INTERNATIONAL STANDARD

ISO/IEC 10118-3

First edition
1998-06-01

Information technology — Security techniques — Hash-functions —

Part 3:
Dedicated hash-functions

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

Partie 3: Fonctions de hachage dédiées

Reference number
ISO/IEC 10118-3:1998(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 JTC1. 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 10118–3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-Committee 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*

Further parts may follow.

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

Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions

1 Scope

This part of ISO/IEC 10118 specifies dedicated hash-functions, i.e. specially designed hash-functions. The hash-functions in this part of ISO/IEC 10118 are based on the iterative use of a round-function. Three distinct round-functions are specified, giving rise to distinct dedicated hash-functions. The first and third provide hash-codes of lengths up to 160 bits, and the second provides hash-codes of lengths up to 128 bits.

2 Normative reference

The following standard contains provisions which, through reference in the 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 definitions apply.

3.1 block: A bit-string of length L_1, i.e. the length of the first input to the round-function.

3.2 hash-function identifier: A byte identifying a specific hash-function.

3.3 round-function: A function $\phi(.,.)$ 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.

3.4 word: A string of 32 bits.

4 Symbols and notation

This part of ISO/IEC 10118 makes use of the following symbols and notation defined in ISO/IEC 10118–1.

D A data string to be input to the hash-function.

H Hash-code.

IV Initializing value.

L_X Length (in bits) of a bit-string X.

$X \oplus Y$ Exclusive-or of bit-strings X and Y.

For the purpose of this Part of ISO/IEC 10118, the following symbols and notation apply:

a_i, a'_i Sequences of indices used in specifying a round-function.

B_i A byte.

C_i, C'_i Constant words used in the round-functions.

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

f_i, g_i Functions taking three words as input and producing a single word as output, used in specifying round-functions.

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

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

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

q The number of blocks in the data string after the padding and splitting processes.

$S^n()$ The operation of 'circular left shift' by n bit positions, i.e. if A is a word and n is a non-negative integer then $S^n(A)$ denotes the word obtained by left-shifting the contents of A by n places in a cyclic fashion.

t_i, t_i' Shift-values used in specifying a round-function.

W, X_i, X_i', Y_i, Z_i Words used to store the results of intermediate computations.

ϕ A round-function, i.e. if $\mathbf{X}, \mathbf{Y}$ are bit-strings of lengths L_1 and L_2 respectively, then $\phi(\mathbf{X}, \mathbf{Y})$ is the string obtained by applying ϕ to $\mathbf{X}$ and $\mathbf{Y}$.

$\wedge$ The bit-wise logical AND operation on bit-strings, i.e. if A, B are words then $A \wedge B$ is the word equal to the bit-wise logical AND of A and B.

$\vee$ The bit-wise logical OR operation on bit-strings, i.e. if A, B are words then $A \vee B$ is the word equal to the bit-wise logical OR of A and B.

$\neg$ The bit-wise logical NOT operation on a bit-string, i.e. if A is a word then $\neg A$ is the word equal to the bit-wise logical NOT of A.

$\boxplus$ The modulo 2^{32} addition operation, i.e. if A, B are words then $A \boxplus B$ is the word obtained by treating A and B as the binary representations of integers and computing their sum modulo 2^{32}, where the result is constrained to lie between 0 and $2^{32} - 1$ inclusive.

$:=$ A symbol denoting the 'set equal to' operation used in procedural specifications of round-functions, where it indicates that the word on the left side of the symbol shall be made equal to the value of the expression on the right side of the symbol.

5 Requirements

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

- one of the dedicated hash-functions specified below; and
- the length L_H of the hash-code H.

NOTE 1 — The first and second dedicated hash-functions are defined so as to facilitate software implementations for 'little-endian' computers, i.e. where the lowest-addressed byte in a word is interpreted as the least significant; conversely, the third round-function is defined so as to facilitate software implementations for 'big-endian' computers, i.e. where the lowest-addressed byte in a word is interpreted as the most significant. However, by adjusting the definition appropriately, any of the round-functions can be implemented on a 'big-endian' or a 'little-endian' computer. All the hash-functions defined in this part of ISO/IEC 10118 take a bit-string as input and give a bit-string as output; this is independent of the internal byte-ordering convention used within each hash-function.

NOTE 2 — The choice of L_H affects the security of the hash-function. All of the hash-functions specified in this part of ISO/IEC 10118 are believed to be collision-resistant hash-functions in environments where performing $2^{L_H/2}$ hash-code computations is deemed to be computationally infeasible.

6 Model for dedicated hash-functions

6.1 General

The hash-functions specified in this standard require the use of a round-function ϕ. In subsequent clauses of this part of ISO/IEC 10118, three alternatives for the function ϕ are specified.

The hash-functions which are specified in this standard 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 ϕ being used.

In the specifications of the hash-functions in this part of ISO/IEC 10118, it is assumed that the padded data-string input to the hash-function is in the form of a sequence of bytes. If the padded data-string is in the form of a sequence of $8n$ bits, $x_0, x_1, \ldots, x_{8n-1}$, then it shall be interpreted as a sequence of n bytes, $B_0, B_1, \ldots, B_{n-1}$, in the following way. Each group of eight consecutive bits is considered as a byte, the first bit of a group being the most significant bit of that byte. Hence

$$B_i = 2^7 x_{8i} + 2^6 x_{8i+1} + \cdots + x_{8i+7}$$

for every i $(0 \leq i < n)$.

Identifiers are defined for each of the three dedicated hash-functions specified in this standard. The hash-function identifiers for the dedicated hash-functions specified in clauses 7, 8 and 9 are equal to 31, 32, and 33 (hexadecimal) respectively. The range of values from 34 to 3F (hexadecimal) are reserved for future use as hash-function identifiers by this part of ISO/IEC 10118.

6.2 Hashing operation

Let ϕ be a round-function and IV be an initializing value of length L_2. For the hash-functions specified in this part of ISO/IEC 10118, the value of the IV shall be fixed for a given round-function ϕ.

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

6.2.1 Step 1 (padding)

The data string D is padded in order to ensure that its length is a multiple of L_1. Specific instances of padding methods are specified in subsequent clauses of this part of ISO/IEC 10118.

6.2.2 Step 2 (splitting)

The padded version of the data string D is split into L_1-bit blocks $D_1, D_2, \ldots, D_q$, 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.

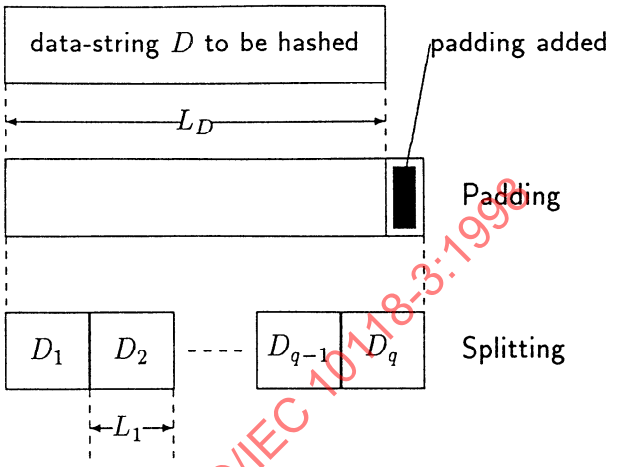

Figure 1: **Padding & splitting processes**

6.2.3 Step 3 (iteration)

Let $D_1, D_2, \ldots, D_q$ 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 $H_1, H_2, \ldots, H_q$ are calculated iteratively in the following way.

for i from 1 to q:

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

The Iteration Process is illustrated in Figure 2.

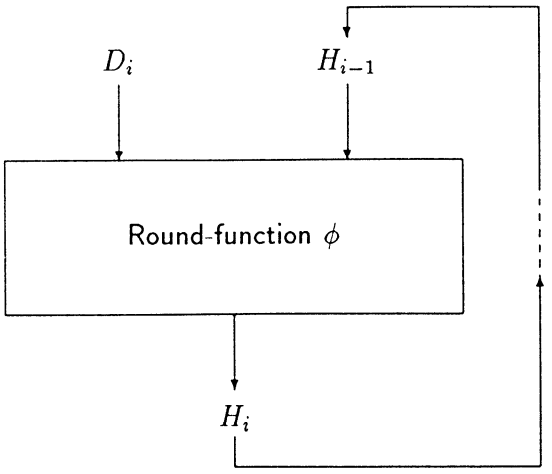

Figure 2: **The Iteration Process**

6.2.4 Step 4 (truncation)

The hash-code H is derived by taking the leftmost L_H bits of the final L_2-bit output string H_q.

7 Dedicated Hash-Function 1

NOTE — This clause contains a description of the round-function, initializing value and padding method for RIPEMD–160, [3].

7.1 General

In this clause we specify a padding method, an initializing value, and a round-function for use in the general model described in this part of ISO/IEC 10118. The padding method, initializing value and round-function specified here, when used in the above general model, together define Dedicated Hash-Function 1. 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 1 is equal to 31 (hexadecimal).

7.2 Parameters, functions and constants

7.2.1 Parameters

For this hash-function $L_1 = 512$ and $L_2 = 160$.

7.2.2 Byte ordering convention

In the specification of the round-function of clause 7 it is assumed that the block input to the round-function is in the form of a sequence of words, each 512-bit block being made up of 16 such words. A sequence of 64 bytes, $B_0, B_1, \ldots, B_{63}$, shall be interpreted as a sequence of 16 words, $Z_0, Z_1, \ldots, Z_{15}$, in the following way. Each group of four consecutive bytes is considered as a word, the first byte of a word being the least significant byte of that word. Hence

$$Z_i = 2^{24}B_{4i+3} + 2^{16}B_{4i+2} + 2^8 B_{4i+1} + B_{4i}, \quad (0 \le i \le 15).$$

To convert the hash-code from a sequence of words to a byte-sequence, the inverse process shall be followed.

NOTE — The byte-ordering specified here is different from that of subclause 9.2.2.

7.2.3 Functions

To facilitate software implementation, the round-function ϕ is described in terms of operations on words. A sequence of functions $g_0, g_1, \ldots, g_{79}$ is used in this round-function, where each function g_i, $0 \le i \le 79$, takes three words X_0, X_1 and X_2 as input and produces a single word as output.

The functions g_i are defined as follows:

$$\begin{aligned}
g_i(X_0, X_1, X_2) &= X_0 \oplus X_1 \oplus X_2, \quad (0 \le i \le 15),\\
g_i(X_0, X_1, X_2) &= (X_0 \wedge X_1) \vee (\neg X_0 \wedge X_2),\\
&\quad (16 \le i \le 31),\\
g_i(X_0, X_1, X_2) &= (X_0 \vee \neg X_1) \oplus X_2, \quad (32 \le i \le 47),\\
g_i(X_0, X_1, X_2) &= (X_0 \wedge X_2) \vee (X_1 \wedge \neg X_2),\\
&\quad (48 \le i \le 63),\\
g_i(X_0, X_1, X_2) &= X_0 \oplus (X_1 \vee \neg X_2), \quad (64 \le i \le 79).
\end{aligned}$$

7.2.4 Constants

Two sequences of constant words $C_0, C_1, \ldots, C_{79}$ and $C'_0, C'_1, \ldots, C'_{79}$ are used in this round-function. In a hexadecimal representation (where the most significant bit corresponds to the left-most bit) these are defined as follows:

$$\begin{aligned}
C_i &= \texttt{00000000}, \quad (0 \le i \le 15),\\
C_i &= \texttt{5A827999}, \quad (16 \le i \le 31),\\
C_i &= \texttt{6ED9EBA1}, \quad (32 \le i \le 47),\\
C_i &= \texttt{8F1BBCDC}, \quad (48 \le i \le 63),\\
C_i &= \texttt{A953FD4E}, \quad (64 \le i \le 79),
\end{aligned}$$

$$\begin{aligned}
C'_i &= \texttt{50A28BE6}, \quad (0 \le i \le 15),\\
C'_i &= \texttt{5C4DD124}, \quad (16 \le i \le 31),\\
C'_i &= \texttt{6D703EF3}, \quad (32 \le i \le 47),\\
C'_i &= \texttt{7A6D76E9}, \quad (48 \le i \le 63),\\
C'_i &= \texttt{00000000}, \quad (64 \le i \le 79).
\end{aligned}$$

Two sequences of 80 shift-values are used in this round-function, where each shift-value is between 5 and 15. We denote these sequences by $(t_0, t_1, \ldots, t_{79})$ and $(t'_0, t'_1, \ldots, t'_{79})$. A further two sequences of 80 indices are used in this round-function, where each value in the sequence is between 0 and 15. We denote these sequences as $(a_0, a_1, \ldots, a_{79})$, and $(a'_0, a'_1, \ldots, a'_{79})$. All four sequences are defined in the following table.

i	0	1	2	3	4	5	6	7
t_i	11	14	15	12	5	8	7	9
t'_i	8	9	9	11	13	15	15	5
a_i	0	1	2	3	4	5	6	7
a'_i	5	14	7	0	9	2	11	4
i	8	9	10	11	12	13	14	15
t_i	11	13	14	15	6	7	9	8
t'_i	7	7	8	11	14	14	12	6
a_i	8	9	10	11	12	13	14	15
a'_i	13	6	15	8	1	10	3	12
i	16	17	18	19	20	21	22	23
t_i	7	6	8	13	11	9	7	15
t'_i	9	13	15	7	12	8	9	11
a_i	7	4	13	1	10	6	15	3
a'_i	6	11	3	7	0	13	5	10
i	24	25	26	27	28	29	30	31
t_i	7	12	15	9	11	7	13	12
t'_i	7	7	12	7	6	15	13	11
a_i	12	0	9	5	2	14	11	8
a'_i	14	15	8	12	4	9	1	2
i	32	33	34	35	36	37	38	39
t_i	11	13	6	7	14	9	13	15
t'_i	9	7	15	11	8	6	6	14
a_i	3	10	14	4	9	15	8	1
a'_i	15	5	1	3	7	14	6	9
i	40	41	42	43	44	45	46	47
t_i	14	8	13	6	5	12	7	5
t'_i	12	13	5	14	13	13	7	5
a_i	2	7	0	6	13	11	5	12
a'_i	11	8	12	2	10	0	4	13
i	48	49	50	51	52	53	54	55
t_i	11	12	14	15	14	15	9	8
t'_i	15	5	8	11	14	14	6	14
a_i	1	9	11	10	0	8	12	4
a'_i	8	6	4	1	3	11	15	0
i	56	57	58	59	60	61	62	63
t_i	9	14	5	6	8	6	5	12
t'_i	6	9	12	9	12	5	15	8
a_i	13	3	7	15	14	5	6	2
a'_i	5	12	2	13	9	7	10	14
i	64	65	66	67	68	69	70	71
t_i	9	15	5	11	6	8	13	12
t'_i	8	5	12	9	12	5	14	6
a_i	4	0	5	9	7	12	2	10
a'_i	12	15	10	4	1	5	8	7

i	72	73	74	75	76	77	78	79
t_i	5	12	13	14	11	8	5	6
t'_i	8	13	6	5	15	13	11	11
a_i	14	1	3	8	11	6	15	13
a'_i	6	2	13	14	0	3	9	11

7.2.5 Initializing Value

For this round-function the initializing value, IV, shall always be the following 160-bit string, represented here as a sequence of five words Y_0, Y_1, Y_2, Y_3, Y_4 in a hexadecimal representation, where Y_0 represents the left-most 32 of the 160 bits:

$$\begin{aligned} Y_0 &= \texttt{67452301}, \\ Y_1 &= \texttt{EFCDAB89}, \\ Y_2 &= \texttt{98BADCFE}, \\ Y_3 &= \texttt{10325476}, \\ Y_4 &= \texttt{C3D2E1F0}. \end{aligned}$$

7.3 Padding method

The data string D needs to be padded to make it contain a number of bits which is an integer multiple of 512. The padding procedure operates as follows:

1. D is concatenated with a single '1' bit.
2. The result of the previous step is concatenated with between zero and 511 '0' bits such that the length (in bits) of the resultant string is congruent to 448 modulo 512. More explicitly, if the original length of D is L_D, and letting r be the remainder when L_D is divided by 512, then the number of concatenated zeros is equal to either $447 - r$ (if $r \leq 447$) or $959 - r$ (if $r > 447$). The result will be a bit string whose length will be 64 bits short of an integer multiple of 512 bits.
3. Divide the 64-bit binary representation of L_D into two 32-bit strings, one representing the 'most significant half' of L_D and the other the 'least significant half'. Now concatenate the string resulting from the previous step with these two 32-bit strings, with the 'least significant half' preceding the 'most significant half'.

In the description of the round-function which follows, each 512-bit data block D_i, $1 \leq i \leq q$, is treated as a sequence of 16 words, $Z_0, Z_1, \ldots, Z_{15}$, where Z_0 corresponds to the left-most 32 bits of D_i.

7.4 Description of the round-function

The round-function ϕ operates as follows. Note that, in this description, we use the symbols $W, X_0, X_1, X_2, X_3, X_4, X'_0, X'_1, X'_2, X'_3, X'_4$ to denote eleven distinct words which contain values required in the computations.

1. Suppose the 512-bit (first) input to ϕ is contained in $Z_0, Z_1, \ldots, Z_{15}$, where Z_0 contains the left-most 32 of the 512 bits. Suppose also that the 160-bit (second) input to ϕ is contained in five words, Y_0, Y_1, Y_2, Y_3, Y_4.

