INTERNATIONAL STANDARD

ISO/IEC 10118-3

Third edition
2004-03-01

AMENDMENT 1
2006-02-15

Information technology — Security techniques — Hash-functions —

Part 3: Dedicated hash-functions

AMENDMENT 1: Dedicated Hash-Function 8 (SHA-224)

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

Partie 3: Fonctions de brouillage dédiées

AMENDEMENT 1: Fonction de brouillage dédiée 8 (SHA-224)

Reference number
ISO/IEC 10118-3:2004/Amd.1:2006(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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.

Amendment 1 to ISO/IEC 10118-3:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

Amendment 1 to ISO/IEC 10118-3:2004 was written to serve two purposes.

First, with the inclusion of Dedicated Hash-Function 8 (SHA-224), ISO/IEC 10118-3:2004 now includes the complete family of SHA hash-functions. The description of SHA-224 is given in Clause 14. Test vectors for SHA-224 are given in A.8.

Second, it was noted that there were implementations of SHA-384 and SHA-512 in the field which correctly reproduced the test vector examples in ISO/IEC 10118-3:2004, yet still failed for inputs containing bytes that were not standard ASCII codes. In order to perform comprehensive testing of the SHA-224, SHA-256, SHA-384 and SHA-512 hash-functions, an extended set of test vectors is included for ISO/IEC 10118-3:2004 as a part of this amendment. This additional test vector information is given in A.9.

Information technology — Security techniques — Hash-functions —

Part 3: Dedicated hash-functions

AMENDMENT 1: Dedicated Hash-Function 8 (SHA-224)

Page 22

Add the following after Figure 6.

14 Dedicated Hash-Function 8 (SHA-224)

In this clause we specify a padding method, an initialising value, and a round-function for use in the general model for hash-functions described in ISO/IEC 10118-1:2000. The padding method, initialising value and round-function specified here, when used in the above general model, together define Dedicated Hash-Function 8. This dedicated hash-function can be applied to all data strings D containing at most 2^{64}-1 bits.

The ISO/IEC hash-function identifier for Dedicated Hash-Function 8 is equal to 38 (hexadecimal).

NOTE Dedicated Hash-Function 8 defined in this clause is commonly called SHA-224, [2].

14.1 Parameters, functions and constants

14.1.1 Parameters

For this hash-function $L_1 = 512$, $L_2 = 256$ and $L_H = 224$.

14.1.2 Byte ordering convention

The byte ordering convention for this hash-function is the same as that for the hash-function of clause 10.

14.1.3 Functions

The functions for this hash-function are the same as those for the hash-function of clause 10.

14.1.4 Constants

The constants for this hash-function are the same as those for the hash-function of clause 10.

14.1.5 Initialising value

For this round-function the initialising value, *IV*, shall always be the following 256-bit string, represented here as a sequence of eight words Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7 in a hexadecimal representation, where Y_0 represents the left-most 32 of the 256 bits:

Y_0 = c1059ed8
Y_1 = 367cd507
Y_2 = 3070dd17
Y_3 = f70e5939
Y_4 = ffc00b31
Y_5 = 68581511
Y_6 = 64f98fa7
Y_7 = befa4fa4

NOTE These values are the low order 32-bits of the values specified in 12.1.5.

14.2 Padding method

The padding method to be used with this hash-function shall be the same as the padding method defined in 10.2.

14.3 Description of the round-function

The round-function to be used with this hash-function shall be the same as the round-function defined in 10.3.

The final 224-bit hash is obtained by truncating the SHA-256-based hash output to its left-most 224 bits.

Page 23, Annex A

In the first line, replace “Dedicated Hash-Functions 1-7” by “Dedicated Hash-Functions 1-8”.

Page 77

Add the following after A.7.9.

A.8 Dedicated Hash-Function 8

A.8.1 Example 1

In this example the data-string is the empty string, i.e., the string of length zero.

The hash-code is the following 224-bit string.

```
d14a028c 2a3a2bc9 476102bb 288234c4 15a2b01f 828ea62a c5b3e42f
```

A.8.2 Example 2

In this example the data-string consists of a single byte, namely the ASCII-coded version of the letter 'a'.

