of this part of ISO/IEC 10118.

6.2 Hashing operation

Let e be an n-bit block cipher algorithm and IV be an initializing value of length n. IV shall be selected from a prescribed set of fixed values, the specification of which is beyond the scope of this part of ISO/IEC 10118.

The hash-code H of the data D is calculated in four steps.

6.2.1 Step 1 (splitting)

The data D are split into n-bit blocks D_1, D_2, ...

NOTE - The last block may be incomplete (i.e., its length may be less than n).

6.2.2 Step 2 (padding)

The data are padded in order to ensure that the last block has length n. The padding method is beyond the scope of this part of ISO/IEC 10118. Examples of such a method are presented in annex B of part 1 of ISO/IEC 10118.

6.2.3 Step 3 (iteration)

Let D_1, D_2, ..., D_q be the n-bit blocks of the data after padding. Set H_0 equal to IV. The output blocks H_1, H_2, ..., H_q are calculated iteratively in the following way, for i from 1 to q:

$$K_i = u(H_{i-1})$$

$$H_i = eK_i(D_i) \oplus D_i$$

Step 3 is shown in figure 1.

6.2.4 Step 4 (truncation)

The hash-code H is derived by taking the leftmost L_H bits of the final output block H_q.

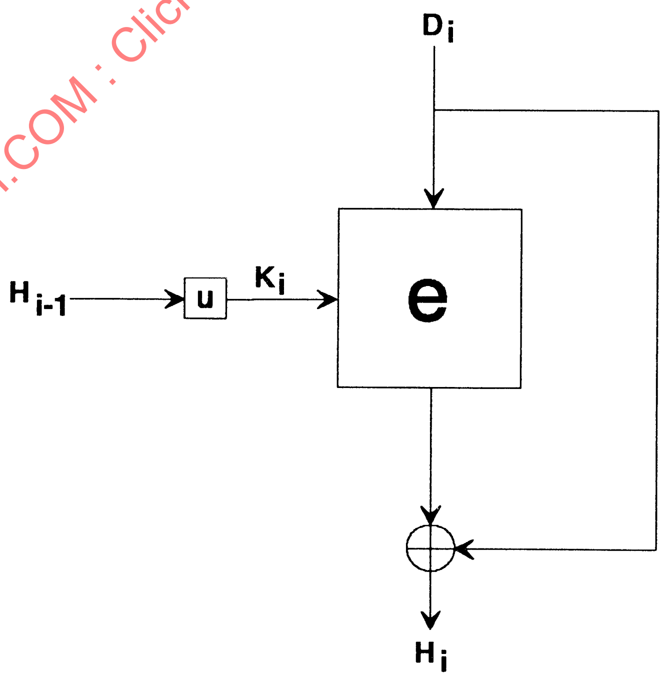

Figure 1 - Iteration of single length hashing operation

7 Hash-functions providing a double length hash-code

7.1 General

The hash-functions which are specified in this clause provide hash-codes of length L_H, where L_H is less than or equal to $2n$.

Two transformations denoted by u and u' are used, the purpose of which is to transform an output block into a suitable L_K-bit key for the algorithm e. The specification of u and u' is beyond the scope of this part of ISO/IEC 10118. However, it should be taken into consideration that the selection of u and u' is important for the security of the hash-function.

7.2 Hashing operation

Let e be an n-bit block cipher algorithm, IV and IV' be two initializing values each of length n. IV and IV' shall be selected from a prescribed set of fixed values, the specification of which is beyond the scope of this part of ISO/IEC 10118. Moreover, IV and IV' shall be selected so that $u(IV)$ and $u'(IV')$ are different.

The hash-code H of the data D is calculated in four steps.

7.2.1 Step 1 (splitting)

The data D are split into n-bit blocks D_1, D_2, ...

NOTE - The last block may be incomplete (i.e., its length may be less than n).

7.2.2 Step 2 (padding)

The data are padded in order to ensure that the last block has length n. The padding method is beyond the scope of this part of ISO/IEC 10118. Examples of such a method are presented in annex B of part 1 of ISO/IEC 10118.