2. Let $X_0 := Y_0$, $X_1 := Y_1$, $X_2 := Y_2$, $X_3 := Y_3$ and $X_4 := Y_4$.

3. Let $X'_0 := Y_0$, $X'_1 := Y_1$, $X'_2 := Y_2$, $X'_3 := Y_3$ and $X'_4 := Y_4$.

4. For $i := 0$ to 79 do the following four steps in the order specified:

   (a) $W := S^{t_i}(X_0 \boxplus g_i(X_1, X_2, X_3) \boxplus Z_{a_i} \boxplus C_i) \boxplus X_4$;

   (b) $X_0 := X_4$; $X_4 := X_3$; $X_3 := S^{10}(X_2)$; $X_2 := X_1$; $X_1 := W$;

   (c) $W := S^{t'_i}(X'_0 \boxplus g_{79-i}(X'_1, X'_2, X'_3) \boxplus Z_{a'_i} \boxplus C'_i) \boxplus X'_4$;

   (d) $X'_0 := X'_4$; $X'_4 := X'_3$; $X'_3 := S^{10}(X'_2)$; $X'_2 := X'_1$; $X'_1 := W$;

5. Let

$$\begin{aligned} W &:= Y_0, \\ Y_0 &:= Y_1 \boxplus X_2 \boxplus X'_3, \\ Y_1 &:= Y_2 \boxplus X_3 \boxplus X'_4, \\ Y_2 &:= Y_3 \boxplus X_4 \boxplus X'_0, \\ Y_3 &:= Y_4 \boxplus X_0 \boxplus X'_1, \\ Y_4 &:= W \boxplus X_1 \boxplus X'_2. \end{aligned}$$

6. The five words Y_0, Y_1, Y_2, Y_3, Y_4 represent the output of the round-function ϕ. After the final iteration of the round-function, the five words Y_0, Y_1, Y_2, Y_3, Y_4 shall be converted to a sequence of 20 bytes using the inverse of the procedure specified in 7.2.2, and where Y_0 shall yield the first four bytes, Y_1 the next four bytes, and so on. Thus the first (left-most) byte will correspond to the least significant byte of Y_0, and the 20th (right-most) byte will corespond to the most significant byte of Y_4. The 20 bytes shall be converted to a string of 160 bits using the inverse of the procedure specified in 6.1, i.e. the first (left-most) bit will correspond to the most significant bit of the first (left-most) byte, and the 160th (right-most) bit will correspond to the least significant bit of the 20th (right-most) byte.

8 Dedicated Hash-Function 2

NOTE — This clause contains a description of the round-function, initializing value and padding method for RIPEMD–128, [3].

This hash-function should only be used in applications where a hash-code containing 128 bits or less is considered adequately secure.

8.1 General

In this clause we specify a padding method, an initializing value, and a round-function for use in the general model described in this part of ISO/IEC 10118. The padding method, initializing value and round-function specified here, when used in the above general model, together define Dedicated Hash-Function 2. 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 2 is equal to 32 (hexadecimal).

8.2 Parameters, functions and constants

8.2.1 Parameters

For this hash-function $L_1 = 512$ and $L_2 = 128$.

8.2.2 Byte ordering convention

The byte ordering convention for this hash-function is the same as that for the hash-function of clause 7.

8.2.3 Functions

To facilitate software implementation, the round-function ϕ is described in terms of operations on words. A sequence of functions $g_0, g_1, \ldots, g_{63}$ is used in this round-function, where each function g_i, $0 \leq i \leq 63$, takes three words X_0, X_1 and X_2 as input and produces a single word as output.

The functions g_i are defined to be the same as the first 64 of the functions defined in subclause 7.2.3.

8.2.4 Constants

Two sequences of constant words $C_0, C_1, \ldots, C_{63}$ and $C'_0, C'_1, \ldots, C'_{63}$ are used in this round-function. In a hexadecimal representation (where the most significant bit corresponds to the left-most bit) these are defined as follows:

$$\begin{aligned} C_i &= \texttt{00000000}, \quad (0 \le i \le 15), \\ C_i &= \texttt{5A827999}, \quad (16 \le i \le 31), \\ C_i &= \texttt{6ED9EBA1}, \quad (32 \le i \le 47), \\ C_i &= \texttt{8F1BBCDC}, \quad (48 \le i \le 63), \end{aligned}$$

$$\begin{aligned} C'_i &= \texttt{50A28BE6}, \quad (0 \le i \le 15), \\ C'_i &= \texttt{5C4DD124}, \quad (16 \le i \le 31), \\ C'_i &= \texttt{6D703EF3}, \quad (32 \le i \le 47), \\ C'_i &= \texttt{00000000}, \quad (48 \le i \le 63). \end{aligned}$$

Two sequences of 64 shift-values are also used in this round-function, where each shift-value is between 5 and 15. We denote these sequences by $(t_0, t_1, \ldots, t_{63})$ and $(t'_0, t'_1, \ldots, t'_{63})$, and they are defined to be equal to the first 64 values of the corresponding sequences defined in subclause 7.2.4.

Finally, two further sequences of 64 indices are used in this round-function, where each value in the sequence is between 0 and 15. We denote these sequences by $(a_0, a_1, \ldots, a_{63})$, and $(a'_0, a'_1, \ldots, a'_{63})$, and they are defined to be equal to the first 64 values of the corresponding sequences defined in subclause 7.2.4.

8.2.5 Initializing Value

For this hash-function the initializing value, IV, shall always be the following 128-bit string, represented here as a sequence of four words Y_0, Y_1, Y_2, Y_3 in a hexadecimal representation, where Y_0 represents the left-most 32 of the 128 bits:

$$\begin{aligned} Y_0 &= \texttt{67452301}, \\ Y_1 &= \texttt{EFCDAB89}, \\ Y_2 &= \texttt{98BADCFE}, \\ Y_3 &= \texttt{10325476}. \end{aligned}$$

8.3 Padding method

The padding method to be used with this hash-function shall be the same as the padding method defined in subclause 7.3.

8.4 Description of the round-function

The round-function ϕ operates as follows. Note that, in this description, we use the symbols $W, X_0, X_1, X_2, X_3, X'_0, X'_1, X'_2, X'_3$ to denote nine distinct words which contain values required in the computations.

1. Suppose the 512-bit (first) input to ϕ is contained in $Z_0, Z_1, \ldots, Z_{15}$, where Z_0 contains the left-most 32 of the 512 bits. Suppose also that the 128-bit (second) input to ϕ is contained in four words, Y_0, Y_1, Y_2, Y_3.

2. Let $X_0 := Y_0$, $X_1 := Y_1$, $X_2 := Y_2$ and $X_3 := Y_3$.

3. Let $X'_0 := Y_0$, $X'_1 := Y_1$, $X'_2 := Y_2$ and $X'_3 := Y_3$.

4. For $i := 0$ to 63 do the following four steps in the order specified:

   (a) $W := S^{t_i}(X_0 \boxplus g_i(X_1, X_2, X_3) \boxplus Z_{a_i} \boxplus C_i)$;

   (b) $X_0 := X_3$; $X_3 := X_2$; $X_2 := X_1$; $X_1 := W$;

   (c) $W := S^{t'_i}(X'_0 \boxplus g_{63-i}(X'_1, X'_2, X'_3) \boxplus Z_{a'_i} \boxplus C'_i)$;

   (d) $X'_0 := X'_3$; $X'_3 := X'_2$; $X'_2 := X'_1$; $X'_1 := W$;

5. Let

$$\begin{aligned} W &:= Y_0, \\ Y_0 &:= Y_1 \boxplus X_2 \boxplus X'_3, \\ Y_1 &:= Y_2 \boxplus X_3 \boxplus X'_0, \\ Y_2 &:= Y_3 \boxplus X_0 \boxplus X'_1, \\ Y_3 &:= W \boxplus X_1 \boxplus X'_2. \end{aligned}$$

6. The four words Y_0, Y_1, Y_2, Y_3 represent the output of the round-function ϕ. After the final iteration of the round-function, the four words Y_0, Y_1, Y_2, Y_3 shall be converted to a sequence of 16 bytes using the inverse of the procedure specified in 7.2.2, and where Y_0 shall yield the first four bytes, Y_1 the next four bytes, and so on. Thus the first (left-most) byte will correspond to the least significant byte of Y_0, and the 16th (right-most) byte will corespond to the most significant byte of Y_3. The 16 bytes shall be converted to a string of 128 bits using the inverse of the procedure specified in 6.1, i.e. the

first (left-most) bit will correspond to the most significant bit of the first (left-most) byte, and the 128th (right-most) bit will correspond to the least significant bit of the 16th (right-most) byte.

9 Dedicated Hash-Function 3

NOTE — This clause contains a description of the round-function, initializing value and padding method for SHA–1 (the US NIST 'Secure Hash Algorithm'), [2].

9.1 General

In this clause we specify a padding method, an initializing value, and a round-function for use in the general model described in this part of ISO/IEC 10118. The padding method, initializing value and round-function specified here, when used in the above general model, together define Dedicated Hash-Function 3. 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 3 is equal to 33 (hexadecimal).

9.2 Parameters, functions and constants

9.2.1 Parameters

For this hash-function $L_1 = 512$ and $L_2 = 160$.

9.2.2 Byte ordering convention

In the specification of the round-function of clause 9 it is assumed that the block input to the round-function is in the form of a sequence of words, each 512-bit block being made up of 16 such words. A sequence of 64 bytes, $B_0, B_1, \ldots, B_{63}$, shall be interpreted as a sequence of 16 words, $Z_0, Z_1, \ldots, Z_{15}$, in the following way. Each group of four consecutive bytes is considered as a word, the first byte of a word being the most significant byte of that word. Hence

$$Z_i = 2^{24}B_{4i} + 2^{16}B_{4i+1} + 2^8 B_{4i+2} + B_{4i+3}, \quad (0 \le i \le 15).$$

To convert the hash-code from a sequence of words to a sequence of bytes, the inverse process shall be followed.

NOTE — The byte-ordering specified here is different from that of subclause 7.2.2.

9.2.3 Functions

To facilitate software implementation, the round-function ϕ is described in terms of operations on words. A sequence of functions $f_0, f_1, \ldots, f_{79}$ is used in this round-function, where each function f_i, $0 \le i \le 79$, takes three words X_0, X_1 and X_2 as input and produces a single word as output.

The functions f_i are defined as follows:

$$\begin{aligned}
f_i(X_0, X_1, X_2) &= (X_0 \wedge X_1) \vee (\neg X_0 \wedge X_2), \\
&\quad (0 \le i \le 19), \\
f_i(X_0, X_1, X_2) &= X_0 \oplus X_1 \oplus X_2, \quad (20 \le i \le 39), \\
f_i(X_0, X_1, X_2) &= (X_0 \wedge X_1) \vee (X_0 \wedge X_2) \vee (X_1 \wedge X_2), \\
&\quad (40 \le i \le 59), \\
f_i(X_0, X_1, X_2) &= X_0 \oplus X_1 \oplus X_2, \quad (60 \le i \le 79).
\end{aligned}$$

9.2.4 Constants

A sequence of constant words $C_0, C_1, \ldots, C_{79}$ is used in this round-function. In a hexadecimal representation (where the most significant bit corresponds to the left-most bit) these are defined as follows:

$$\begin{aligned}
C_i &= \texttt{5A827999}, \quad (0 \le i \le 19), \\
C_i &= \texttt{6ED9EBA1}, \quad (20 \le i \le 39), \\
C_i &= \texttt{8F1BBCDC}, \quad (40 \le i \le 59), \\
C_i &= \texttt{CA62C1D6}, \quad (60 \le i \le 79).
\end{aligned}$$

9.2.5 Initializing Value

For this round-function the initializing value, IV, shall always be the following 160-bit string, represented here as a sequence of five words Y_0, Y_1, Y_2, Y_3, Y_4 in a hexadecimal representation, where Y_0 represents the left-most 32 of the 160 bits:

$$\begin{aligned}
Y_0 &= \texttt{67452301}, \\
Y_1 &= \texttt{EFCDAB89}, \\
Y_2 &= \texttt{98BADCFE}, \\
Y_3 &= \texttt{10325476}, \\
Y_4 &= \texttt{C3D2E1F0}.
\end{aligned}$$

9.3 Padding method

The data string D needs to be padded to make it contain a number of bits which is an integer multiple of 512. The padding procedure operates as follows:

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

2. The result of the previous step is concatenated with between zero and 511 '0' bits such that the length (in bits) of the resultant string is congruent to 448 modulo 512. More explicitly, if the original length of D is L_D, and letting r be the remainder when L_D is divided by 512, then the number of concatenated zeros is equal to either $447 - r$ (if $r \leq 447$) or $959 - r$ (if $r > 447$). The result will be a bit string whose length will be 64 bits short of an integer multiple of 512 bits.

3. Concatenate the string resulting from the previous step with the 64-bit binary representation of L_D, most significant bit first.

In the description of the round-function which follows, each 512-bit data block D_i, $1 \leq i \leq q$, is treated as a sequence of 16 words, $Z_0, Z_1, \ldots, Z_{15}$, where Z_0 corresponds to the left-most 32 bits of D_i.

9.4 Description of the round-function

The round-function ϕ operates as follows. Note that, in this description, we use the symbols $W, X_0, X_1, X_2, X_3, X_4, Z_0, Z_1, \ldots, Z_{79}$ to denote 86 distinct words which contain values required in the computations.

1. Suppose the 512-bit (first) input to ϕ is contained in $Z_0, Z_1, \ldots, Z_{15}$, where Z_0 contains the left-most 32 of the 512 bits. Suppose also that the 160-bit (second) input to ϕ is contained in five words, Y_0, Y_1, Y_2, Y_3, Y_4.

2. For $i = 16$ to 79 let

$$Z_i := S^1(Z_{i-3} \oplus Z_{i-8} \oplus Z_{i-14} \oplus Z_{i-16}).$$

3. Let $X_0 := Y_0$, $X_1 := Y_1$, $X_2 := Y_2$, $X_3 := Y_3$ and $X_4 := Y_4$.

4. For $i = 0$ to 79 do the following two steps

   (a) $W := S^5(X_0) \uplus f_i(X_1, X_2, X_3) \uplus X_4 \uplus Z_i \uplus C_i$;

   (b) $X_4 := X_3$; $X_3 := X_2$; $X_2 := S^{30}(X_1)$; $X_1 := X_0$; $X_0 := W$.

5. Let $Y_0 := Y_0 \uplus X_0$, $Y_1 := Y_1 \uplus X_1$, $Y_2 := Y_2 \uplus X_2$, $Y_3 := Y_3 \uplus X_3$ and $Y_4 := Y_4 \uplus X_4$.

6. The five words Y_0, Y_1, Y_2, Y_3, Y_4 represent the output of the round-function ϕ. After the final iteration of the round-function, the five words Y_0, Y_1, Y_2, Y_3, Y_4 shall be converted to a sequence of 20 bytes using the inverse of the procedure specified in 9.2.2, and where Y_0 shall yield the first four bytes, Y_1 the next four bytes, and so on. Thus the first (left-most) byte will correspond to the most significant byte of Y_0, and the 20th (right-most) byte will corespond to the least significant byte of Y_4. The 20 bytes shall be converted to a string of 160 bits using the inverse of the procedure specified in 6.1, i.e. the first (left-most) bit will correspond to the most significant bit of the first (left-most) byte, and the 160th (right-most) bit will correspond to the least significant bit of the 20th (right-most) byte.

Annex A
(informative)

Examples

A.1 General

This annex gives examples for the computation of Dedicated Hash-Functions 1, 2 and 3. Nine examples of hash-code calculation are given for each of the hash-functions. For each of the hash-functions, intermediate values derived during the hash-function's operation are given for examples numbers 3 and 8.

A.2 Dedicated Hash-Function 1

Throughout this annex we refer to ASCII coding of data strings; this is equivalent to coding using ISO 646.

NOTE — Reference [3] contains a pseudocode description of Dedicated Hash-Function 1.

