INTERNATIONAL STANDARD

ISO/IEC 10118-4

First edition
1998-12-15

Information technology — Security techniques — Hash-functions —

Part 4:
Hash-functions using modular arithmetic

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

Partie 4: Fonctions de hachage utilisant l'arithmétique modulaire

Reference number
ISO/IEC 10118-4:1998(E)

Contents

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 10118-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC27, *IT Security techniques.*

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*

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

1 Scope

This part of ISO/IEC 10118 specifies two hash-functions which make use of modular arithmetic. These hash-functions, which are believed to be collision-resistant, compress messages of arbitrary but limited length to a hash-code whose length is determined by the length of the prime number used in the reduction-function defined in 7.3. Thus, the hash-code is easily scaled to the input length of any mechanism (e.g., signature algorithm, identification scheme).

The hash-functions specified in this part of ISO/IEC 10118, known as MASH-1 and MASH-2 (Modular Arithmetic Secure Hash) are particularly suitable for environments in which implementations of modular arithmetic of sufficient length are already available. The two hash-functions differ only in the exponent used in the round-function.

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 Terms and definitions

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

3.1 From ISO/IEC 10118-1

- collision-resistant hash-function
- data string (data)
- hash-code
- hash-function
- initializing value
- padding.

3.2 Unique to this part of ISO/IEC 10118

3.2.1
block

a string of bits of length L_ϕ, which shall be an integer multiple of 16 (see also clause 6.1)

EXAMPLE The length of the output H_j of the round-function.

3.2.2
half-block
a string of bits of length $L_\phi/2$

EXAMPLE Half the length of the block H_j.

3.2.3
hash-function identifier
a byte identifying a specific hash-function

3.2.4
modulus
a parameter which is a positive integer and a product of two distinct prime numbers

3.2.5
reduction-function
a function *RED* that is applied to the block H_q of length L_ϕ to generate the hash-code H of length L_p

3.2.6
round-function
a function $\phi(\cdot,\cdot)$ that transforms two binary strings of length L_ϕ to a binary string of length L_ϕ

NOTE It is used iteratively as part of a hash-function, where it combines an 'expanded' data block of length L_ϕ with the previous output of length L_ϕ.

3.3 Conventions

3.3.1 Bit ordering

Bit ordering in this part of ISO/IEC 10118 is as described in clause 3 of ISO/IEC 10118-1.

3.3.2 Converting a number to a string

During computation of the round-function, integers need to be converted to strings of L bits. Where this is required, the string of bits shall be made equal to the binary representation of the integer, with the left-most bit of the string corresponding to the most significant bit of the binary representation. If the resulting string of bits has less than L bits, then the string shall be left-padded with the appropriate number of zeros to make it of length L.

3.3.3 Converting a string to a number

During computation of the round-function, strings of bits need to be converted into integers. Where this is required, the integer shall be made equal to the number having binary representation equal to the binary string, where the left-most bit of the string is considered as the most significant bit of the binary representation.

3.4 Hash-function identifier

Identifiers are defined for each of the two MASH hash-functions specified in this standard. The hash-function identifiers for the hash-functions specified in clause 8.1 and 8.2 are equal to 41 and 42 (hexadecimal) respectively. The range of values from 43 to 4f (hexadecimal) are reserved for future use as hash-function identifiers by this part of ISO/IEC 10118.

4 Symbols and abbreviated terms

Throughout this part of ISO/IEC 10118, the following symbols and abbreviations apply.

4.1 From ISO/IEC 10118-1

D	Data
H	Hash-code
IV	Initializing value
$X \oplus Y$	Exclusive-or of strings of bits X and Y

4.2 Unique to this part of ISO/IEC 10118

B_j The jth block derived from the data string D after the padding, splitting, and expansion process.

D_j The jth half-block derived from the data string D after the padding and the splitting process. D_{q+1} through D_{q+8} are additional data blocks computed in the reduction-function.

e The exponent used in the round-function.

E A constant block equal to four ones (in the left-most position) followed by $L_\phi - 4$ zeros.

H_j The output of the round-function in the jth round. H_j has length L_ϕ.

L_D The length of the input string D in bits.

L_ϕ The length of the output H_j of the round-function ϕ. It shall be an integer multiple of 16.

L_N The length of the modulus N used in the round-function.

L_p The length of the prime number p used in the reduction-function.

mod If Z_1 is an integer and Z_2 is a positive integer, then Z_1 mod Z_2 denotes the unique integer Z_3 which satisfies

a) $0 \leq Z_3 < Z_2$, and

b) $Z_1 - Z_3$ is an integer multiple of Z_2.

N A composite integer, used as the modulus in the round-function.

NOTE For the determination of the value of N, see clause 5.

p A prime number used in the reduction-function.

NOTE For the determination of the value of p, see clause 5.

q The number of half-blocks in the data string D after the padding and splitting processes, also the number of blocks after the padding, splitting, and expansion process.

RED The reduction-function, that is applied as the last operation of the hashing procedure to reduce the block H_q of length L_ϕ to the hash-code H of length L_p.

Y_j The jth sub-string of length $L_\phi/4$ bits used in the reduction-function.

ϕ A round-function. If X and Y denote strings of L_ϕ bits, then $\phi(X,Y)$ denotes a string of L_ϕ bits obtained by applying ϕ to X and Y.

$\vee$ The bit-wise inclusive OR operation on strings of bits, i.e., if X and Y are strings of the same length, then $X \vee Y$ denotes the string obtained as the bit-wise inclusive OR of X and Y.