The hash-code is the following 224-bit string.

```
abd37534 c7d9a2ef b9465de9 31cd7055 ffdb8879 563ae980 78d6d6d5
```

A.8.3 Example 3

In this example the data-string is the three-byte string consisting of the ASCII-coded version of 'abc'. This is equivalent to the bit-string: '01100001 01100010 01100011'.

After the padding process, the single 16-word block derived from the data-string is as follows.

```
61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018
```

The following are (hexadecimal representations of) the successive values of the variables Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7.

```
init: c1059ed8 367cd507 3070dd17 f70e5939 ffc00b31 68581511 64f98fa7 befa4fa4
   0 0e96b2da c1059ed8 367cd507 3070dd17 0434225e ffc00b31 68581511 64f98fa7
   1 c20dab6b 0e96b2da c1059ed8 367cd507 9cab416f 0434225e ffc00b31 68581511
   2 ab113b7a c20dab6b 0e96b2da c1059ed8 82177fe8 9cab416f 0434225e ffc00b31
   3 8253cc1a ab113b7a c20dab6b 0e96b2da 8346b27d 82177fe8 9cab416f 0434225e
   4 08a0dc0c 8253cc1a ab113b7a c20dab6b 05b557db 8346b27d 82177fe8 9cab416f
   5 b2ca3a91 08a0dc0c 8253cc1a ab113b7a 898dc7bb 05b557db 8346b27d 82177fe8
   6 0b6b9023 b2ca3a91 08a0dc0c 8253cc1a a2e49147 898dc7bb 05b557db 8346b27d
   7 f09d116d 0b6b9023 b2ca3a91 08a0dc0c 7a84120d a2e49147 898dc7bb 05b557db
   8 ed6fa633 f09d116d 0b6b9023 b2ca3a91 c037faad 7a84120d a2e49147 898dc7bb
   9 55e6a367 ed6fa633 f09d116d 0b6b9023 aae50091 c037faad 7a84120d a2e49147
  10 0817e82b 55e6a367 ed6fa633 f09d116d c8c53a2c aae50091 c037faad 7a84120d
  11 17142334 0817e82b 55e6a367 ed6fa633 dd4c7be9 c8c53a2c aae50091 c037faad
  12 fc4f023e 17142334 0817e82b 55e6a367 87bea51a dd4c7be9 c8c53a2c aae50091
  13 be316902 fc4f023e 17142334 0817e82b 65141125 87bea51a dd4c7be9 c8c53a2c
  14 1d80d178 be316902 fc4f023e 17142334 4545f53a 65141125 87bea51a dd4c7be9
  15 9f341a45 1d80d178 be316902 fc4f023e 6a61c411 4545f53a 65141125 87bea51a
  16 0f324db9 9f341a45 1d80d178 be316902 06c80d6a 6a61c411 4545f53a 65141125
  17 ffe7012b 0f324db9 9f341a45 1d80d178 b7b601f4 06c80d6a 6a61c411 4545f53a
  18 62932ab8 ffe7012b 0f324db9 9f341a45 763b627a b7b601f4 06c80d6a 6a61c411
  19 5207d867 62932ab8 ffe7012b 0f324db9 7fbba936 763b627a b7b601f4 06c80d6a
  20 07d55ccb 5207d867 62932ab8 ffe7012b 9ba5a6ea 7fbba936 763b627a b7b601f4
  21 dece98a4 07d55ccb 5207d867 62932ab8 293ffb5d 9ba5a6ea 7fbba936 763b627a
  22 e62a812e dece98a4 07d55ccb 5207d867 28fe0fd9 293ffb5d 9ba5a6ea 7fbba936
  23 57206fb8 e62a812e dece98a4 07d55ccb c76084ea 28fe0fd9 293ffb5d 9ba5a6ea
  24 6a6abcf0 57206fb8 e62a812e dece98a4 b2614c5e c76084ea 28fe0fd9 293ffb5d
  25 937514f0 6a6abcf0 57206fb8 e62a812e b42ec21c b2614c5e c76084ea 28fe0fd9
  26 82af3ffb 937514f0 6a6abcf0 57206fb8 be6f6760 b42ec21c b2614c5e c76084ea