A.2.1 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 160-bit string.

```
9C 11 85 A5 C5 E9 FC 54 61 28 08 97 7E E8 F5 48 B2 25 8D 31
```

A.2.2 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 160-bit string.

```
0B DC 9D 2D 25 6B 3E E9 DA AE 34 7B E6 F4 DC 83 5A 46 7F FE
```

A.2.3 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.

```
80636261 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000018 00000000
```

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X_4, X'_0, X'_1, X'_2, X'_3, X'_4.

```
67452301, EFCDAB89, 98BADCFE, 10325476, C3D2E1F0, 67452301, EFCDAB89, 98BADCFE, 10325476, C3D2E1F0
C3D2E1F0, 3115FC67, EFCDAB89, EB73FA62, 10325476, C3D2E1F0, DDD63FB8, EFCDAB89, EB73FA62, 10325476
10325476, B41192D5, 3115FC67, 36AE27BF, EB73FA62, 10325476, 322E7AE3, DDD63FB8, 36AE27BF, EB73FA62
EB73FA62, 3A35DC50, B41192D5, 57F19CC4, 36AE27BF, EB73FA62, 883EE903, 322E7AE3, 58FEE377, 36AE27BF
```

36AE27BF, D3786413, 3A35DC50, 464B56D0, 57F19CC4, 36AE27BF, 92B2B79B, 883EE903, B9EB8CC8, 58FEE377
57F19CC4, 0E946720, D3786413, D77140E8, 464B56D0, 58FEE377, F9091FF2, 92B2B79B, FBA40E20, B9EB8CC8
464B56D0, D52BF632, 0E946720, E1904F4D, D77140E8, B9EB8CC8, E5B09992, F9091FF2, CADE6E4A, FBA40E20
D77140E8, 150BD8A8, D52BF632, 519C803A, E1904F4D, FBA40E20, 8B2D9FB3, E5B09992, 247FCBE4, CADE6E4A
E1904F4D, 3D6F601F, 150BD8A8, AFD8CB54, 519C803A, CADE6E4A, E755F422, 8B2D9FB3, C2664B96, 247FCBE4
519C803A, B7B60384, 3D6F601F, 2F62A054, AFD8CB54, 247FCBE4, 5922D09E, E755F422, B67ECE2C, C2664B96
AFD8CB54, B85A0A3F, B7B60384, BD807CF5, 2F62A054, C2664B96, CF24E72C, 5922D09E, 57D08B9D, B67ECE2C
2F62A054, 7F8B38E5, B85A0A3F, D80E12DE, BD807CF5, B67ECE2C, CA6A1C75, CF24E72C, 8B427964, 57D08B9D
BD807CF5, 9DACA495, 7F8B38E5, 6828FEE1, D80E12DE, 57D08B9D, 227F6D84, CA6A1C75, 939CB33C, 8B427964
D80E12DE, BC05F46F, 9DACA495, 2CE395FE, 6828FEE1, 8B427964, 5D801685, 227F6D84, A871D729, 939CB33C
6828FEE1, 1494F053, BC05F46F, B2925676, 2CE395FE, 939CB33C, B3C3F4D5, 5D801685, FDB61089, A871D729
2CE395FE, 85861D02, 1494F053, 17D1BEF0, B2925676, A871D729, 3D16242D, B3C3F4D5, 005A1576, FDB61089
B2925676, 597BF629, 85861D02, 53C14C52, 17D1BEF0, FDB61089, FF459078, 3D16242D, 0FD356CF, 005A1576
17D1BEF0, 6347EF78, 597BF629, 18740A16, 53C14C52, 005A1576, 927E40A8, FF459078, 5890B4F4, 0FD356CF
53C14C52, 45C8FA44, 6347EF78, EFD8A565, 18740A16, 0FD356CF, ACBB994E, 927E40A8, 1641E3FD, 5890B4F4
18740A16, AD2956AF, 45C8FA44, 1FBDE18D, EFD8A565, 5890B4F4, AD30AD24, ACBB994E, F902A249, 1641E3FD
EFD8A565, 5EAF16B7, AD2956AF, 23E91117, 1FBDE18D, 1641E3FD, 6261732E, AD30AD24, EE653AB2, F902A249
1FBDE18D, 41730D4B, 5EAF16B7, A55ABEB4, 23E91117, F902A249, 45ED27AF, 6261732E, C2B492B4, EE653AB2
23E91117, FC0CCBD3, 41730D4B, BC5ADD7A, A55ABEB4, EE653AB2, 243C5668, 45ED27AF, 85CCB989, C2B492B4
A55ABEB4, 042ECC93, FC0CCBD3, CC352D05, BC5ADD7A, C2B492B4, 82F89BD1, 243C5668, B49EBD17, 85CCB989
BC5ADD7A, 4D4D4377, 042ECC93, 332F4FF0, CC352D05, 85CCB989, 5FC74686, 82F89BD1, F159A090, B49EBD17
CC352D05, 5207002B, 4D4D4377, BB324C10, 332F4FF0, B49EBD17, B2720031, 5FC74686, E26F460B, F159A090
332F4FF0, 388278F5, 5207002B, 350DDD35, BB324C10, F159A090, 58A100F8, B2720031, 1D1A197F, E26F460B
BB324C10, 62879D70, 388278F5, 1C00AD48, 350DDD35, E26F460B, 5992068B, 58A100F8, C800C6C9, 1D1A197F
350DDD35, A30A1FD9, 62879D70, 09E3D4E2, 1C00AD48, 1D1A197F, CC290DCA, 5992068B, 8403E162, C800C6C9
1C00AD48, BDA2B31B, A30A1FD9, 1E75C18A, 09E3D4E2, C800C6C9, 863D625E, CC290DCA, 481A2D66, 8403E162
09E3D4E2, F7211DEE, BDA2B31B, 287F668C, 1E75C18A, 8403E162, 6061B5A5, 863D625E, A4372B30, 481A2D66
1E75C18A, B6A665C6, F7211DEE, 8ACC6EF6, 287F668C, 481A2D66, AA98ADB5, 6061B5A5, F5897A18, A4372B30
287F668C, 2D30FA02, B6A665C6, 8477BBDC, 8ACC6EF6, A4372B30, 2999255A, AA98ADB5, 86D69581, F5897A18
8ACC6EF6, C76D12F9, 2D30FA02, 99971ADA, 8477BBDC, F5897A18, 98237631, 2999255A, 62B6D6AA, 86D69581
8477BBDC, 516F84DF, C76D12F9, C3E808B4, 99971ADA, 86D69581, 6C472A90, 98237631, 649568A6, 62B6D6AA
99971ADA, F3FA5B05, 516F84DF, B44BE71D, C3E808B4, 62B6D6AA, 2EAD5672, 6C472A90, 8DD8C660, 649568A6
C3E808B4, D539625E, F3FA5B05, BE137D45, B44BE71D, 649568A6, C5CB48BA, 2EAD5672, 1CAA41B1, 8DD8C660
B44BE71D, D8500C99, D539625E, E96C17CF, BE137D45, 8DD8C660, 05286DFB, C5CB48BA, B559C8BA, 1CAA41B1
BE137D45, 7ECDE5B2, D8500C99, E5897B54, E96C17CF, 1CAA41B1, 88396DD2, 05286DFB, 2D22EB17, B559C8BA
E96C17CF, 681D30B9, 7ECDE5B2, 40326761, E5897B54, B559C8BA, 333F2212, 88396DD2, A1B7EC14, 2D22EB17
E5897B54, 960F7BFD, 681D30B9, 3796C9FB, 40326761, 2D22EB17, C699295B, 333F2212, E5B74A20, A1B7EC14
40326761, 6770E498, 960F7BFD, 74C2E5A0, 3796C9FB, A1B7EC14, BFD68874, C699295B, FC8848CC, E5B74A20
3796C9FB, 75EB06C5, 6770E498, 3DEFF658, 74C2E5A0, E5B74A20, BDDF3474, BFD68874, 64A56F1A, FC8848CC
74C2E5A0, 14FA827A, 75EB06C5, C392619D, 3DEFF658, FC8848CC, 8CBC87E9, BDDF3474, 5A21D2FF, 64A56F1A
3DEFF658, 804B0068, 14FA827A, AC1B15D7, C392619D, 64A56F1A, CDDA6EBF, 8CBC87E9, 7CD1D2F7, 5A21D2FF
C392619D, 475BA81B, 804B0068, EA09E853, AC1B15D7, 5A21D2FF, 656C7DA3, CDDA6EBF, F21FA632, 7CD1D2F7
AC1B15D7, D26BC25D, 475BA81B, 2C01A201, EA09E853, 7CD1D2F7, 76D66CA3, 656C7DA3, 69BAFF37, F21FA632
EA09E853, DBC5A2CB, D26BC25D, 6EA06D1D, 2C01A201, F21FA632, C9B17F72, 76D66CA3, B1F68D95, 69BAFF37
2C01A201, 77367F5E, DBC5A2CB, AF097749, 6EA06D1D, 69BAFF37, 65A60151, C9B17F72, 59B28DDB, B1F68D95
6EA06D1D, 8155A6B4, 77367F5E, 168B2F6F, AF097749, B1F68D95, 33F3AC81, 65A60151, C5FDCB26, 59B28DDB
AF097749, C90C4D38, 8155A6B4, D9FD79DC, 168B2F6F, 59B28DDB, 9BFB827D, 33F3AC81, 98054596, C5FDCB26
168B2F6F, 9762713B, C90C4D38, 569AD205, D9FD79DC, C5FDCB26, DDC8130E, 9BFB827D, CEB204CF, 98054596
D9FD79DC, 7EBF9C32, 9762713B, 3134E324, 569AD205, 98054596, C24C2C79, DDC8130E, EE09F66F, CEB204CF
569AD205, 20EFFA01, 7EBF9C32, 89C4EE5D, 3134E324, CEB204CF, F255847E, C24C2C79, 204C3B77, EE09F66F
3134E324, 75B7117F, 20EFFA01, FE70C9FA, 89C4EE5D, EE09F66F, DCD63949, F255847E, 30B1E709, 204C3B77
89C4EE5D, A96BE4C7, 75B7117F, BFE80483, FE70C9FA, 204C3B77, 5B99238D, DCD63949, 5611FBC9, 30B1E709
FE70C9FA, 5E3201FC, A96BE4C7, DC45FDD6, BFE80483, 30B1E709, B43484F4, 5B99238D, 58E52773, 5611FBC9
BFE80483, 2CF95A98, 5E3201FC, AF931EA5, DC45FDD6, 5611FBC9, 52325A09, B43484F4, 648E356E, 58E52773
DC45FDD6, 1393F0C3, 2CF95A98, C807F178, AF931EA5, 58E52773, D015577D, 52325A09, D213D2D0, 648E356E
AF931EA5, BB49CCF7, 1393F0C3, E56A60B3, C807F178, 648E356E, BB9C87C4, D015577D, C9682548, D213D2D0
C807F178, 6A330EB4, BB49CCF7, 4FC30C4E, E56A60B3, D213D2D0, B1BB1A2E, BB9C87C4, 555DF740, C9682548
E56A60B3, 14E58204, 6A330EB4, 2733DEED, 4FC30C4E, C9682548, AC77F96D, B1BB1A2E, 721F12EE, 555DF740
4FC30C4E, 79AAF53E, 14E58204, CC3AD1A8, 2733DEED, 555DF740, 1774D326, AC77F96D, EC68BAC6, 721F12EE
2733DEED, 210769B3, 79AAF53E, 96081053, CC3AD1A8, 721F12EE, A625F112, 1774D326, DFE5B6B1, EC68BAC6
CC3AD1A8, F44B53A7, 210769B3, ABD4F9E6, 96081053, EC68BAC6, 5DCA4D12, A625F112, D34C985D, DFE5B6B1

```
96081053, 7C1E3640, F44B53A7, 1DA6CC84, ABD4F9E6, DFE5B6B1, EBC4D9C6, 5DCA4D12, 97C44A98, D34C985D
ABD4F9E6, 06B59EE8, 7C1E3640, 2D4E9FD1, 1DA6CC84, D34C985D, 095F37FD, EBC4D9C6, 29344977, 97C44A98
1DA6CC84, C422C3CD, 06B59EE8, 78D901F0, 2D4E9FD1, 97C44A98, 5BBEE487, 095F37FD, 13671BAF, 29344977
2D4E9FD1, AD864025, C422C3CD, D67BA01A, 78D901F0, 29344977, BF5B2529, 5BBEE487, 7CDFF425, 13671BAF
78D901F0, 29A83BB5, AD864025, 8B0F3710, D67BA01A, 13671BAF, FB5747C5, BF5B2529, FB921D6E, 7CDFF425
D67BA01A, 626E3910, 29A83BB5, 190096B6, 8B0F3710, 7CDFF425, DD935A5F, FB5747C5, 6C94A6FD, FB921D6E
8B0F3710, A719D8BC, 626E3910, A0EED4A6, 190096B6, FB921D6E, 27754F3A, DD935A5F, 5D1F17ED, 6C94A6FD
190096B6, BA84C782, A719D8BC, B8E44189, A0EED4A6, 6C94A6FD, 4F5CA4A5, 27754F3A, 4D697F76, 5D1F17ED
A0EED4A6, 9F6887A9, BA84C782, 6762F29C, B8E44189, 5D1F17ED, 325AFE7E, 4F5CA4A5, D53CE89D, 4D697F76
B8E44189, 3A88288C, 9F6887A9, 131E0AEA, 6762F29C, 4D697F76, 86AFE021, 325AFE7E, 7292953D, D53CE89D
6762F29C, AB23F78F, 3A88288C, A21EA67D, 131E0AEA, D53CE89D, C97F9EA1, 86AFE021, 6BF9F8C9, 7292953D
131E0AEA, 7299044A, AB23F78F, 20A230EA, A21EA67D, 7292953D, 9F60751C, C97F9EA1, BF80861A, 6BF9F8C9
A21EA67D, 6A3F10CF, 7299044A, 8FDE3EAC, 20A230EA, 6BF9F8C9, 1E9CE713, 9F60751C, FE7A8725, BF80861A
20A230EA, 1A1B904D, 6A3F10CF, 641129CA, 8FDE3EAC, BF80861A, C13F038A, 1E9CE713, 81D4727D, FE7A8725
8FDE3EAC, 0B2CDC01, 1A1B904D, FC433DA8, 641129CA, FE7A8725, BF627814, C13F038A, 739C4C7A, 81D4727D
641129CA, D563BFDC, 0B2CDC01, 6E413468, FC433DA8, 81D4727D, 5FCCBADE, BF627814, FC0E2B04, 739C4C7A
```

The hash-code is the following 160-bit string.

```
8E B2 08 F7 E0 5D 98 7A 9B 04 4A 8E 98 C6 B0 87 F1 5A 0B FC
```

A.2.4 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 160-bit string.

```
5D 06 89 EF 49 D2 FA E5 72 B8 81 B1 23 A8 5F FA 21 59 5F 36
```

A.2.5 Example 5

In this example the data-string is the 26-byte string consisting of the ASCII-coded version of

'abcdefghijklmnopqrstuvwxyz'

The hash-code is the following 160-bit string.

```
F7 1C 27 10 9C 69 2C 1B 56 BB DC EB 5B 9D 28 65 B3 70 8D BC
```

A.2.6 Example 6

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 160-bit string.

```
B0 E2 0B 6E 31 16 64 02 86 ED 3A 87 A5 71 30 79 B2 1F 51 89
```

A.2.7 Example 7

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 160-bit string.

9B 75 2E 45 57 3D 4B 39 F4 DB D3 32 3C AB 82 BF 63 32 6B FB

A.2.8 Example 8

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

'abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq'

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

64636261	65646362	66656463	67666564	68676665	69686766	6A696867	6B6A6968
6C6B6A69	6D6C6B6A	6E6D6C6B	6F6E6D6C	706F6E6D	71706F6E	00000080	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	000001C0	00000000

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X_4, X'_0, X'_1, X'_2, X'_3, X'_4, obtained during the processing of the first block.

67452301, EFCDAB89, 98BADCFE, 10325476, C3D2E1F0, 67452301, EFCDAB89, 98BADCFE, 10325476, C3D2E1F0
C3D2E1F0, 3115FB87, EFCDAB89, EB73FA62, 10325476, C3D2E1F0, 463DA521, EFCDAB89, EB73FA62, 10325476
10325476, CC21EC2E, 3115FB87, 36AE27BF, EB73FA62, 10325476, DB247A12, 463DA521, 36AE27BF, EB73FA62
EB73FA62, DFEB9B7A, CC21EC2E, 57EE1CC4, 36AE27BF, EB73FA62, 1D166A23, DB247A12, F6948518, 36AE27BF
36AE27BF, 2363912E, DFEB9B7A, 87B0BB30, 57EE1CC4, 36AE27BF, CE7A12F6, 1D166A23, 91E84B6C, F6948518
57EE1CC4, A1B60DC7, 2363912E, AE6DEB7F, 87B0BB30, F6948518, 57FF19DD, CE7A12F6, 59A88C74, 91E84B6C
87B0BB30, 96AC7C1E, A1B60DC7, 8E44B88D, AE6DEB7F, 91E84B6C, 01A9FEFA, 57FF19DD, E84BDB39, 59A88C74
AE6DEB7F, 6AE46154, 96AC7C1E, D8371E86, 8E44B88D, 59A88C74, 5D9A609C, 01A9FEFA, FC67755F, E84BDB39
8E44B88D, 3CF61F09, 6AE46154, B1F07A5A, D8371E86, E84BDB39, 030F7FE7, 5D9A609C, A7FBE806, FC67755F
D8371E86, 696F0D9A, 3CF61F09, 918551AB, B1F07A5A, FC67755F, 7456C8E3, 030F7FE7, 69827176, A7FBE806
B1F07A5A, AB957B91, 696F0D9A, D87C24F3, 918551AB, A7FBE806, F64C4453, 7456C8E3, 3DFF9C0C, 69827176
918551AB, 9FF4A064, AB957B91, BC3669A5, D87C24F3, 69827176, 22A5FE6E, F64C4453, 5B238DD1, 3DFF9C0C
D87C24F3, 912FE998, 9FF4A064, 55EE46AE, BC3669A5, 3DFF9C0C, 8D7E53E4, 22A5FE6E, 31114FD9, 5B238DD1
BC3669A5, C45F164E, 912FE998, D281927F, 55EE46AE, 5B238DD1, 695B23B7, 8D7E53E4, 97F9B88A, 31114FD9
55EE46AE, 2211A508, C45F164E, BFA66244, D281927F, 31114FD9, 6FAA776F, 695B23B7, F94F9235, 97F9B88A
D281927F, 80B1F3DE, 2211A508, 7C593B11, BFA66244, 97F9B88A, 4D94F720, 6FAA776F, 6C8EDDA5, F94F9235
BFA66244, 3AA6A8F5, 80B1F3DE, 46942088, 7C593B11, F94F9235, D81C6137, 4D94F720, A9DDBDBE, 6C8EDDA5
7C593B11, 9E4C4BF6, 3AA6A8F5, C7CF7A02, 46942088, 6C8EDDA5, B2ECCABD, D81C6137, 53DC8136, A9DDBDBE
46942088, F929216E, 9E4C4BF6, 9AA3D4EA, C7CF7A02, A9DDBDBE, A96B1820, B2ECCABD, 7184DF60, 53DC8136
C7CF7A02, D9AEEFAF, F929216E, 312FDA79, 9AA3D4EA, 53DC8136, 5A5E09B3, A96B1820, B32AF6CB, 7184DF60
9AA3D4EA, 8BB34505, D9AEEFAF, A485BBE4, 312FDA79, 7184DF60, 616711FA, 5A5E09B3, AC6082A5, B32AF6CB
312FDA79, 07067302, 8BB34505, BBBEBF66, A485BBE4, B32AF6CB, F4F47116, 616711FA, 7826CD69, AC6082A5
A485BBE4, 51997747, 07067302, CD14162E, BBBEBF66, AC6082A5, FAE97297, F4F47116, 9C47E985, 7826CD69
BBBEBF66, C213132C, 51997747, 19CC081C, CD14162E, 7826CD69, 887E5A3F, FAE97297, D1C45BD3, 9C47E985
CD14162E, 29D001F0, C213132C, 65DD1D46, 19CC081C, 9C47E985, 187068EF, 887E5A3F, A5CA5FEB, D1C45BD3
19CC081C, 2B59B58A, 29D001F0, 4C4CB308, 65DD1D46, D1C45BD3, 56C66FD3, 187068EF, F968FE21, A5CA5FEB