7.2.3 Step 3 (iteration)

Let D_1, D_2, ..., D_q be the n-bit blocks of the data after padding. Set H_0 and H'_0 equal to IV and IV' respectively. The output blocks H_1, H_2, ..., H_q and H'_1, H'_2, ..., H'_q are calculated iteratively in the following way, for i from 1 to q :

$K_i = u(H_{i-1})$ and $K'_i = u'(H'_{i-1})$

$T_i = eK_i(D_i) \oplus D_i$ and $T'_i = eK'_i(D_i) \oplus D_i$

$H_i = T_{i[left]} \,||\, T'_{i[right]}$ and $H'_i = T'_{i[left]} \,||\, T_{i[right]}$

Step 3 is shown in figure 2.

7.2.4 Step 4 (truncation)

If L_H is even, the hash-code is the concatenation of the $L_H/2$ leftmost bits of H_q and the $L_H/2$ leftmost bits of H'_q. If L_H is odd, the hash-code is the concatenation of the $(L_H+1)/2$ leftmost bits of H_q and the $(L_H-1)/2$ leftmost bits of H'_q.

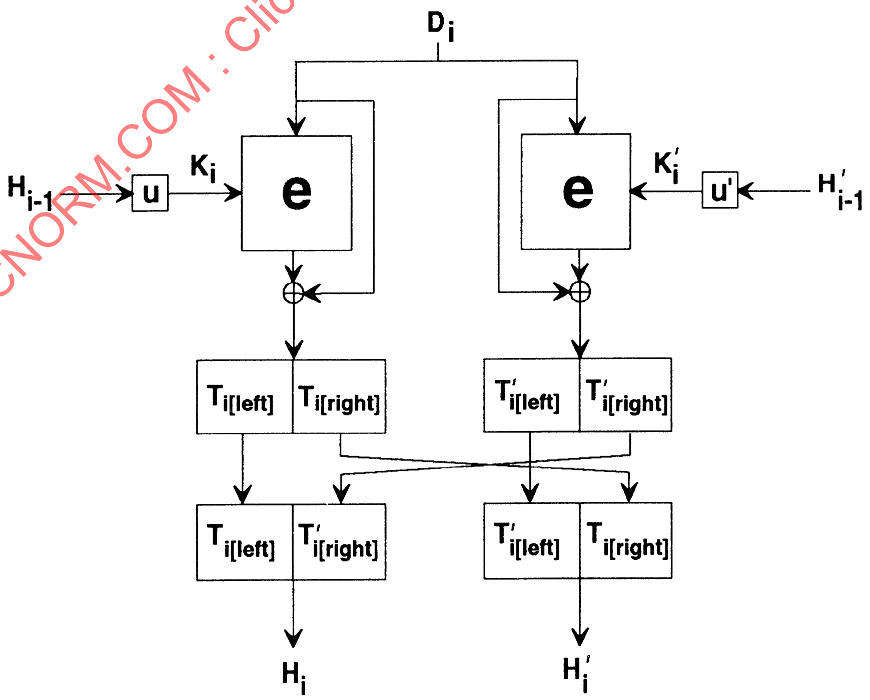

Figure 2 - Iteration of double length hashing operation

Annex A
(informative)

Use of DEA

A.1 General

This annex presents a way of using the DEA (ANSI X3.92) in conjunction with hashing operations specified in this part of ISO/IEC 10118. The DEA is also known under the name DES. These methods have been described in [4] (see annex C).

The parameters for DEA are $n = 64$ and $L_K = 56$.

A.2 Single length hash-code hashing operation

See 6.2.

IV should be equal to '5252525252525252' (in hexadecimal notation).

The transformation *u* should be chosen as follows. Let $X = x_1x_2...x_{64}$ be the binary decomposition of a 64-bit string *X*. Then $Y = u(X)$ is the string obtained after removing the bits x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} of *X* and forcing the bits x_2 and x_3 to the values '10'. The result is: $Y = x_1$'10'$x_4x_5x_6x_7x_9x_{10}...x_{63}$.