  27 eca3bcd5 82af3ffb 937514f0 6a6abcf0 1dccbb10 be6f6760 b42ec21c b2614c5e
  28 2d1576c4 eca3bcd5 82af3ffb 937514f0 01641929 1dccbb10 be6f6760 b42ec21c
  29 fe3c8658 2d1576c4 eca3bcd5 82af3ffb fc4b36c5 01641929 1dccbb10 be6f6760
  30 0d7cce07 fe3c8658 2d1576c4 eca3bcd5 a4a4a3a4 fc4b36c5 01641929 1dccbb10
  31 cce1951d 0d7cce07 fe3c8658 2d1576c4 4be9475c a4a4a3a4 fc4b36c5 01641929
  32 09b76257 cce1951d 0d7cce07 fe3c8658 0ccddd86 4be9475c a4a4a3a4 fc4b36c5
  33 f827767e 09b76257 cce1951d 0d7cce07 db116db7 0ccddd86 4be9475c a4a4a3a4
  34 e4a0bb48 f827767e 09b76257 cce1951d 994e2bac db116db7 0ccddd86 4be9475c
  35 d8bb1041 e4a0bb48 f827767e 09b76257 5b730abb 994e2bac db116db7 0ccddd86
```

```
36 2a2e32f4 d8bb1041 e4a0bb48 f827767e 22e15c59 5b730abb 994e2bac db116db7
37 0d275ca8 2a2e32f4 d8bb1041 e4a0bb48 f6c39382 22e15c59 5b730abb 994e2bac
38 7902369c 0d275ca8 2a2e32f4 d8bb1041 d9f8c2e0 f6c39382 22e15c59 5b730abb
39 f3c80288 7902369c 0d275ca8 2a2e32f4 00e3a7bb d9f8c2e0 f6c39382 22e15c59
40 483bba4d f3c80288 7902369c 0d275ca8 f0a8198c 00e3a7bb d9f8c2e0 f6c39382
41 d75d4d26 483bba4d f3c80288 7902369c fcecdcd4 f0a8198c 00e3a7bb d9f8c2e0
42 0744b618 d75d4d26 483bba4d f3c80288 03186faa fcecdcd4 f0a8198c 00e3a7bb
43 9cce9f01 0744b618 d75d4d26 483bba4d a56f6bbf 03186faa fcecdcd4 f0a8198c
44 a3701bd9 9cce9f01 0744b618 d75d4d26 af1bef5f a56f6bbf 03186faa fcecdcd4
45 131d4c09 a3701bd9 9cce9f01 0744b618 ecb77e1b af1bef5f a56f6bbf 03186faa
46 fb3777d9 131d4c09 a3701bd9 9cce9f01 1d601f44 ecb77e1b af1bef5f a56f6bbf
47 847ea00e fb3777d9 131d4c09 a3701bd9 503a7b95 1d601f44 ecb77e1b af1bef5f
48 aaa69347 847ea00e fb3777d9 131d4c09 5eeb9930 503a7b95 1d601f44 ecb77e1b
49 505caf28 aaa69347 847ea00e fb3777d9 ce695893 5eeb9930 503a7b95 1d601f44
50 675e0b02 505caf28 aaa69347 847ea00e c22dd75f ce695893 5eeb9930 503a7b95
51 abd26099 675e0b02 505caf28 aaa69347 1409c3f8 c22dd75f ce695893 5eeb9930
52 0df9857a abd26099 675e0b02 505caf28 2d864d9f 1409c3f8 c22dd75f ce695893
53 308b8799 0df9857a abd26099 675e0b02 02524f02 2d864d9f 1409c3f8 c22dd75f
54 909cc059 308b8799 0df9857a abd26099 6f2a444a 02524f02 2d864d9f 1409c3f8
55 8d25bd94 909cc059 308b8799 0df9857a 1273c622 6f2a444a 02524f02 2d864d9f
56 f32141da 8d25bd94 909cc059 308b8799 1771ed3f 1273c622 6f2a444a 02524f02
57 8ce24395 f32141da 8d25bd94 909cc059 f52f66a6 1771ed3f 1273c622 6f2a444a
58 07bcd846 8ce24395 f32141da 8d25bd94 149db547 f52f66a6 1771ed3f 1273c622
59 622d5e5b 07bcd846 8ce24395 f32141da b6f4c630 149db547 f52f66a6 1771ed3f
60 c693fc7a 622d5e5b 07bcd846 8ce24395 13dfb889 b6f4c630 149db547 f52f66a6
61 55d1c760 c693fc7a 622d5e5b 07bcd846 7e730e00 13dfb889 b6f4c630 149db547
62 fd89031b 55d1c760 c693fc7a 622d5e5b 55489ee6 7e730e00 13dfb889 b6f4c630
63 6203de4a fd89031b 55d1c760 c693fc7a 2aedb1b3 55489ee6 7e730e00 13dfb889
```