65DD1D46, C45681A6, 2B59B58A, 4007C0A7, 4C4CB308, A5CA5FEB, D718432A, 56C66FD3, C1A3BC61, F968FE21
4C4CB308, 2E32CA16, C45681A6, 66D628AD, 4007C0A7, F968FE21, 775BA27D, D718432A, 19BF4D5B, C1A3BC61
4007C0A7, 5C712D51, 2E32CA16, 5A069B11, 66D628AD, C1A3BC61, 6243D22F, 775BA27D, 610CAB5C, 19BF4D5B
66D628AD, 989BC126, 5C712D51, CB2858B8, 5A069B11, 19BF4D5B, 44DCD35A, 6243D22F, 6E89F5DD, 610CAB5C
5A069B11, 9EE4CA1F, 989BC126, C4B54571, CB2858B8, 610CAB5C, 8FBE3F7E, 44DCD35A, 0F48BD89, 6E89F5DD
CB2858B8, F417F849, 9EE4CA1F, 6F049A62, C4B54571, 6E89F5DD, DA718428, 8FBE3F7E, 734D6913, 0F48BD89
C4B54571, 75239882, F417F849, 93287E7B, 6F049A62, 0F48BD89, 91573E0A, DA718428, F8FDFA3E, 734D6913
6F049A62, 3AC6B69F, 75239882, 5FE127D0, 93287E7B, 734D6913, 2A5224A6, 91573E0A, C610A369, F8FDFA3E
93287E7B, 0B7C24AC, 3AC6B69F, 8E6209D4, 5FE127D0, F8FDFA3E, 8128FFB7, 2A5224A6, 5CF82A45, C610A369
5FE127D0, 2854DCE0, 0B7C24AC, 1ADA7CEB, 8E6209D4, C610A369, FF374DFD, 8128FFB7, 489298A9, 5CF82A45
8E6209D4, 267080E2, 2854DCE0, F092B02D, 1ADA7CEB, 5CF82A45, C5E0CCD7, FF374DFD, A3FEDE04, 489298A9
1ADA7CEB, 7806D96F, 267080E2, 537380A1, F092B02D, 489298A9, 31860C44, C5E0CCD7, DD37F7FC, A3FEDE04
F092B02D, 52638496, 7806D96F, C2038899, 537380A1, A3FEDE04, CEE7092B, 31860C44, 83335F17, DD37F7FC
537380A1, 59FC5CDB, 52638496, 1B65BDE0, C2038899, DD37F7FC, 46827AAE, CEE7092B, 183110C6, 83335F17
C2038899, 8AE30FBE, 59FC5CDB, 8E125949, 1B65BDE0, 83335F17, A757A907, 46827AAE, 9C24AF3B, 183110C6
1B65BDE0, 4F4AEBED, 8AE30FBE, F1736D67, 8E125949, 183110C6, E90F38FC, A757A907, 09EAB91A, 9C24AF3B
8E125949, 65BBCCCC, 4F4AEBED, 8C3EFA2B, F1736D67, 9C24AF3B, EC65CB85, E90F38FC, 5EA41E9D, 09EAB91A
F1736D67, 0B3B88C1, 65BBCCCC, 2BAFB53D, 8C3EFA2B, 09EAB91A, 54B06FBD, EC65CB85, 3CE3F3A4, 5EA41E9D
8C3EFA2B, 6DF30989, 0B3B88C1, EF333196, 2BAFB53D, 5EA41E9D, D8D6F0E3, 54B06FBD, 972E17B1, 3CE3F3A4
2BAFB53D, 156421AC, 6DF30989, EE23042C, EF333196, 3CE3F3A4, B30DA892, D8D6F0E3, C1BEF552, 972E17B1
EF333196, 6F54F9CA, 156421AC, CC2625B7, EE23042C, 972E17B1, F526A85A, B30DA892, 5BC38F63, C1BEF552
EE23042C, A5D28921, 6F54F9CA, 9086B055, CC2625B7, C1BEF552, 5F5587DB, F526A85A, 36A24ACC, 5BC38F63
CC2625B7, 2959D915, A5D28921, 53E729BD, 9086B055, 5BC38F63, 9FABAC24, 5F5587DB, 9AA16BD4, 36A24ACC
9086B055, 4EFF0384, 2959D915, 4A248697, 53E729BD, 36A24ACC, 52E4FB9B, 9FABAC24, 561F6D7D, 9AA16BD4
53E729BD, 17292945, 4EFF0384, 676454A5, 4A248697, 9AA16BD4, E13C3BDA, 52E4FB9B, AEB0927E, 561F6D7D
4A248697, 5FE71F22, 17292945, FC0E113B, 676454A5, 561F6D7D, 71244E49, E13C3BDA, 93EE6D4B, AEB0927E
676454A5, DC06A80F, 5FE71F22, A4A5145C, FC0E113B, AEB0927E, AA49234C, 71244E49, F0EF6B84, 93EE6D4B
FC0E113B, 5BD21FC5, DC06A80F, 9C7C897F, A4A5145C, 93EE6D4B, 42532D95, AA49234C, 913925C4, F0EF6B84
A4A5145C, 5587BC4F, 5BD21FC5, 1AA03F70, 9C7C897F, F0EF6B84, CDA86FD0, 42532D95, 248D32A9, 913925C4
9C7C897F, A1755F6B, 5587BC4F, 487F156F, 1AA03F70, 913925C4, 69C12F76, CDA86FD0, 4CB65509, 248D32A9
1AA03F70, 100A6B19, A1755F6B, 1EF13D56, 487F156F, 248D32A9, 44272219, 69C12F76, A1BF4336, 4CB65509
487F156F, AA2CFD07, 100A6B19, D57DAE85, 1EF13D56, 4CB65509, CBD360C3, 44272219, 04BDD9A7, A1BF4336
1EF13D56, 28246D22, AA2CFD07, 29AC6440, D57DAE85, A1BF4336, 27A64C2D, CBD360C3, 9C886510, 04BDD9A7
D57DAE85, 4909C2BD, 28246D22, B3F41EA8, 29AC6440, 04BDD9A7, CCB70B88, 27A64C2D, 4D830F2F, 9C886510
29AC6440, 9020271B, 4909C2BD, 91B488A0, B3F41EA8, 9C886510, 2020C0FC, CCB70B88, 9930B49E, 4D830F2F
B3F41EA8, A557D838, 9020271B, 270AF524, 91B488A0, 4D830F2F, 7541E108, 2020C0FC, DC2E2332, 9930B49E
91B488A0, F879D1F8, A557D838, 809C6E40, 270AF524, 9930B49E, 0A66EBF9, 7541E108, 8303F080, DC2E2332
270AF524, 39BAC08A, F879D1F8, 5F60E295, 809C6E40, DC2E2332, A0AB24D8, 0A66EBF9, 078421D5, 8303F080
809C6E40, DF212B9C, 39BAC08A, E747E3E1, 5F60E295, 8303F080, 44C068DD, A0AB24D8, 9BAFE429, 078421D5
5F60E295, 46F2CD86, DF212B9C, EB0228E6, E747E3E1, 078421D5, 3F8B3B48, 44C068DD, AC936282, 9BAFE429
E747E3E1, A17766F4, 46F2CD86, 84AE737C, EB0228E6, 9BAFE429, 873A41C4, 3F8B3B48, 01A37513, AC936282
EB0228E6, FC20AA01, A17766F4, CB36191B, 84AE737C, AC936282, A2969EB4, 873A41C4, 2CED20FE, 01A37513
84AE737C, 93A30DD9, FC20AA01, DD9BD285, CB36191B, 01A37513, 7B345F4F, A2969EB4, E907121C, 2CED20FE
CB36191B, 98554E1C, 93A30DD9, 82A807F0, DD9BD285, 2CED20FE, 07B2EA78, 7B345F4F, 5A7AD28A, E907121C
DD9BD285, 79D46BD1, 98554E1C, 8C37664E, 82A807F0, E907121C, 93451653, 07B2EA78, D17D3DEC, 5A7AD28A
82A807F0, 5FBC55DB, 79D46BD1, 55387261, 8C37664E, 5A7AD28A, AA0DF949, 93451653, CBA9E01E, D17D3DEC
8C37664E, DEF23A3B, 5FBC55DB, 51AF45E7, 55387261, D17D3DEC, 030FFB9A, AA0DF949, 14594E4D, CBA9E01E
55387261, 287DB1EB, DEF23A3B, F1576D7E, 51AF45E7, CBA9E01E, 0D9CD217, 030FFB9A, 37E526A8, 14594E4D
51AF45E7, CF955B8E, 287DB1EB, C8E8EF7B, F1576D7E, 14594E4D, BECE1BBD, 0D9CD217, 3FEE680C, 37E526A8
F1576D7E, 83B6B7E8, CF955B8E, F6C7ACA1, C8E8EF7B, 37E526A8, D97CFEEC, BECE1BBD, 73485C36, 3FEE680C
C8E8EF7B, 7943C443, 83B6B7E8, 556E3B3E, F6C7ACA1, 3FEE680C, DBEA79F5, D97CFEEC, 386EF6FB, 73485C36
F6C7ACA1, F336AA45, 7943C443, DADFA20E, 556E3B3E, 73485C36, 91704BDB, DBEA79F5, F3FBB365, 386EF6FB
556E3B3E, 2FF847D6, F336AA45, 0F110DE5, DADFA20E, 386EF6FB, 40CBA97D, 91704BDB, A9E7D76F, F3FBB365
DADFA20E, 33FE64C9, 2FF847D6, DAA917CC, 0F110DE5, F3FBB365, B0BD2456, 40CBA97D, C12F6E45, A9E7D76F
0F110DE5, 78378FE9, 33FE64C9, E11F58BF, DAA917CC, A9E7D76F, CA09D415, B0BD2456, 2EA5F503, C12F6E45

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X_4, X'_0, X'_1, X'_2, X'_3, X'_4, obtained during the processing of the second block.

52720555, 3B09A402, 94C343B1, 9CEDC3EA, 9039D740, 52720555, 3B09A402, 94C343B1, 9CEDC3EA, 9039D740

9039D740, 59874B6C, 3B09A402, 0D0EC653, 9CEDC3EA, 9039D740, 7FA6C9AF, 3B09A402, 0D0EC653, 9CEDC3EA
9CEDC3EA, 1D0D43D8, 59874B6C, 269008EC, 0D0EC653, 9CEDC3EA, 149F92B4, 7FA6C9AF, 269008EC, 0D0EC653
0D0EC653, EF3045D6, 1D0D43D8, 1D2DB166, 269008EC, 0D0EC653, 0E887E05, 149F92B4, 9B26BDFE, 269008EC
269008EC, 1E6BC8AD, EF3045D6, 350F6074, 1D2DB166, 269008EC, 6E8757AC, 0E887E05, 7E4AD052, 9B26BDFE
1D2DB166, 79CC70E3, 1E6BC8AD, C1175BBC, 350F6074, 9B26BDFE, 32C1290B, 6E8757AC, 21F8143A, 7E4AD052
350F6074, 13A4B937, 79CC70E3, AF22B479, C1175BBC, 7E4AD052, 8EB02C5A, 32C1290B, 1D5EB1BA, 21F8143A
C1175BBC, EE066CB9, 13A4B937, 31C38DE7, AF22B479, 21F8143A, 719EB9D9, 8EB02C5A, 04A42CCB, 1D5EB1BA
AF22B479, A08AFF93, EE066CB9, 92E4DC4E, 31C38DE7, 1D5EB1BA, 3D5B8A9A, 719EB9D9, C0B16A3A, 04A42CCB
31C38DE7, 89E27A43, A08AFF93, 19B2E7B8, 92E4DC4E, 04A42CCB, 47DEA0A3, 3D5B8A9A, 7AE765C6, C0B16A3A
92E4DC4E, 50EEC8A1, 89E27A43, 2BFE4E82, 19B2E7B8, C0B16A3A, A6AACEE1, 47DEA0A3, 6E2A68F5, 7AE765C6
19B2E7B8, 0FDE892D, 50EEC8A1, 89E90E27, 2BFE4E82, 7AE765C6, 4456D048, A6AACEE1, 7A828D1F, 6E2A68F5
2BFE4E82, 47B046C8, 0FDE892D, BB228543, 89E90E27, 6E2A68F5, 072D166E, 4456D048, AB3B869A, 7A828D1F
89E90E27, 5C8F582E, 47B046C8, 7A24B43F, BB228543, 7A828D1F, B37A11D1, 072D166E, 5B412111, AB3B869A
BB228543, 3D7F05B8, 5C8F582E, C11B211E, 7A24B43F, AB3B869A, 654CBE94, B37A11D1, B459B81C, 5B412111
7A24B43F, 962BCAF7, 3D7F05B8, 3D60B972, C11B211E, 5B412111, 6AFF9ABA, 654CBE94, E84746CD, B459B81C
C11B211E, 1A459D2E, 962BCAF7, FC16E0F5, 3D60B972, B459B81C, EE0E390E, 6AFF9ABA, 32FA5195, E84746CD
3D60B972, 1622907A, 1A459D2E, AF2BDE58, FC16E0F5, E84746CD, 569023C2, EE0E390E, FE6AE9AB, 32FA5195
FC16E0F5, B75B2E49, 1622907A, 1674B869, AF2BDE58, 32FA5195, 5C2944E8, 569023C2, 38E43BB8, FE6AE9AB
AF2BDE58, 6F16D4C4, B75B2E49, 8A41E858, 1674B869, FE6AE9AB, 103CE067, 5C2944E8, 408F095A, 38E43BB8
1674B869, 46FDEE89, 6F16D4C4, 6CB926DD, 8A41E858, 38E43BB8, AB641473, 103CE067, A513A170, 408F095A
8A41E858, E9F89F50, 46FDEE89, 5B5311BC, 6CB926DD, 408F095A, 25643DBF, AB641473, F3819C40, A513A170
6CB926DD, EC9A614C, E9F89F50, F7BA251B, 5B5311BC, A513A170, E60A5336, 25643DBF, 9051CEAD, F3819C40
5B5311BC, D525F69D, EC9A614C, E27D43A7, F7BA251B, F3819C40, FF4D318D, E60A5336, 90F6FC95, 9051CEAD
F7BA251B, EDFBF331, D525F69D, 698533B2, E27D43A7, 9051CEAD, 6D5A28DD, FF4D318D, 294CDB98, 90F6FC95
E27D43A7, 93C5E732, EDFBF331, 97DA7754, 698533B2, 90F6FC95, 855C140A, 6D5A28DD, 34C637FD, 294CDB98
698533B2, 24907FDF, 93C5E732, EFCCC7B7, 97DA7754, 294CDB98, 79C1BC35, 855C140A, 68A375B5, 34C637FD
97DA7754, E2193F3E, 24907FDF, 179CCA4F, EFCCC7B7, 34C637FD, B2D5EF34, 79C1BC35, 70502A15, 68A375B5
EFCCC7B7, D3AD6006, E2193F3E, 41FF7C92, 179CCA4F, 68A375B5, DB87209A, B2D5EF34, 06F0D5E7, 70502A15
179CCA4F, 6B8BFAB4, D3AD6006, 64FCFB88, 41FF7C92, 70502A15, 4DEC84F2, DB87209A, 57BCD2CB, 06F0D5E7
41FF7C92, 5052D6EF, 6B8BFAB4, B5801B4E, 64FCFB88, 06F0D5E7, D4F6A30D, 4DEC84F2, 1C826B6E, 57BCD2CB
64FCFB88, FF36EBC8, 5052D6EF, 2FEAD1AE, B5801B4E, 57BCD2CB, 0191C9F0, D4F6A30D, B213C937, 1C826B6E
B5801B4E, 5A010C53, FF36EBC8, 4B5BBD41, 2FEAD1AE, 1C826B6E, 20FBAB36, 0191C9F0, DA8C3753, B213C937
2FEAD1AE, 952BFB5D, 5A010C53, DBAF23FC, 4B5BBD41, B213C937, 7E796493, 20FBAB36, 4727C006, DA8C3753
4B5BBD41, FE05BEE3, 952BFB5D, 04314D68, DBAF23FC, DA8C3753, C9EABB3E, 7E796493, EEACD883, 4727C006
DBAF23FC, 2256AF69, FE05BEE3, AFED7654, 04314D68, 4727C006, B44977A5, C9EABB3E, E5924DF9, EEACD883
04314D68, 5285B0D3, 2256AF69, 16FB8FF8, AFED7654, EEACD883, 287580C6, B44977A5, AAECFB27, E5924DF9
AFED7654, 1DFB856C, 5285B0D3, 5ABDA489, 16FB8FF8, E5924DF9, 1E1DBD16, 287580C6, 25DE96D1, AAECFB27
16FB8FF8, 32974404, 1DFB856C, 16C34D4A, 5ABDA489, AAECFB27, FBEB21BA, 1E1DBD16, D60318A1, 25DE96D1
5ABDA489, 90AC71CE, 32974404, EE15B077, 16C34D4A, 25DE96D1, B74BF3E2, FBEB21BA, 76F45878, D60318A1
16C34D4A, 849CCC12, 90AC71CE, 5D1010CA, EE15B077, D60318A1, 755BEDDF, B74BF3E2, AC86EBEF, 76F45878
EE15B077, 340EBE92, 849CCC12, B1C73A42, 5D1010CA, 76F45878, 3CD099C6, 755BEDDF, 2FCF8ADD, AC86EBEF
5D1010CA, F531E5F5, 340EBE92, 73304A12, B1C73A42, AC86EBEF, A19BBAA2, 3CD099C6, 6FB77DD5, 2FCF8ADD
B1C73A42, 27528557, F531E5F5, 3AFA48D0, 73304A12, 2FCF8ADD, EFC554F1, A19BBAA2, 426718F3, 6FB77DD5
73304A12, E4AFA69F, 27528557, C797D7D4, 3AFA48D0, 6FB77DD5, F56F1485, EFC554F1, 6EEA8A86, 426718F3
3AFA48D0, E3462C93, E4AFA69F, 4A155C9D, C797D7D4, 426718F3, E0A1480A, F56F1485, 1553C7BF, 6EEA8A86
C797D7D4, 3CF5CD85, E3462C93, BE9A7F92, 4A155C9D, 6EEA8A86, 9F80007D, E0A1480A, BC5217D5, 1553C7BF
4A155C9D, B6C756F9, 3CF5CD85, 18B24F8D, BE9A7F92, 1553C7BF, 090898BE, 9F80007D, 85202B82, BC5217D5
BE9A7F92, CC2AB627, B6C756F9, D73614F3, 18B24F8D, BC5217D5, A0CD75A2, 090898BE, 0001F67E, 85202B82
18B24F8D, E5471921, CC2AB627, 1D5BE6DB, D73614F3, 85202B82, 95FE46E6, A0CD75A2, 2262F824, 0001F67E
D73614F3, E8FEFBC6, E5471921, AAD89F30, 1D5BE6DB, 0001F67E, 4B55D832, 95FE46E6, 35D68A83, 2262F824
1D5BE6DB, 788FFBE7, E8FEFBC6, 1C648795, AAD89F30, 2262F824, 681302D4, 4B55D832, F91B9A57, 35D68A83
AAD89F30, FA97F1BB, 788FFBE7, FBEF1BA3, 1C648795, 35D68A83, 860F8E32, 681302D4, 5760C92D, F91B9A57
1C648795, 2FE154B4, FA97F1BB, 3FEF9DE2, FBEF1BA3, F91B9A57, CA3DDAC0, 860F8E32, 4C0B51A0, 5760C92D
FBEF1BA3, D884695B, 2FE154B4, 5FC6EFEA, 3FEF9DE2, 5760C92D, 7E790793, CA3DDAC0, 3E38CA18, 4C0B51A0
3FEF9DE2, A09357E9, D884695B, 8552D0BF, 5FC6EFEA, 4C0B51A0, 4E0DF927, 7E790793, F76B0328, 3E38CA18
5FC6EFEA, 019B9791, A09357E9, 11A56F62, 8552D0BF, 3E38CA18, 311DFB90, 4E0DF927, E41E4DF9, F76B0328
8552D0BF, 70DB6FDF, 019B9791, 4D5FA682, 11A56F62, F76B0328, 24FA9DC7, 311DFB90, 37E49D38, E41E4DF9
11A56F62, 82F104B4, 70DB6FDF, 6E5E4406, 4D5FA682, E41E4DF9, CE45E142, 24FA9DC7, 77EE40C4, 37E49D38
4D5FA682, BFAB29F8, 82F104B4, 6DBF7DC3, 6E5E4406, 37E49D38, 9C4F267F, CE45E142, EA771C93, 77EE40C4
6E5E4406, 880198A9, BFAB29F8, C412D20B, 6DBF7DC3, 77EE40C4, 06880805, 9C4F267F, 17850B39, EA771C93
6DBF7DC3, 917C197C, 880198A9, ACA7E2FE, C412D20B, EA771C93, 7625BD09, 06880805, 3C99FE71, 17850B39