NOTE - The resulting function is believed to be a hash-function but may not be collision-resistant.

A.3 Double length hash-code hashing operation

See 7.2.

IV should be the same as in clause A.2.

IV' should be equal to '2525252525252525' (in hexadecimal notation).

The transformation *u* should be the same as in clause A.2 and the transformation *u'* should be chosen as follows. Let $X = x_1x_2...x_{64}$ be the binary decomposition of a 64-bit string *X*. Then $Y = u'(X)$ is the string obtained after removing the bits x_8, x_{16}, x_{24}, x_{32}, x_{40}, x_{48}, x_{56}, x_{64} of *X* and forcing the bits x_2 and x_3 to the values '01'. The result is: $Y = x_1$'01'$x_4x_5x_6x_7x_9x_{10}...x_{63}$.

NOTE - The resulting function is believed to be a hash-function. It is also believed to be a collision-resistant hash-function in environments where performing 2^{55} DES encipherment operations with a fixed key is deemed to be computationally infeasible.

Annex B
(informative)

Examples

B.1 General

This annex gives examples for the computation of a hash-code using the hash-functions specified in annex A of this part of ISO/IEC 10118 and the padding methods specified in annex B of part 1 of ISO/IEC 10118.

The data string is the 7-bit ASCII code (no parity) for "Now_is_the_time_for_all_", where "_" denotes a blank, in hexadecimal notation:

'4E6F77206973207468652074696D6520666F7220616C6C20'

B.2 Single length hash-code hashing operation

See A.2.

Padding method 1

i	D_i	H_{i-1}	H_i
1	4E6F772069732074	5252525252525252	858A260F7391482D
2	68652074696D6520	858A260F7391482D	BDE06E66A0454081
3	666F7220616C6C20	BDE06E66A0454081	FF87B67E29BB87B1

Padding method 2

i	D_i	H_{i-1}	H_i
1	4E6F772069732074	5252525252525252	858A260F7391482D
2	68652074696D6520	858A260F7391482D	BDE06E66A0454081
3	666F7220616C6C20	BDE06E66A0454081	FF87B67E29BB87B1
4	8000000000000000	FF87B67E29BB87B1	D992E6CBDFD9BA81

B.3 Double length hash-code hashing operation

See A.3.

Padding method 1

i	D_i	H_{i-1}	H'_{i-1}
1	4E6F772069732074	5252525252525252	2525252525252525
2	68652074696D6520	858A260FFD4873A8	49771DD37391482D
3	666F7220616C6C20	B002740352F7CF4F	CFE8087E1B93CCB2

i		H_i	H'_i
1		858A260FFD4873A8	49771DD37391482D
2		B002740352F7CF4F	CFE8087E1B93CCB2
3		42E50CD224BACEBA	760BDD2BD409281A

Padding method 2

i	D_i	H_{i-1}	H'_{i-1}
1	4E6F772069732074	5252525252525252	2525252525252525
2	68652074696D6520	858A260FFD4873A8	49771DD37391482D
3	666F7220616C6C20	B002740352F7CF4F	CFE8087E1B93CCB2
4	8000000000000000	42E50CD224BACEBA	760BDD2BD409281A

i		H_i	H'_i
1		858A260FFD4873A8	49771DD37391482D
2		B002740352F7CF4F	CFE8087E1B93CCB2
3		42E50CD224BACEBA	760BDD2BD409281A
4		2E4679B5ADD9CA75	35D87AFEAB33BEE2

Annex C
(informative)

Bibliography

[1] ISO/IEC 9979: 1991, *Data cryptographic techniques - Procedures for the registration of cryptographic algorithms.*

[2] ISO/IEC 10116: 1991, *Information technology - Security techniques - Modes of operation for an n-bit block cipher algorithm.*

[3] ANSI X3.92: 1981, *American National Standard for Information Systems - Data Encryption Algorithm.*

[4] S.M. Matyas, *Key Processing with Control Vectors, J. of Cryptology, Vol. 3, n° 2, 1991, pp. 113-136.*