

Australian Standard[®]

**Information technology—Security
techniques—Hash-functions**

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



This Australian Standard® was prepared by Committee IT-012, Information Systems, Security and Identification. It was approved on behalf of the Council of Standards Australia on 23 June 2006.

This Standard was published on 2 August 2006.

The following are represented on Committee IT-012:

- Attorney General's Department
 - Australian Association of Permanent Building Societies
 - Australian Bankers Association
 - Australian Chamber Commerce and Industry
 - Australian Electrical and Electronic Manufacturers Association
 - Certification Forum of Australia
 - Department of Defence
 - Department of Social Welfare, NZ
 - Government Communications Security Bureau, NZ
 - Internet Industry Association
 - NSW Police Service
 - New Zealand Defence Force
 - Reserve Bank of Australia
-

This Standard was issued in draft form for comment as DR 062301.

Standards Australia wishes to acknowledge the participation of the expert individuals that contributed to the development of this Standard through their representation on the Committee and through public comment period.

Keeping Standards up-to-date

Australian Standards® are living documents that reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued.

Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments that may have been published since the Standard was published.

Detailed information about Australian Standards, drafts, amendments and new projects can be found by visiting **www.standards.org.au**

Standards Australia welcomes suggestions for improvements, and encourages readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at **mail@standards.org.au**, or write to Standards Australia, GPO Box 476, Sydney, NSW 2001.

Australian Standard[®]

Information technology—Security techniques—Hash-functions

Part 4: Hash-functions using modular arithmetic

First published as AS ISO/IEC 10118.4—2006.

COPYRIGHT

© Standards Australia

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia GPO Box 476, Sydney, NSW 2001, Australia

ISBN 0 7337 7667 1

PREFACE

This Standard was prepared by the Standards Australia Committee IT-012, Information Systems, Security and Identification.

The objective of this Standard is to specify two hash-functions which make use of modular arithmetic. These hash-functions, which are believed to be collision-resistant, compress message 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 Clause 7.3. Thus, the hash-code is easily scaled to the input length of any mechanism (e.g. signature algorithm, identification scheme).

This Standard is identical with, and has been reproduced from ISO/IEC 10118-4:1998, *Information technology—Security techniques—Hash-functions—Part 4: Hash-functions using modular arithmetic*.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this part of ISO/IEC 10118’ should read ‘this Australian Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

The term ‘informative’ has been used in this Standard to define the application of the annex to which it applies. An ‘informative’ annex is only for information and guidance.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>		<i>Australian Standard</i>	
ISO/IEC		AS ISO/IEC	
10118	Information technology—Security techniques—Hash functions	10118	Information technology—Security techniques—Hash functions
10118-1	Part 1: General	10118.1	Part 1: General

CONTENTS

	<i>Page</i>
1	Scope 1
2	Normative reference 1
3	Terms and definitions..... 1
3.1	From ISO/IEC 10118-1..... 1
3.2	Unique to this part of ISO/IEC 10118..... 1
3.3	Conventions 2
4	Symbols and abbreviated terms..... 2
4.1	From ISO/IEC 10118-1..... 2
4.2	Unique to this part of ISO/IEC 10118..... 3
5	Requirements 4
6	Variables and values needed for the hash operation..... 4
6.1	The length of the hash-code and of the modulus..... 4
6.2	The modulus of the round-function 4
6.3	Initializing value 5
6.4	Exponent..... 5
6.5	Reduction-function prime number 5
7	Hashing procedure 5
7.1	Preparation of the data string..... 5
7.1.1	Padding the data string 5
7.1.2	Appending the length 5
7.1.3	Splitting the data string 5
7.1.4	Expansion 5
7.2	Application of the round-function..... 5

	<i>Page</i>
7.3 The Reduction-function.....	6
7.3.1 Splitting the block H_q	6
7.3.2 Extending the data string.....	6
7.3.3 Processing the half-blocks	6
7.3.4 Reduction	6
8 Hash-functions	6
8.1 MASH-1	6
8.2 MASH-2	7
Annex A (informative) Examples	9
Annex B (informative) Additional Information.....	22
Annex C (informative) Bibliography	23

Information technology — Security techniques — Hash functions —

Part 4: Hash-functions using modular arithmetic

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 j th block derived from the data string D after the padding, splitting, and expansion process.
D_j	The j th 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 j th 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 $Z_1 \bmod Z_2$ 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.
<i>RED</i>	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 j th 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 .
$\sim$	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.
$X \parallel Y$	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 , $L_p \leq L_N/2$.

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.