C412D20B, 03E7992A, 917C197C, 0662A620, ACA7E2FE, 17850B39, 8720C8E7, 7625BD09, 2020141A, 3C99FE71
ACA7E2FE, 824CEF7A, 03E7992A, F065F245, 0662A620, 3C99FE71, CBB7DA7A, 8720C8E7, 96F425D8, 2020141A
0662A620, AF16F218, 824CEF7A, 9E64A80F, F065F245, 2020141A, 88851068, CBB7DA7A, 83239E1C, 96F425D8
F065F245, EFC8943D, AF16F218, 33BDEA09, 9E64A80F, 96F425D8, C85C4EB8, 88851068, DF69EB2E, 83239E1C
9E64A80F, C80FF53B, EFC8943D, 5BC862BC, 33BDEA09, 83239E1C, 57BF18E2, C85C4EB8, 1441A222, DF69EB2E
33BDEA09, 28DF9E36, C80FF53B, 2250F7BF, 5BC862BC, DF69EB2E, 48932C1A, 57BF18E2, 713AE321, 1441A222
5BC862BC, 6E1D8950, 28DF9E36, 3FD4EF20, 2250F7BF, 1441A222, 15C7B0BD, 48932C1A, FC63895E, 713AE321
2250F7BF, 21EEE621, 6E1D8950, 7E78D8A3, 3FD4EF20, 713AE321, FCBC9E78, 15C7B0BD, 4CB06922, FC63895E
3FD4EF20, 561379BA, 21EEE621, 762541B8, 7E78D8A3, FC63895E, DD28EA60, FCBC9E78, 1EC2F457, 4CB06922
7E78D8A3, 4D0255C5, 561379BA, BB988487, 762541B8, 4CB06922, CF1BB810, DD28EA60, F279E3F2, 1EC2F457
762541B8, 966845EC, 4D0255C5, 4DE6E958, BB988487, 1EC2F457, 5D899D62, CF1BB810, A3A98374, F279E3F2
BB988487, D922DEB8, 966845EC, 09571534, 4DE6E958, F279E3F2, F1144141, 5D899D62, 6EE0433C, A3A98374
4DE6E958, B919B2A3, D922DEB8, A117B259, 09571534, A3A98374, 940BBA12, F1144141, 26758976, 6EE0433C
09571534, D3CF80F9, B919B2A3, 8B7AE364, A117B259, 6EE0433C, 33DDA9B5, 940BBA12, 510507C4, 26758976
A117B259, F548EA98, D3CF80F9, 66CA8EE4, 8B7AE364, 26758976, DCE0B562, 33DDA9B5, 2EE84A50, 510507C4
8B7AE364, A1D3372D, F548EA98, 3E03E74F, 66CA8EE4, 510507C4, C103FBE9, DCE0B562, 76A6D4CF, 2EE84A50
66CA8EE4, 6578D66C, A1D3372D, 23AA63D5, 3E03E74F, 2EE84A50, 832961D9, C103FBE9, 82D58B73, 76A6D4CF
3E03E74F, 57C29604, 6578D66C, 4CDCB687, 23AA63D5, 76A6D4CF, B183744E, 832961D9, 0FEFA704, 82D58B73
23AA63D5, 27F5E937, 57C29604, E359B195, 4CDCB687, 82D58B73, E710A112, B183744E, A587660C, 0FEFA704

The hash-code is the following 160-bit string.

12 A0 53 38 4A 9C 0C 88 E4 05 A0 6C 27 DC F4 9A DA 62 EB 2B

A.2.9 Example 9

In this example the data-string is the 1000000-byte string consisting of the ASCII-coded version of 'a' repeated 10^6 times.

The hash-code is the following 160-bit string.

52 78 32 43 C1 69 7B DB E1 6D 37 F9 7F 68 F0 83 25 DC 15 28

A.3 Dedicated Hash-Function 2

Hash-Function 1.

A.3.1 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 128-bit string.

CD F2 62 13 A1 50 DC 3E CB 61 0F 18 F6 B3 8B 46

A.3.2 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 128-bit string.

86 BE 7A FA 33 9D 0F C7 CF C7 85 E7 2F 57 8D 33

A.3.3 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.

80636261	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000018	00000000

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X'_0, X'_1, X'_2, X'_3.

67452301, EFCDAB89, 98BADCFE, 10325476, 67452301, EFCDAB89, 98BADCFE, 10325476
10325476, 6D431A77, EFCDAB89, 98BADCFE, 10325476, 70376F40, EFCDAB89, 98BADCFE
98BADCFE, B05D8A99, 6D431A77, EFCDAB89, 98BADCFE, 989F6BB0, 70376F40, EFCDAB89
EFCDAB89, 0C32E5C7, B05D8A99, 6D431A77, EFCDAB89, 39B14904, 989F6BB0, 70376F40
6D431A77, A20B2C0F, 0C32E5C7, B05D8A99, 70376F40, 671C03CC, 39B14904, 989F6BB0
B05D8A99, 74EBB911, A20B2C0F, 0C32E5C7, 989F6BB0, BFD55C42, 671C03CC, 39B14904
0C32E5C7, 2FFB728B, 74EBB911, A20B2C0F, 39B14904, A12F346F, BFD55C42, 671C03CC
A20B2C0F, A766AE02, 2FFB728B, 74EBB911, 671C03CC, 989C2210, A12F346F, BFD55C42
74EBB911, 03234F3D, A766AE02, 2FFB728B, BFD55C42, 0F95FBEA, 989C2210, A12F346F
2FFB728B, 52662805, 03234F3D, A766AE02, A12F346F, 068D5115, 0F95FBEA, 989C2210
A766AE02, E778A4C3, 52662805, 03234F3D, 989C2210, AFCD27FC, 068D5115, 0F95FBEA
03234F3D, 1C7F5769, E778A4C3, 52662805, 0F95FBEA, CBD1F3F8, AFCD27FC, 068D5115
52662805, 95765642, 1C7F5769, E778A4C3, 068D5115, CFFE405F, CBD1F3F8, AFCD27FC
E778A4C3, 35F37B70, 95765642, 1C7F5769, AFCD27FC, 2B55C9C3, CFFE405F, CBD1F3F8
1C7F5769, 398F8F52, 35F37B70, 95765642, CBD1F3F8, DD6A43FB, 2B55C9C3, CFFE405F
95765642, 13F3C36B, 398F8F52, 35F37B70, CFFE405F, 049B909E, DD6A43FB, 2B55C9C3
35F37B70, 058D8BB5, 13F3C36B, 398F8F52, 2B55C9C3, 3713BFFD, 049B909E, DD6A43FB
398F8F52, FCBE3664, 058D8BB5, 13F3C36B, DD6A43FB, 82ADDB53, 3713BFFD, 049B909E
13F3C36B, F7F306A6, FCBE3664, 058D8BB5, 049B909E, CC1D8105, 82ADDB53, 3713BFFD
058D8BB5, 34CC3963, F7F306A6, FCBE3664, 3713BFFD, BE09159A, CC1D8105, 82ADDB53
FCBE3664, 416E8BA0, 34CC3963, F7F306A6, 82ADDB53, 541AE568, BE09159A, CC1D8105
F7F306A6, EDE91870, 416E8BA0, 34CC3963, CC1D8105, 27D40F94, 541AE568, BE09159A
34CC3963, C352C547, EDE91870, 416E8BA0, BE09159A, 675C363A, 27D40F94, 541AE568
416E8BA0, 5D5EEE28, C352C547, EDE91870, 541AE568, 77F3A38B, 675C363A, 27D40F94
EDE91870, 6CC4BEF2, 5D5EEE28, C352C547, 27D40F94, 84D73C44, 77F3A38B, 675C363A
C352C547, E140970B, 6CC4BEF2, 5D5EEE28, 675C363A, D2958F37, 84D73C44, 77F3A38B
5D5EEE28, 79F631A9, E140970B, 6CC4BEF2, 77F3A38B, FC39C927, D2958F37, 84D73C44
6CC4BEF2, 038E0E91, 79F631A9, E140970B, 84D73C44, E3A5A4DE, FC39C927, D2958F37
E140970B, 1B942D52, 038E0E91, 79F631A9, D2958F37, 4BA3A889, E3A5A4DE, FC39C927
79F631A9, 496AECFD, 1B942D52, 038E0E91, FC39C927, A964BA74, 4BA3A889, E3A5A4DE
038E0E91, FE6CD56F, 496AECFD, 1B942D52, E3A5A4DE, 7AF9DBB0, A964BA74, 4BA3A889
1B942D52, 2E94F501, FE6CD56F, 496AECFD, 4BA3A889, 7DA68EA9, 7AF9DBB0, A964BA74
496AECFD, 584E8E58, 2E94F501, FE6CD56F, A964BA74, 9C7247E5, 7DA68EA9, 7AF9DBB0
FE6CD56F, 41A17EFA, 584E8E58, 2E94F501, 7AF9DBB0, 0130312B, 9C7247E5, 7DA68EA9
2E94F501, 8981C6CD, 41A17EFA, 584E8E58, 7DA68EA9, 90552232, 0130312B, 9C7247E5
584E8E58, 400A93E1, 8981C6CD, 41A17EFA, 9C7247E5, 99C1FBA4, 90552232, 0130312B
41A17EFA, 841F817F, 400A93E1, 8981C6CD, 0130312B, 9D481CD2, 99C1FBA4, 90552232
8981C6CD, 659379BE, 841F817F, 400A93E1, 90552232, F5AABE07, 9D481CD2, 99C1FBA4
400A93E1, AB3D9A70, 659379BE, 841F817F, 99C1FBA4, C3AFB7E6, F5AABE07, 9D481CD2
841F817F, D3D21DC8, AB3D9A70, 659379BE, 9D481CD2, 473E2B79, C3AFB7E6, F5AABE07
659379BE, 38C8D29D, D3D21DC8, AB3D9A70, F5AABE07, C4CAFF99, 473E2B79, C3AFB7E6
AB3D9A70, 738B9B0F, 38C8D29D, D3D21DC8, C3AFB7E6, A2879AA4, C4CAFF99, 473E2B79
D3D21DC8, 8528B83E, 738B9B0F, 38C8D29D, 473E2B79, 56565EDB, A2879AA4, C4CAFF99
38C8D29D, 7345AF18, 8528B83E, 738B9B0F, C4CAFF99, E7A4BD86, 56565EDB, A2879AA4
738B9B0F, FFCCC52B, 7345AF18, 8528B83E, A2879AA4, 974B9E10, E7A4BD86, 56565EDB
8528B83E, A77E902B, FFCCC52B, 7345AF18, 56565EDB, 96CC5AE1, 974B9E10, E7A4BD86

```
7345AF18, CB9C6C83, A77E902B, FFCCC52B, E7A4BD86, 57E6A772, 96CC5AE1, 974B9E10
FFCCC52B, 38A2DA83, CB9C6C83, A77E902B, 974B9E10, F10B6CF5, 57E6A772, 96CC5AE1
A77E902B, 487F9401, 38A2DA83, CB9C6C83, 96CC5AE1, 90426E6B, F10B6CF5, 57E6A772
CB9C6C83, C7184576, 487F9401, 38A2DA83, 57E6A772, 0066E6BE, 90426E6B, F10B6CF5
38A2DA83, 56D619B1, C7184576, 487F9401, F10B6CF5, 22D17257, 0066E6BE, 90426E6B
487F9401, 3A35A3C5, 56D619B1, C7184576, 90426E6B, 016777A4, 22D17257, 0066E6BE
C7184576, B5517538, 3A35A3C5, 56D619B1, 0066E6BE, 9A8DC5A0, 016777A4, 22D17257
56D619B1, 4609C4C2, B5517538, 3A35A3C5, 22D17257, A9C46E68, 9A8DC5A0, 016777A4
3A35A3C5, D5C2B699, 4609C4C2, B5517538, 016777A4, 13B0D540, A9C46E68, 9A8DC5A0
B5517538, 342AF741, D5C2B699, 4609C4C2, 9A8DC5A0, 983D8B08, 13B0D540, A9C46E68
4609C4C2, 38286DDA, 342AF741, D5C2B699, A9C46E68, 96084F4E, 983D8B08, 13B0D540
D5C2B699, 9BCEEC0A, 38286DDA, 342AF741, 13B0D540, D25FDBB1, 96084F4E, 983D8B08
342AF741, 5803DF3A, 9BCEEC0A, 38286DDA, 983D8B08, 35EA6FE0, D25FDBB1, 96084F4E
38286DDA, E1B026EB, 5803DF3A, 9BCEEC0A, 96084F4E, B862709F, 35EA6FE0, D25FDBB1
9BCEEC0A, 31587C22, E1B026EB, 5803DF3A, D25FDBB1, C02839EB, B862709F, 35EA6FE0
5803DF3A, 9B25E1DC, 31587C22, E1B026EB, 35EA6FE0, 00245200, C02839EB, B862709F
E1B026EB, 2205379E, 9B25E1DC, 31587C22, B862709F, CB116A95, 00245200, C02839EB
31587C22, 5E3334A3, 2205379E, 9B25E1DC, C02839EB, B90EE1BF, CB116A95, 00245200
9B25E1DC, 56F80FA9, 5E3334A3, 2205379E, 00245200, 64132D32, B90EE1BF, CB116A95
```

The hash-code is the following 128-bit string.

```
C1 4A 12 19 9C 66 E4 BA 84 63 6B 0F 69 14 4C 77
```

A.3.4 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 128-bit string.

```
9E 32 7B 3D 6E 52 30 62 AF C1 13 2D 7D F9 D1 B8
```

A.3.5 Example 5

In this example the data-string is the 26-byte string consisting of the ASCII-coded version of

'abcdefghijklmnopqrstuvwxyz'

The hash-code is the following 128-bit string.

```
FD 2A A6 07 F7 1D C8 F5 10 71 49 22 B3 71 83 4E
```

A.3.6 Example 6

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 128-bit string.