~ A symbol denoting the truncate operation. If X is a bitstring then $X\sim j$ denotes the bitstring obtained by taking the right-most j bits of X.

:= A symbol denoting the 'set equal to' operation. It is used in the procedural specification of the round-function and of the reduction-function, where it indicates that the block on the left side of the symbol shall be changed to equal the value of the expression on the right side of the symbol.

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

5 Requirements

5.1 To employ either of the hash-functions specified in this part of ISO/IEC 10118, two integers shall be selected: the modulus N used in the round-function and the prime p used in the reduction-function.

Both integers, N and p, are determined by the security requirements of the application for which these hash-functions are used.

5.1.1 The modulus N shall be chosen so that factoring it is computationally infeasible.

5.1.2 The modulus N shall be generated in a way that the factors remain secret. This can be accomplished by a trusted third party or by a secure multiparty computation.

NOTE 1 Generating a modulus N with the property that its factors remain secret can be accomplished by using a trusted third party, trusted hardware, and/or a secure multiparty computation. Examples can be found in Boneh [1], Cocks [2], and Frankel [3].

NOTE 2 If the factors of the modulus are kept secret, and if the size of the prime p is sufficiently large, then the best known algorithm to find a collision takes approximately $2^{L_p/2}$ evaluations of the round-function, and the best known algorithm to find a (2nd) pre-image requires approximately 2^{L_p} evaluations of the round-function. Thus in these circumstances MASH-1 and MASH-2 are believed to be collision-free hash-functions.

5.1.3 The reduction-function prime p shall not be a factor of the modulus N of the round-function.

5.1.4 The length L_p of the prime p shall be at most half of the length of the modulus N, $L_p \leq L_\phi/2$.

5.1.5 The three high order bits of prime p shall consist of ones.

5.2 To employ one of the hash-functions, MASH-1 or MASH-2, the user has to select one of the two exponents e used in the round-function ϕ.

5.3 MASH-1 and MASH-2 can be applied to all data strings D containing at most $2^{L_\phi/2}-1$ bits.

6 Variables and values needed for the hash operation

6.1 The length of the hash-code and of the modulus

The length of the modulus N and the length of the blocks H_j are related in the following manner:

$$L_\phi+1 \leq L_N \leq L_\phi+16$$

The length L_ϕ of the block H_q shall be an integer multiple of 16.

NOTE 1 If the length L_ϕ is chosen, then the length L_N is constrained by the inequalities above. If the length L_N is chosen, then the length L_ϕ will be the largest multiple of 16 less than L_N.

NOTE 2 Knowledge of N is sufficient to determine L_N, and consequently L_ϕ.

6.2 The modulus of the round-function

The modulus N used in the round-function is a composite integer generated as a product of two prime numbers of about the same length such that it is computationally infeasible to factorize N.

NOTE 1 In addition to the infeasibility of the factorization of the modulus, the security of the MASH hash-functions is based in part on the difficulty of extracting modular roots.

NOTE 2 The choice of a specific modulus N of appropriate length is outside the scope of this part of ISO/IEC 10118.

6.3 Initializing value

The initializing value *IV* is defined to be the string of L_ϕ binary zeros.

6.4 Exponent

For MASH-1 the value of the exponent e in the round-function equals 2. For MASH-2 the value of the exponent e in the round-function equals 257.

6.5 Reduction-function prime number

The reduction-function specified in 7.3 requires a prime p. The length L_p of prime p is determined by the security requirements, and by the input length of any mechanism using the hash-code. The length L_p shall be at most half of the length of the modulus N, $L_p \le L_\phi/2$.

NOTE 1 The choice of a specific prime p of appropriate length is outside the scope of this part of ISO/IEC 10118.

NOTE 2 To avoid unbalanced results by the reduction modulo p, the prime number p shall be selected with the three high order bits equal to ones.

7 Hashing procedure

The hash-code H of the data string D shall be calculated using the following steps (see Figure 1):

7.1 Preparation of the data string

The data string D is transformed into a sequence of blocks for input to the round-function ϕ. The preparation consists of padding, splitting, and expanding as detailed in the following sub-clauses.

7.1.1 Padding the data string

If the length L_D of the data string D is not an integer multiple of $L_\phi/2$, D is right-padded with binary zeros according to padding method 1 described in ISO/IEC 10118-1, Appendix B.

7.1.2 Appending the length

An additional half-block is right-appended to the data string. It contains the binary representation of the length L_D of the original (unpadded) data string D, left-padded with binary zeros (see 3.3.2).

NOTE – If the data block D is empty, only the length block is input to the hashing procedure.

7.1.3 Splitting the data string

The resulting string is divided into a sequence of q half-blocks $D_1, D_2, \ldots, D_{q-1}, D_q$.

7.1.4 Expansion

Every half-block D_j, $j = 1,2,\ldots,q$, is now doubled in length from $L_\phi/2$ bits to L_ϕ bits. This is achieved by dividing D_j into half-bytes and preceding each half-byte of D_j with a half-byte consisting of four ones (1111), for $j=1,2,\ldots,q$. The result of this process when applied to half-block D_j is denoted as B_j, $j=1,2,\ldots,q$.

7.2 Application of the round-function

The round-function ϕ on which the MASH hash-functions are based takes as input two blocks, H_{j-1} and B_j, both of length L_{ϕ} . It returns a block H_j of length L_{ϕ}. It is defined as follows:

$\phi(B_j,H_{j-1})$ =

8.2 MASH-2

For MASH-2, the round-function ϕ as specified in clause 7 becomes:

$\phi(B_j,H_{j-1}) =$