The following eight words Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7 represent the output of the final iteration of the round-function.

$$Y_0 = \text{c1059ed8} \boxplus \text{6203de4a} = \text{23097d22}$$

$$Y_1 = \text{367cd507} \boxplus \text{fd89031b} = \text{3405d822}$$

$$Y_2 = \text{3070dd17} \boxplus \text{55d1c760} = \text{8642a477}$$

$$Y_3 = \text{f70e5939} \boxplus \text{c693fc7a} = \text{bda255b3}$$

$$Y_4 = \text{ffc00b31} \boxplus \text{2aedb1b3} = \text{2aadbce4}$$

$$Y_5 = \text{68581511} \boxplus \text{55489ee6} = \text{bda0b3f7}$$

$$Y_6 = \text{64f98fa7} \boxplus \text{7e730e00} = \text{e36c9da7}$$

$$Y_7 = \text{befa4fa4} \boxplus \text{13dfb889} = \text{ad25f72d}$$

The hash value is the following 224-bit string.

```
23097d22 3405d822 8642a477 bda255b3 2aadbce4 bda0b3f7 e36c9da7
```

A.8.4 Example 4

In this example the data-string is the 14-byte string consisting of the ASCII-coded version of

‘message digest’

The hash-code is the following 224-bit string.

```
2cb21c83 ae2f004d e7e81c3c 7019cbcb 65b71ab6 56b22d6d 0c39b8eb
```

A.8.5 Example 5

In this example the data-string is the 62-byte string consisting of the ASCII-coded version of

‘ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789’

The hash-code is the following 224-bit string.

```
bff72b4f cb7d75e5 632900ac 5f90d219 e05e97a7 bde72e74 0db393d9
```

A.8.6 Example 6

In this example the data-string is the 80-byte string consisting of the ASCII-coded version of eight repetitions of

‘1234567890’

The hash-code is the following 224-bit string.


```
b50aecbe 4e9bb0b5 7bc5f3ae 760a8e01 db24f203 fb3cdcd1 3148046e
```

A.8.7 Example 7

In this example the data-string is the 56-byte string consisting of the ASCII-coded version of

‘abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq’

After the padding process, the following two 16-word blocks are derived from the data-string.

```
61626364 62636465 63646566 64656667 65666768 66676869 6768696a 68696a6b
696a6b6c 6a6b6c6d 6b6c6d6e 6c6d6e6f 6d6e6f70 6e6f7071 80000000 00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 000001c0
```

The following are (hexadecimal representations of) the successive values of the variables Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7 in the first block process.

```
init:  c1059ed8 367cd507 3070dd17 f70e5939 ffc00b31 68581511 64f98fa7 befa4fa4
   0   0e96b2be c1059ed8 367cd507 3070dd17 04342242 ffc00b31 68581511 64f98fa7
   1   51d17d7b 0e96b2be c1059ed8 367cd507 2f8ea3d4 04342242 ffc00b31 68581511
   2   ff1cbd7f 51d17d7b 0e96b2be c1059ed8 79a896fa 2f8ea3d4 04342242 ffc00b31
   3   24bcc047 ff1cbd7f 51d17d7b 0e96b2be 1f60795a 79a896fa 2f8ea3d4 04342242
   4   7d56a6ac 24bcc047 ff1cbd7f 51d17d7b de395286 1f60795a 79a896fa 2f8ea3d4
   5   745beb11 7d56a6ac 24bcc047 ff1cbd7f d863d132 de395286 1f60795a 79a896fa
   6   0dd41573 745beb11 7d56a6ac 24bcc047 2e60d323 d863d132 de395286 1f60795a
   7   9a2541fd 0dd41573 745beb11 7d56a6ac 08d2b348 2e60d323 d863d132 de395286
```