D1 E9 59 EB 17 9C 91 1F AE A4 62 4C 60 C5 C7 02

A.3.7 Example 7

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 128-bit string.

3F 45 EF 19 47 32 C2 DB B2 C4 A2 C7 69 79 5F A3

A.3.8 Example 8

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

'abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq'

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

64636261	65646362	66656463	67666564	68676665	69686766	6A696867	6B6A6968
6C6B6A69	6D6C6B6A	6E6D6C6B	6F6E6D6C	706F6E6D	71706F6E	00000080	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	000001C0	00000000

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X'_0, X'_1, X'_2, X'_3, obtained during the processing of the first block.

67452301, EFCDAB89, 98BADCFE, 10325476, 67452301, EFCDAB89, 98BADCFE, 10325476
10325476, 6D431997, EFCDAB89, 98BADCFE, 10325476, D89ED5A9, EFCDAB89, 98BADCFE
98BADCFE, C9AE23F2, 6D431997, EFCDAB89, 98BADCFE, 69B10AC1, D89ED5A9, EFCDAB89
EFCDAB89, 69A6A520, C9AE23F2, 6D431997, EFCDAB89, B661DB9C, 69B10AC1, D89ED5A9
6D431997, FB032247, 69A6A520, C9AE23F2, D89ED5A9, ABACC2AF, B661DB9C, 69B10AC1
C9AE23F2, 16C49226, FB032247, 69A6A520, 69B10AC1, D412CAD1, ABACC2AF, B661DB9C
69A6A520, 77A099B7, 16C49226, FB032247, B661DB9C, E2DEDF22, D412CAD1, ABACC2AF
FB032247, 3B9BAEB7, 77A099B7, 16C49226, ABACC2AF, CFB03688, E2DEDF22, D412CAD1
16C49226, DA61AB82, 3B9BAEB7, 77A099B7, D412CAD1, 72599389, CFB03688, E2DEDF22
77A099B7, 54C888CC, DA61AB82, 3B9BAEB7, E2DEDF22, CF3CD682, 72599389, CFB03688
3B9BAEB7, F2635347, 54C888CC, DA61AB82, CFB03688, B235784E, CF3CD682, 72599389
DA61AB82, E2CAC9B4, F2635347, 54C888CC, 72599389, 881678DF, B235784E, CF3CD682
54C888CC, 9596C718, E2CAC9B4, F2635347, CF3CD682, E815373B, 881678DF, B235784E
F2635347, 9DD54912, 9596C718, E2CAC9B4, B235784E, BD994B56, E815373B, 881678DF
E2CAC9B4, 2E8539A7, 9DD54912, 9596C718, 881678DF, B0055655, BD994B56, E815373B
9596C718, 2303C213, 2E8539A7, 9DD54912, E815373B, CC87EF5A, B0055655, BD994B56
9DD54912, EA79BE25, 2303C213, 2E8539A7, BD994B56, 6B24384D, CC87EF5A, B0055655
2E8539A7, 23D7CB45, EA79BE25, 2303C213, B0055655, 93E7329F, 6B24384D, CC87EF5A
2303C213, F028EF04, 23D7CB45, EA79BE25, CC87EF5A, 35B95AE7, 93E7329F, 6B24384D
EA79BE25, 48863F19, F028EF04, 23D7CB45, 6B24384D, 06C6536D, 35B95AE7, 93E7329F
23D7CB45, 514C81B6, 48863F19, F028EF04, 93E7329F, FF1C5DC7, 06C6536D, 35B95AE7
F028EF04, 6102CE67, 514C81B6, 48863F19, 35B95AE7, D0D541F1, FF1C5DC7, 06C6536D
48863F19, 330485FD, 6102CE67, 514C81B6, 06C6536D, A94C0DD9, D0D541F1, FF1C5DC7

514C81B6, 289E8C82, 330485FD, 6102CE67, FF1C5DC7, DEDC1E39, A94C0DD9, D0D541F1
6102CE67, 13CC3A1D, 289E8C82, 330485FD, D0D541F1, 12D926C0, DEDC1E39, A94C0DD9
330485FD, 40A226A6, 13CC3A1D, 289E8C82, A94C0DD9, ED7EDA63, 12D926C0, DEDC1E39
289E8C82, 70BFB1A8, 40A226A6, 13CC3A1D, DEDC1E39, 9E52219C, ED7EDA63, 12D926C0
13CC3A1D, CE1D1A37, 70BFB1A8, 40A226A6, 12D926C0, F5D22339, 9E52219C, ED7EDA63
40A226A6, EC9F7830, CE1D1A37, 70BFB1A8, ED7EDA63, 0BC5B4FC, F5D22339, 9E52219C
70BFB1A8, 3CF2D6EE, EC9F7830, CE1D1A37, 9E52219C, FCFBD391, 0BC5B4FC, F5D22339
CE1D1A37, F0C1F95C, 3CF2D6EE, EC9F7830, F5D22339, 2B6A389B, FCFBD391, 0BC5B4FC
EC9F7830, 9A351A9D, F0C1F95C, 3CF2D6EE, 0BC5B4FC, FBF85B05, 2B6A389B, FCFBD391
3CF2D6EE, 138B0685, 9A351A9D, F0C1F95C, FCFBD391, F7BBBE8B, FBF85B05, 2B6A389B
F0C1F95C, EA3574D1, 138B0685, 9A351A9D, 2B6A389B, C8592ACC, F7BBBE8B, FBF85B05
9A351A9D, 4719C849, EA3574D1, 138B0685, FBF85B05, FE2D3EFA, C8592ACC, F7BBBE8B
138B0685, 57F52A13, 4719C849, EA3574D1, F7BBBE8B, 5411CC34, FE2D3EFA, C8592ACC
EA3574D1, 4751F880, 57F52A13, 4719C849, C8592ACC, DC8ED546, 5411CC34, FE2D3EFA
4719C849, 80605BAF, 4751F880, 57F52A13, FE2D3EFA, 55C1E317, DC8ED546, 5411CC34
57F52A13, 1E53AD4A, 80605BAF, 4751F880, 5411CC34, 0B92E4F0, 55C1E317, DC8ED546
4751F880, 1ABEED79, 1E53AD4A, 80605BAF, DC8ED546, 5E192900, 0B92E4F0, 55C1E317
80605BAF, 75EACBB7, 1ABEED79, 1E53AD4A, 55C1E317, 186EB0CF, 5E192900, 0B92E4F0
1E53AD4A, 08AC1056, 75EACBB7, 1ABEED79, 0B92E4F0, 8F3A64E3, 186EB0CF, 5E192900
1ABEED79, 9BDB7A88, 08AC1056, 75EACBB7, 5E192900, 3701E7B3, 8F3A64E3, 186EB0CF
75EACBB7, ADF32F05, 9BDB7A88, 08AC1056, 186EB0CF, 6CE969E9, 3701E7B3, 8F3A64E3
08AC1056, 2277B80D, ADF32F05, 9BDB7A88, 8F3A64E3, EE7224D5, 6CE969E9, 3701E7B3
9BDB7A88, 535DBB9A, 2277B80D, ADF32F05, 3701E7B3, 3E849D0F, EE7224D5, 6CE969E9
ADF32F05, 2A494EC5, 535DBB9A, 2277B80D, 6CE969E9, DDBD8EE7, 3E849D0F, EE7224D5
2277B80D, 693C7A09, 2A494EC5, 535DBB9A, EE7224D5, C3DDAC40, DDBD8EE7, 3E849D0F
535DBB9A, 148A5796, 693C7A09, 2A494EC5, 3E849D0F, 5E0E10B9, C3DDAC40, DDBD8EE7
2A494EC5, D2932448, 148A5796, 693C7A09, DDBD8EE7, 1CCB75AF, 5E0E10B9, C3DDAC40
693C7A09, 39CA97B6, D2932448, 148A5796, C3DDAC40, 27F81499, 1CCB75AF, 5E0E10B9
148A5796, 770BCE98, 39CA97B6, D2932448, 5E0E10B9, 82843491, 27F81499, 1CCB75AF
D2932448, 8C4DC6AF, 770BCE98, 39CA97B6, 1CCB75AF, 4E4E13E9, 82843491, 27F81499
39CA97B6, 048CC517, 8C4DC6AF, 770BCE98, 27F81499, 03BD1BD9, 4E4E13E9, 82843491
770BCE98, 419960CF, 048CC517, 8C4DC6AF, 82843491, 6FA999B7, 03BD1BD9, 4E4E13E9
8C4DC6AF, 407700EE, 419960CF, 048CC517, 4E4E13E9, 37B18629, 6FA999B7, 03BD1BD9
048CC517, E60ABEC4, 407700EE, 419960CF, 03BD1BD9, 9EA44395, 37B18629, 6FA999B7
419960CF, 0E248A8B, E60ABEC4, 407700EE, 6FA999B7, F877D28C, 9EA44395, 37B18629
407700EE, 10667792, 0E248A8B, E60ABEC4, 37B18629, F63EA862, F877D28C, 9EA44395
E60ABEC4, 646BB7A8, 10667792, 0E248A8B, 9EA44395, 424072F0, F63EA862, F877D28C
0E248A8B, 625CCE22, 646BB7A8, 10667792, F877D28C, 3B7642B8, 424072F0, F63EA862
10667792, 8E0E1101, 625CCE22, 646BB7A8, F63EA862, CD620F4E, 3B7642B8, 424072F0
646BB7A8, C23D3583, 8E0E1101, 625CCE22, 424072F0, BFAA1A02, CD620F4E, 3B7642B8
625CCE22, 81DE3DC5, C23D3583, 8E0E1101, 3B7642B8, 1BA7FD36, BFAA1A02, CD620F4E
8E0E1101, D24E4181, 81DE3DC5, C23D3583, CD620F4E, E62BB2A4, 1BA7FD36, BFAA1A02

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X'_0, X'_1, X'_2, X'_3, obtained during the processing of the second block.

31560350, 285A21CF, 846C181B, 553B61B8, 31560350, 285A21CF, 846C181B, 553B61B8
553B61B8, 1ADDE153, 285A21CF, 846C181B, 553B61B8, 56C8C102, 285A21CF, 846C181B
846C181B, CE8FC309, 1ADDE153, 285A21CF, 846C181B, 702249A4, 56C8C102, 285A21CF
285A21CF, 0DD8403A, CE8FC309, 1ADDE153, 285A21CF, 22CB0A97, 702249A4, 56C8C102
1ADDE153, 4842F01E, 0DD8403A, CE8FC309, 56C8C102, 35B2DCDF, 22CB0A97, 702249A4
CE8FC309, BE6A9014, 4842F01E, 0DD8403A, 702249A4, D2EFFB4A, 35B2DCDF, 22CB0A97
0DD8403A, 7FE339CA, BE6A9014, 4842F01E, 22CB0A97, 59EA6C60, D2EFFB4A, 35B2DCDF
4842F01E, D1CCFD4B, 7FE339CA, BE6A9014, 35B2DCDF, 82DEA3AE, 59EA6C60, D2EFFB4A
BE6A9014, 108966B1, D1CCFD4B, 7FE339CA, D2EFFB4A, 4481FDE2, 82DEA3AE, 59EA6C60
7FE339CA, 899223E8, 108966B1, D1CCFD4B, 59EA6C60, 13BB8F73, 4481FDE2, 82DEA3AE
D1CCFD4B, 5E3B9917, 899223E8, 108966B1, 82DEA3AE, 946BD478, 13BB8F73, 4481FDE2
108966B1, 7666663B, 5E3B9917, 899223E8, 4481FDE2, BD0605EA, 946BD478, 13BB8F73
899223E8, A1BAD92C, 7666663B, 5E3B9917, 13BB8F73, 36F99153, BD0605EA, 946BD478
5E3B9917, DE527A04, A1BAD92C, 7666663B, 946BD478, EB4AE872, 36F99153, BD0605EA

7666663B, E52F1533, DE527A04, A1BAD92C, BD0605EA, 7C346442, EB4AE872, 36F99153
A1BAD92C, 5C3C2C22, E52F1533, DE527A04, 36F99153, AFA320AD, 7C346442, EB4AE872
DE527A04, FC1C4108, 5C3C2C22, E52F1533, EB4AE872, B4905651, AFA320AD, 7C346442
E52F1533, 0A03E84B, FC1C4108, 5C3C2C22, 7C346442, 02E94FA1, B4905651, AFA320AD
5C3C2C22, FB74BD26, 0A03E84B, FC1C4108, AFA320AD, E08D1799, 02E94FA1, B4905651
FC1C4108, C78DC5C4, FB74BD26, 0A03E84B, B4905651, 69AFAA80, E08D1799, 02E94FA1
0A03E84B, ACF60434, C78DC5C4, FB74BD26, 02E94FA1, FA665E46, 69AFAA80, E08D1799
FB74BD26, 58F751E0, ACF60434, C78DC5C4, E08D1799, 269AB7E3, FA665E46, 69AFAA80
C78DC5C4, EB75C7CB, 58F751E0, ACF60434, 69AFAA80, 0F06388B, 269AB7E3, FA665E46
ACF60434, 83C0A8B7, EB75C7CB, 58F751E0, FA665E46, FD44FBD5, 0F06388B, 269AB7E3
58F751E0, 27C87178, 83C0A8B7, EB75C7CB, 269AB7E3, DBBC0190, FD44FBD5, 0F06388B
EB75C7CB, B7B9163F, 27C87178, 83C0A8B7, 0F06388B, D0E3FC2B, DBBC0190, FD44FBD5
83C0A8B7, 0FA1C6DC, B7B9163F, 27C87178, FD44FBD5, 7D87B4BA, D0E3FC2B, DBBC0190
27C87178, 2CC60316, 0FA1C6DC, B7B9163F, DBBC0190, 68367FDB, 7D87B4BA, D0E3FC2B
B7B9163F, 08029C44, 2CC60316, 0FA1C6DC, D0E3FC2B, 53AB5439, 68367FDB, 7D87B4BA
0FA1C6DC, F693A10E, 08029C44, 2CC60316, 7D87B4BA, E78B75B5, 53AB5439, 68367FDB
2CC60316, 356224B9, F693A10E, 08029C44, 68367FDB, 830530DF, E78B75B5, 53AB5439
08029C44, 669F7869, 356224B9, F693A10E, 53AB5439, 67FCB1AC, 830530DF, E78B75B5
F693A10E, 7B70C168, 669F7869, 356224B9, E78B75B5, 757BB243, 67FCB1AC, 830530DF
356224B9, 037FB19C, 7B70C168, 669F7869, 830530DF, F0CA8878, 757BB243, 67FCB1AC
669F7869, 9B0A10B3, 037FB19C, 7B70C168, 67FCB1AC, FA10CB33, F0CA8878, 757BB243
7B70C168, 9D015956, 9B0A10B3, 037FB19C, 757BB243, 5487E56C, FA10CB33, F0CA8878
037FB19C, 6A7DE5F4, 9D015956, 9B0A10B3, F0CA8878, A5D33699, 5487E56C, FA10CB33
9B0A10B3, E522D913, 6A7DE5F4, 9D015956, FA10CB33, BEB495BC, A5D33699, 5487E56C
9D015956, 0EFD42E5, E522D913, 6A7DE5F4, 5487E56C, 05202F93, BEB495BC, A5D33699
6A7DE5F4, 7902100B, 0EFD42E5, E522D913, A5D33699, BACE7DD9, 05202F93, BEB495BC
E522D913, 1ACEFABC, 7902100B, 0EFD42E5, BEB495BC, 08D045DD, BACE7DD9, 05202F93
0EFD42E5, E07378FF, 1ACEFABC, 7902100B, 05202F93, 5448A3A0, 08D045DD, BACE7DD9
7902100B, 489C7A1A, E07378FF, 1ACEFABC, BACE7DD9, D98BE3AA, 5448A3A0, 08D045DD
1ACEFABC, C02A45A5, 489C7A1A, E07378FF, 08D045DD, 12EC982F, D98BE3AA, 5448A3A0
E07378FF, 3068DDE8, C02A45A5, 489C7A1A, 5448A3A0, 4A1EB2B2, 12EC982F, D98BE3AA
489C7A1A, D5DD5018, 3068DDE8, C02A45A5, D98BE3AA, D677AAA8, 4A1EB2B2, 12EC982F
C02A45A5, B9D75D76, D5DD5018, 3068DDE8, 12EC982F, 5AA89133, D677AAA8, 4A1EB2B2
3068DDE8, 51A9B2DD, B9D75D76, D5DD5018, 4A1EB2B2, 49BCE169, 5AA89133, D677AAA8
D5DD5018, 36F589C4, 51A9B2DD, B9D75D76, D677AAA8, CF4FA8D2, 49BCE169, 5AA89133
B9D75D76, B5C60EAF, 36F589C4, 51A9B2DD, 5AA89133, C1985969, CF4FA8D2, 49BCE169
51A9B2DD, 725DF80C, B5C60EAF, 36F589C4, 49BCE169, 427440B4, C1985969, CF4FA8D2
36F589C4, 3F7A2507, 725DF80C, B5C60EAF, CF4FA8D2, 60927896, 427440B4, C1985969
B5C60EAF, 9D539EB6, 3F7A2507, 725DF80C, C1985969, 7050ED96, 60927896, 427440B4
725DF80C, 5A249895, 9D539EB6, 3F7A2507, 427440B4, CBC74513, 7050ED96, 60927896
3F7A2507, A7CECDCD, 5A249895, 9D539EB6, 60927896, 8431C75E, CBC74513, 7050ED96
9D539EB6, F8DCD12B, A7CECDCD, 5A249895, 7050ED96, 0E3A1C68, 8431C75E, CBC74513
5A249895, 3E30DB2A, F8DCD12B, A7CECDCD, CBC74513, 62EEEC87, 0E3A1C68, 8431C75E
A7CECDCD, A25D36CE, 3E30DB2A, F8DCD12B, 8431C75E, 2B1F312D, 62EEEC87, 0E3A1C68
F8DCD12B, A92CF759, A25D36CE, 3E30DB2A, 0E3A1C68, FB124197, 2B1F312D, 62EEEC87
3E30DB2A, 0CD0BA66, A92CF759, A25D36CE, 62EEEC87, DB8A5C11, FB124197, 2B1F312D
A25D36CE, AF62D775, 0CD0BA66, A92CF759, 2B1F312D, EC3264DC, DB8A5C11, FB124197
A92CF759, 69D4E1DF, AF62D775, 0CD0BA66, FB124197, 9AA87F7C, EC3264DC, DB8A5C11
0CD0BA66, 0EE66339, 69D4E1DF, AF62D775, DB8A5C11, 04512915, 9AA87F7C, EC3264DC
AF62D775, 5C5B5FBD, 0EE66339, 69D4E1DF, EC3264DC, C763272A, 04512915, 9AA87F7C
69D4E1DF, 0D80E8CF, 5C5B5FBD, 0EE66339, 9AA87F7C, CCD7DF45, C763272A, 04512915

The hash-code is the following 128-bit string.

A1 AA 06 89 D0 FA FA 2D DC 22 E8 8B 49 13 3A 06

A.3.9 Example 9

In this example the data-string is the 1000000-byte string consisting of the ASCII-coded version of 'a' repeated 10^6 times.

The hash-code is the following 128-bit string.

```
4A 7F 57 23 F9 54 EB A1 21 6C 9D 8F 63 20 43 1F
```

A.4 Dedicated Hash-Function 3

Hash-Function 3.

A.4.1 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 160-bit string.

```
DA 39 A3 EE 5E 6B 4B 0D 32 55 BF EF 95 60 18 90 AF D8 07 09
```

A.4.2 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 160-bit string.

```
86 F7 E4 37 FA A5 A7 FC E1 5D 1D DC B9 EA EA EA 37 76 67 B8
```

A.4.3 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 X_0, X_1, X_2, X_3, X_4.

```
0116FC33, 67452301, 7BF36AE2, 98BADCFE, 10325476
8990536D, 0116FC33, 59D148C0, 7BF36AE2, 98BADCFE
A1390F08, 8990536D, C045BF0C, 59D148C0, 7BF36AE2
CDD8E11B, A1390F08, 626414DB, C045BF0C, 59D148C0
CFD499DE, CDD8E11B, 284E43C2, 626414DB, C045BF0C
3FC7CA40, CFD499DE, F3763846, 284E43C2, 626414DB
993E30C1, 3FC7CA40, B3F52677, F3763846, 284E43C2
9E8C07D4, 993E30C1, 0FF1F290, B3F52677, F3763846
4B6AE328, 9E8C07D4, 664F8C30, 0FF1F290, B3F52677
8351F929, 4B6AE328, 27A301F5, 664F8C30, 0FF1F290
```

FBDA9E89, 8351F929, 12DAB8CA, 27A301F5, 664F8C30
63188FE4, FBDA9E89, 60D47E4A, 12DAB8CA, 27A301F5
4607B664, 63188FE4, 7EF6A7A2, 60D47E4A, 12DAB8CA
9128F695, 4607B664, 18C623F9, 7EF6A7A2, 60D47E4A
196BEE77, 9128F695, 1181ED99, 18C623F9, 7EF6A7A2
20BDD62F, 196BEE77, 644A3DA5, 1181ED99, 18C623F9
4E925823, 20BDD62F, C65AFB9D, 644A3DA5, 1181ED99
82AA6728, 4E925823, C82F758B, C65AFB9D, 644A3DA5
DC64901D, 82AA6728, D3A49608, C82F758B, C65AFB9D
FD9E1D7D, DC64901D, 20AA99CA, D3A49608, C82F758B
1A37B0CA, FD9E1D7D, 77192407, 20AA99CA, D3A49608
33A23BFC, 1A37B0CA, 7F67875F, 77192407, 20AA99CA
21283486, 33A23BFC, 868DEC32, 7F67875F, 77192407
D541F12D, 21283486, 0CE88EFF, 868DEC32, 7F67875F
C7567DC6, D541F12D, 884A0D21, 0CE88EFF, 868DEC32
48413BA4, C7567DC6, 75507C4B, 884A0D21, 0CE88EFF
BE35FBD5, 48413BA4, B1D59F71, 75507C4B, 884A0D21
4AA84D97, BE35FBD5, 12104EE9, B1D59F71, 75507C4B
8370B52E, 4AA84D97, 6F8D7EF5, 12104EE9, B1D59F71
C5FBAF5D, 8370B52E, D2AA1365, 6F8D7EF5, 12104EE9
1267B407, C5FBAF5D, A0DC2D4B, D2AA1365, 6F8D7EF5
3B845D33, 1267B407, 717EEBD7, A0DC2D4B, D2AA1365
046FAA0A, 3B845D33, C499ED01, 717EEBD7, A0DC2D4B
2C0EBC11, 046FAA0A, CEE1174C, C499ED01, 717EEBD7
21796AD4, 2C0EBC11, 811BEA82, CEE1174C, C499ED01
DCBBB0CB, 21796AD4, 4B03AF04, 811BEA82, CEE1174C
0F511FD8, DCBBB0CB, 085E5AB5, 4B03AF04, 811BEA82
DC63973F, 0F511FD8, F72EEC32, 085E5AB5, 4B03AF04
4C986405, DC63973F, 03D447F6, F72EEC32, 085E5AB5
32DE1CBA, 4C986405, F718E5CF, 03D447F6, F72EEC32
FC87DEDF, 32DE1CBA, 53261901, F718E5CF, 03D447F6
970A0D5C, FC87DEDF, 8CB7872E, 53261901, F718E5CF
7F193DC5, 970A0D5C, FF21F7B7, 8CB7872E, 53261901
EE1B1AAF, 7F193DC5, 25C28357, FF21F7B7, 8CB7872E
40F28E09, EE1B1AAF, 5FC64F71, 25C28357, FF21F7B7
1C51E1F2, 40F28E09, FB86C6AB, 5FC64F71, 25C28357
A01B846C, 1C51E1F2, 503CA382, FB86C6AB, 5FC64F71
BEAD02CA, A01B846C, 8714787C, 503CA382, FB86C6AB
BAF39337, BEAD02CA, 2806E11B, 8714787C, 503CA382
120731C5, BAF39337, AFAB40B2, 2806E11B, 8714787C
641DB2CE, 120731C5, EEBCE4CD, AFAB40B2, 2806E11B
3847AD66, 641DB2CE, 4481CC71, EEBCE4CD, AFAB40B2
E490436D, 3847AD66, 99076CB3, 4481CC71, EEBCE4CD
27E9F1D8, E490436D, 8E11EB59, 99076CB3, 4481CC71
7B71F76D, 27E9F1D8, 792410DB, 8E11EB59, 99076CB3
5E6456AF, 7B71F76D, 09FA7C76, 792410DB, 8E11EB59
C846093F, 5E6456AF, 5EDC7DDB, 09FA7C76, 792410DB
D262FF50, C846093F, D79915AB, 5EDC7DDB, 09FA7C76
09D785FD, D262FF50, F211824F, D79915AB, 5EDC7DDB
3F52DE5A, 09D785FD, 3498BFD4, F211824F, D79915AB
D756C147, 3F52DE5A, 4275E17F, 3498BFD4, F211824F
548C9CB2, D756C147, 8FD4B796, 4275E17F, 3498BFD4
B66C020B, 548C9CB2, F5D5B051, 8FD4B796, 4275E17F
6B61C9E1, B66C020B, 9523272C, F5D5B051, 8FD4B796
19DFA7AC, 6B61C9E1, ED9B0082, 9523272C, F5D5B051
101655F9, 19DFA7AC, 5AD87278, ED9B0082, 9523272C
0C3DF2B4, 101655F9, 0677E9EB, 5AD87278, ED9B0082
78DD4D2B, 0C3DF2B4, 4405957E, 0677E9EB, 5AD87278
497093C0, 78DD4D2B, 030F7CAD, 4405957E, 0677E9EB
3F2588C2, 497093C0, DE37534A, 030F7CAD, 4405957E
C199F8C7, 3F2588C2, 125C24F0, DE37534A, 030F7CAD

```
39859DE7, C199F8C7, 8FC96230, 125C24F0, DE37534A
EDB42DE4, 39859DE7, F0667E31, 8FC96230, 125C24F0
11793F6F, EDB42DE4, CE616779, F0667E31, 8FC96230
5EE76897, 11793F6F, 3B6D0B79, CE616779, F0667E31
63F7DAB7, 5EE76897, C45E4FDB, 3B6D0B79, CE616779
A079B7D9, 63F7DAB7, D7B9DA25, C45E4FDB, 3B6D0B79
860D21CC, A079B7D9, D8FDF6AD, D7B9DA25, C45E4FDB
5738D5E1, 860D21CC, 681E6DF6, D8FDF6AD, D7B9DA25
42541B35, 5738D5E1, 21834873, 681E6DF6, D8FDF6AD
```

The hash-code is the following 160-bit string.

```
A9 99 3E 36 47 06 81 6A BA 3E 25 71 78 50 C2 6C 9C D0 D8 9D
```

A.4.4 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 160-bit string.

```
C1 22 52 CE DA 8B E8 99 4D 5F A0 29 0A 47 23 1C 1D 16 AA E3
```

A.4.5 Example 5

In this example the data-string is the 26-byte string consisting of the ASCII-coded version of

'abcdefghijklmnopqrstuvwxyz'

The hash-code is the following 160-bit string.

```
32 D1 0C 7B 8C F9 65 70 CA 04 CE 37 F2 A1 9D 84 24 0D 3A 89
```

A.4.6 Example 6

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 160-bit string.

```
76 1C 45 7B F7 3B 14 D2 7E 9E 92 65 C4 6F 4B 4D DA 11 F9 40
```

A.4.7 Example 7

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 160-bit string.

50 AB F5 70 6A 15 09 90 A0 8B 2C 5E A4 0F A0 E5 85 55 47 32

A.4.8 Example 8

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

'abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq'

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

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 X_0, X_1, X_2, X_3, X_4, obtained during the processing of the first block.

0116FC17, 67452301, 7BF36AE2, 98BADCFE, 10325476
EBF3B452, 0116FC17, 59D148C0, 7BF36AE2, 98BADCFE
5109913A, EBF3B452, C045BF05, 59D148C0, 7BF36AE2
2C4F6EAC, 5109913A, BAFCED14, C045BF05, 59D148C0
33F4AE5B, 2C4F6EAC, 9442644E, BAFCED14, C045BF05
96B85189, 33F4AE5B, 0B13DBAB, 9442644E, BAFCED14
DB04CB58, 96B85189, CCFD2B96, 0B13DBAB, 9442644E
45833F0F, DB04CB58, 65AE1462, CCFD2B96, 0B13DBAB
C565C35E, 45833F0F, 36C132D6, 65AE1462, CCFD2B96
6350AFDA, C565C35E, D160CFC3, 36C132D6, 65AE1462
8993EA77, 6350AFDA, B15970D7, D160CFC3, 36C132D6
E19ECAA2, 8993EA77, 98D42BF6, B15970D7, D160CFC3
8603481E, E19ECAA2, E264FA9D, 98D42BF6, B15970D7
32F94A85, 8603481E, B867B2A8, E264FA9D, 98D42BF6
B2E7A8BE, 32F94A85, A180D207, B867B2A8, E264FA9D
42637E39, B2E7A8BE, 4CBE52A1, A180D207, B867B2A8
6B068048, 42637E39, ACB9EA2F, 4CBE52A1, A180D207
426B9C35, 6B068048, 5098DF8E, ACB9EA2F, 4CBE52A1
944B1BD1, 426B9C35, 1AC1A012, 5098DF8E, ACB9EA2F
6C445652, 944B1BD1, 509AE70D, 1AC1A012, 5098DF8E
95836DA5, 6C445652, 6512C6F4, 509AE70D, 1AC1A012
09511177, 95836DA5, 9B111594, 6512C6F4, 509AE70D
E2B92DC4, 09511177, 6560DB69, 9B111594, 6512C6F4
FD224575, E2B92DC4, C254445D, 6560DB69, 9B111594
EEB82D9A, FD224575, 38AE4B71, C254445D, 6560DB69
5A142C1A, EEB82D9A, 7F48915D, 38AE4B71, C254445D
2972F7C7, 5A142C1A, BBAE0B66, 7F48915D, 38AE4B71
D526A644, 2972F7C7, 96850B06, BBAE0B66, 7F48915D
E1122421, D526A644, CA5CBDF1, 96850B06, BBAE0B66
05B457B2, E1122421, 3549A991, CA5CBDF1, 96850B06
A9C84BEC, 05B457B2, 78448908, 3549A991, CA5CBDF1
52E31F60, A9C84BEC, 816D15EC, 78448908, 3549A991

```
5AF3242C, 52E31F60, 2A7212FB, 816D15EC, 78448908
31C756A9, 5AF3242C, 14B8C7D8, 2A7212FB, 816D15EC
E9AC987C, 31C756A9, 16BCC90B, 14B8C7D8, 2A7212FB
AB7C32EE, E9AC987C, 4C71D5AA, 16BCC90B, 14B8C7D8
5933FC99, AB7C32EE, 3A6B261F, 4C71D5AA, 16BCC90B
43F87AE9, 5933FC99, AADF0CBB, 3A6B261F, 4C71D5AA
24957F22, 43F87AE9, 564CFF26, AADF0CBB, 3A6B261F
ADEB7478, 24957F22, 50FE1EBA, 564CFF26, AADF0CBB
D70E5010, ADEB7478, 89255FC8, 50FE1EBA, 564CFF26
79BCFB08, D70E5010, 2B7ADD1E, 89255FC8, 50FE1EBA
F9BCB8DE, 79BCFB08, 35C39404, 2B7ADD1E, 89255FC8
633E9561, F9BCB8DE, 1E6F3EC2, 35C39404, 2B7ADD1E
98C1EA64, 633E9561, BE6F2E37, 1E6F3EC2, 35C39404
C6EA241E, 98C1EA64, 58CFA558, BE6F2E37, 1E6F3EC2
A2AD4F02, C6EA241E, 26307A99, 58CFA558, BE6F2E37
C8A69090, A2AD4F02, B1BA8907, 26307A99, 58CFA558
88341600, C8A69090, A8AB53C0, B1BA8907, 26307A99
7E846F58, 88341600, 3229A424, A8AB53C0, B1BA8907
86E358BA, 7E846F58, 220D0580, 3229A424, A8AB53C0
8D2E76C8, 86E358BA, 1FA11BD6, 220D0580, 3229A424
CE892E10, 8D2E76C8, A1B8D62E, 1FA11BD6, 220D0580
EDEA95B1, CE892E10, 234B9DB2, A1B8D62E, 1FA11BD6
36D1230A, EDEA95B1, 33A24B84, 234B9DB2, A1B8D62E
776C3910, 36D1230A, 7B7AA56C, 33A24B84, 234B9DB2
A681B723, 776C3910, 8DB448C2, 7B7AA56C, 33A24B84
AC0A794F, A681B723, 1DDB0E44, 8DB448C2, 7B7AA56C
F03D3782, AC0A794F, E9A06DC8, 1DDB0E44, 8DB448C2
9EF775C3, F03D3782, EB029E53, E9A06DC8, 1DDB0E44
36254B13, 9EF775C3, BC0F4DE0, EB029E53, E9A06DC8
4080D4DC, 36254B13, E7BDDD70, BC0F4DE0, EB029E53
2BFAF7A8, 4080D4DC, CD8952C4, E7BDDD70, BC0F4DE0
513F9CA0, 2BFAF7A8, 10203537, CD8952C4, E7BDDD70
E5895C81, 513F9CA0, 0AFEBDEA, 10203537, CD8952C4
1037D2D5, E5895C81, 144FE728, 0AFEBDEA, 10203537
14A82DA9, 1037D2D5, 79625720, 144FE728, 0AFEBDEA
6D17C9FD, 14A82DA9, 440DF4B5, 79625720, 144FE728
2C7B07BD, 6D17C9FD, 452A0B6A, 440DF4B5, 79625720
FDF6EFFF, 2C7B07BD, 5B45F27F, 452A0B6A, 440DF4B5
112B96E3, FDF6EFFF, 4B1EC1EF, 5B45F27F, 452A0B6A
84065712, 112B96E3, FF7DBBFF, 4B1EC1EF, 5B45F27F
AB89FB71, 84065712, C44AE5B8, FF7DBBFF, 4B1EC1EF
C5210E35, AB89FB71, A10195C4, C44AE5B8, FF7DBBFF
352D9F4B, C5210E35, 6AE27EDC, A10195C4, C44AE5B8
1A0E0E0A, 352D9F4B, 7148438D, 6AE27EDC, A10195C4
D0D47349, 1A0E0E0A, CD4B67D2, 7148438D, 6AE27EDC
AD38620D, D0D47349, 86838382, CD4B67D2, 7148438D
D3AD7C25, AD38620D, 74351CD2, 86838382, CD4B67D2
8CE34517, D3AD7C25, 6B4E1883, 74351CD2, 86838382
```

The following are (hexadecimal representations of) the successive values of the variables X_0, X_1, X_2, X_3, X_4, obtained during the processing of the second block.

```
2DF257E9, F4286818, B0DEC9EB, 0408F581, 84677148
4D3DC58F, 2DF257E9, 3D0A1A06, B0DEC9EB, 0408F581
C352BB05, 4D3DC58F, 4B7C95FA, 3D0A1A06, B0DEC9EB
EEF743C6, C352BB05, D34F7163, 4B7C95FA, 3D0A1A06
41E34277, EEF743C6, 70D4AEC1, D34F7163, 4B7C95FA
5443915C, 41E34277, BBBDD0F1, 70D4AEC1, D34F7163
E7FA0377, 5443915C, D078D09D, BBBDD0F1, 70D4AEC1
C6946813, E7FA0377, 1510E457, D078D09D, BBBDD0F1
```

FDDE1DE1, C6946813, F9FE80DD, 1510E457, D078D09D
B8538ACA, FDDE1DE1, F1A51A04, F9FE80DD, 1510E457
6BA94F63, B8538ACA, 7F778778, F1A51A04, F9FE80DD
43A2792F, 6BA94F63, AE14E2B2, 7F778778, F1A51A04
FECD7BBF, 43A2792F, DAEA53D8, AE14E2B2, 7F778778
A2604CA8, FECD7BBF, D0E89E4B, DAEA53D8, AE14E2B2
258B0BAA, A2604CA8, FFB35EEF, D0E89E4B, DAEA53D8
D9772360, 258B0BAA, 2898132A, FFB35EEF, D0E89E4B
5507DB6E, D9772360, 8962C2EA, 2898132A, FFB35EEF
A51B58BC, 5507DB6E, 365DC8D8, 8962C2EA, 2898132A
C2EB709F, A51B58BC, 9541F6DB, 365DC8D8, 8962C2EA
D8992153, C2EB709F, 2946D62F, 9541F6DB, 365DC8D8
37482F5F, D8992153, F0BADC27, 2946D62F, 9541F6DB
EE8700BD, 37482F5F, F6264854, F0BADC27, 2946D62F
9AD594B9, EE8700BD, CDD20BD7, F6264854, F0BADC27
8FBAA5B9, 9AD594B9, 7BA1C02F, CDD20BD7, F6264854
88FB5867, 8FBAA5B9, 66B5652E, 7BA1C02F, CDD20BD7
EEC50521, 88FB5867, 63EEA96E, 66B5652E, 7BA1C02F
50BCE434, EEC50521, E23ED619, 63EEA96E, 66B5652E
5C416DAF, 50BCE434, 7BB14148, E23ED619, 63EEA96E
2429BE5F, 5C416DAF, 142F390D, 7BB14148, E23ED619
0A2FB108, 2429BE5F, D7105B6B, 142F390D, 7BB14148
17986223, 0A2FB108, C90A6F97, D7105B6B, 142F390D
8A4AF384, 17986223, 028BEC42, C90A6F97, D7105B6B
6B629993, 8A4AF384, C5E61888, 028BEC42, C90A6F97
F15F04F3, 6B629993, 2292BCE1, C5E61888, 028BEC42
295CC25B, F15F04F3, DAD8A664, 2292BCE1, C5E61888
696DA404, 295CC25B, FC57C13C, DAD8A664, 2292BCE1
CEF5AE12, 696DA404, CA573096, FC57C13C, DAD8A664
87D5B80C, CEF5AE12, 1A5B6901, CA573096, FC57C13C
84E2A5F2, 87D5B80C, B3BD6B84, 1A5B6901, CA573096
03BB6310, 84E2A5F2, 21F56E03, B3BD6B84, 1A5B6901
C2D8F75F, 03BB6310, A138A97C, 21F56E03, B3BD6B84
BFB25768, C2D8F75F, 00EED8C4, A138A97C, 21F56E03
28589152, BFB25768, F0B63DD7, 00EED8C4, A138A97C
EC1D3D61, 28589152, 2FEC95DA, F0B63DD7, 00EED8C4
3CAED7AF, EC1D3D61, 8A162454, 2FEC95DA, F0B63DD7
C3D033EA, 3CAED7AF, 7B074F58, 8A162454, 2FEC95DA
7316056A, C3D033EA, CF2BB5EB, 7B074F58, 8A162454
46F93B68, 7316056A, B0F40CFA, CF2BB5EB, 7B074F58
DC8E7F26, 46F93B68, 9CC5815A, B0F40CFA, CF2BB5EB
850D411C, DC8E7F26, 11BE4EDA, 9CC5815A, B0F40CFA
7E4672C0, 850D411C, B7239FC9, 11BE4EDA, 9CC5815A
89FBD41D, 7E4672C0, 21435047, B7239FC9, 11BE4EDA
1797E228, 89FBD41D, 1F919CB0, 21435047, B7239FC9
431D65BC, 1797E228, 627EF507, 1F919CB0, 21435047
2BDBB8CB, 431D65BC, 05E5F88A, 627EF507, 1F919CB0
6DA72E7F, 2BDBB8CB, 10C7596F, 05E5F88A, 627EF507
A8495A9B, 6DA72E7F, CAF6EE32, 10C7596F, 05E5F88A
E785655A, A8495A9B, DB69CB9F, CAF6EE32, 10C7596F
5B086C42, E785655A, EA1256A6, DB69CB9F, CAF6EE32
A65818F7, 5B086C42, B9E15956, EA1256A6, DB69CB9F
7AAB101B, A65818F7, 96C21B10, B9E15956, EA1256A6
93614C9C, 7AAB101B, E996063D, 96C21B10, B9E15956
F66D9BF4, 93614C9C, DEAAC406, E996063D, 96C21B10
D504902B, F66D9BF4, 24D85327, DEAAC406, E996063D
60A9DA62, D504902B, 3D9B66FD, 24D85327, DEAAC406
8B687819, 60A9DA62, F541240A, 3D9B66FD, 24D85327
083E90C3, 8B687819, 982A7698, F541240A, 3D9B66FD
F6226BBF, 083E90C3, 62DA1E06, 982A7698, F541240A
76C0563B, F6226BBF, C20FA430, 62DA1E06, 982A7698

```
989DD165, 76C0563B, FD889AEF, C20FA430, 62DA1E06
8B2C7573, 989DD165, DDB0158E, FD889AEF, C20FA430
AE1B8E7B, 8B2C7573, 66277459, DDB0158E, FD889AEF
CA1840DE, AE1B8E7B, E2CB1D5C, 66277459, DDB0158E
16F3BABB, CA1840DE, EB86E39E, E2CB1D5C, 66277459
D28D83AD, 16F3BABB, B2861037, EB86E39E, E2CB1D5C
6BC02DFE, D28D83AD, C5BCEEAE, B2861037, EB86E39E
D3A6E275, 6BC02DFE, 74A360EB, C5BCEEAE, B2861037
DA955482, D3A6E275, 9AF00B7F, 74A360EB, C5BCEEAE
58C0AAC0, DA955482, 74E9B89D, 9AF00B7F, 74A360EB
906FD62C, 58C0AAC0, B6A55520, 74E9B89D, 9AF00B7F
```

The hash-code is the following 160-bit string.

```
84 98 3E 44 1C 3B D2 6E BA AE 4A A1 F9 51 29 E5 E5 46 70 F1
```

A.4.9 Example 9

In this example the data-string is the 1000000-byte string consisting of the ASCII-coded version of 'a' repeated 10^6 times.

The hash-code is the following 160-bit string.

```
34 AA 97 3C D4 C4 DA A4 F6 1E EB 2B DB AD 27 31 65 34 01 6F
```

Annex B

(Informative)

Formal specifications

B.0 Introduction

In the following sections are complete specifications of the Dedicated Hash-Functions 1, 2 and 3 in the specification language called Z. The notation for Z is that as described in [1].

The Z retains much of the naming, structure, etc. used in the main body of this standard.

The Z is written purely in Z, including the comments. The comments point to sections of the main text of the standard from which the Z is derived.

The Z models a message as a sequence of natural numbers 0 and 1 (*String*).

B.1 Specification of Dedicated Hash-Function 1

#3 Definitions

#3.2 round-function

$$Bit == \{0, 1\}$$
$$String == \operatorname{seq} Bit$$

$$L_1 : \mathbb{N}$$
$$L_2 : \mathbb{N}_1$$

$$String\_L_1 == \{ s : String \mid \#s = L_1 \}$$
$$String\_L_2 == \{ s : String \mid \#s = L_2 \}$$

$$\phi : String\_L_1 \times String\_L_2 \longrightarrow String\_L_2$$

#3.3 word

$$Word == \{ w : String \mid \#w = 32 \}$$
$$Word\_capacity == 2 \uparrow 32$$
$$Word\_capacity\_m\_1 == (2 \uparrow 32) - 1$$
$$IWord == 0 \,..\, Word\_capacity\_m\_1$$

#4 Symbols and notation

$S^n()$ It is only necessary to define S as S^n (relation iteration) is defined in Z.

$$S : Word \longrightarrow Word$$

$$\forall A : Word \bullet$$
$$\quad (\textbf{let } I == W\_to\_I(A) \bullet$$
$$\quad (\textbf{let } Shift\_I == (I * 2 + (I \textbf{ div } (2 \uparrow 31))) \textbf{ mod } (2 \uparrow 32) \bullet$$
$$\quad\quad S(A) = I\_to\_W(Shift\_I)))$$

$\wedge$ $\vee$ $\oplus$ These are defined for words only as that is all that is required.

$$BO == Bit \times Bit \longrightarrow Bit$$

$$\begin{array}{|l} LO : Word \times Word \times BO \longrightarrow Word \\ \hline \forall\, p, q : Word; bo : BO \bullet \\ \quad LO(p, q, bo) = \{\, n : 1 \,..\, \#p \bullet n \mapsto bo(p(n), q(n)) \,\} \end{array}$$

$$\begin{array}{|l} \_ \text{ xor } \_, \_ \text{ or } \_, \_ \text{ and } \_ : BO \\ \hline 0 \text{ xor } 1 = 1 \\ 0 \text{ xor } 0 = 0 \\ 1 \text{ xor } 0 = 1 \\ 1 \text{ xor } 1 = 0 \\ \\ 0 \text{ or } 1 = 1 \\ 0 \text{ or } 0 = 0 \\ 1 \text{ or } 1 = 1 \\ \\ 0 \text{ and } 1 = 0 \\ 0 \text{ and } 0 = 0 \\ 1 \text{ and } 1 = 1 \end{array}$$

$$\begin{array}{|l} \_ \text{XOR} \_, \_ \text{OR} \_, \_ \text{AND} \_ : Word \times Word \longrightarrow Word \\ \hline \forall\, A, B : Word \bullet \\ \quad A \text{ XOR } B = LO(A, B, (\_ \text{ xor } \_)) \wedge \\ \quad A \text{ OR } B = LO(A, B, (\_ \text{ or } \_)) \wedge \\ \quad A \text{ AND } B = LO(A, B, (\_ \text{ and } \_)) \end{array}$$

¬

$$\begin{array}{|l} \text{NOT} : Word \longrightarrow Word \\ \hline \forall\, A : Word \bullet \\ \quad \text{NOT } A = A \text{ XOR } \{\, n : 1 \,..\, \#A \bullet n \mapsto 1 \,\} \end{array}$$

⊞

$$\begin{array}{|l} \_ \boxplus \_ : Word \times Word \longrightarrow Word \\ \hline \forall\, A, B : Word \bullet \\ \quad A \boxplus B = I\_to\_W((W\_to\_I(A) + W\_to\_I(B)) \bmod Word\_capacity) \end{array}$$

#5 Requirements

#6 Model for dedicated hash-functions

#6.1 General

$$L_H : \mathbb{N}_1$$
$$L_H \leq L_2$$

$$Byte == \{ b : String \mid \#b = 8 \}$$
$$IByte == 0 \,..\, 255$$

$$B\_to\_I : Byte \rightarrow IByte$$
$$\forall x : Byte \bullet B\_to\_I(x) = x(1) * 2 \upharpoonright 7 + x(2) * 2 \upharpoonright 6 + x(3) * 2 \upharpoonright 5 + x(4) * 2 \upharpoonright 4 + x(5) * 2 \upharpoonright 3 + x(6) * 2 \upharpoonright 2 + x(7) * 2 + x(8)$$

#6.2 Hashing operation

$$IV : String\_L_2$$

$$Maximum\_Length\_of\_String : \mathbb{N}$$

$$hash : String \nrightarrow String\_L_H$$
$$\forall D : String \mid \#D \leq Maximum\_Length\_of\_String \bullet hash(D) = (\textbf{let } PD == pad(D) \bullet (\textbf{let } SD == split(PD) \bullet (\textbf{let } H_q == iterate(SD, IV) \bullet truncate(H_q))))$$

#6.2.1 Step 1 (padding)

$$StringMultiple\_L_1 == \{ s : String \mid \#s \bmod L_1 = 0 \}$$

$$pad : String \rightarrow StringMultiple\_L_1$$

#6.2.2 Step 2 (splitting)

$$StringBlocks == seq_1 String\_L_1$$

$$split : StringMultiple\_L_1 \rightarrow StringBlocks$$
$$split = \{ sml1 : StringMultiple\_L_1; sb : StringBlocks \mid sml1 = \frown / sb \bullet sml1 \mapsto sb \}$$

#6.2.3 Step 3 (iteration)

$$iterate : StringBlocks \times String\_L_2 \nrightarrow String\_L_2$$
$$\forall sb : StringBlocks; H_{i-1} : String\_L_2 \mid \#sb \geq 1 \bullet iterate(sb, H_{i-1}) = (\textbf{let } D_i == sb(1) \bullet (\textbf{let } H_i == \phi(D_i, H_{i-1}) \bullet \textbf{if } \#sb = 1 \textbf{ then } H_i \textbf{ else } iterate(tail\, sb, H_i)))$$

#6.2.4 Step 4 (truncation)

$$String\_L_H == \{ s : String \mid \#s = L_H \}$$

$$\begin{array}{|l} truncate : String\_L_2 \rightarrow String\_L_H \\ \hline \forall\, sy : String\_L_2 \bullet \\ \quad truncate(sy) = (1 \,..\, L_H) \upharpoonright sy \end{array}$$

#7 Dedicated Hash-Function 1

#7.1 General

$$Maximum\_Length\_of\_String = (2 \uparrow 64) - 1$$

#7.2 Parameters, functions and constants

#7.2.1 Parameters

$$L_1 = 512$$
$$L_2 = 160$$
$$L_H = 160$$

#7.2.2 Byte ordering convention

$$\begin{array}{|l} W\_to\_I : Word \rightarrow IWord \\ \hline \forall\, w : Word \bullet \\ \quad W\_to\_I(w) = \\ \qquad (\textbf{let}\ B_0 == B\_to\_I((1 \,..\, 8) \upharpoonright w) \bullet \\ \qquad (\textbf{let}\ B_1 == B\_to\_I((9 \,..\, 16) \upharpoonright w) \bullet \\ \qquad (\textbf{let}\ B_2 == B\_to\_I((17 \,..\, 24) \upharpoonright w) \bullet \\ \qquad (\textbf{let}\ B_3 == B\_to\_I((25 \,..\, 32) \upharpoonright w) \bullet \\ \qquad\quad B_3 * 2 \uparrow 24 + B_2 * 2 \uparrow 16 + B_1 * 2 \uparrow 8 + B_0)))) \end{array}$$

$$\begin{array}{|l} I\_to\_W : IWord \rightarrow Word \\ \hline I\_to\_W = W\_to\_I^{\sim} \end{array}$$

#7.2.3 Functions

$$Indexed\_g == \{ g : \mathrm{seq}\,(Word \times Word \times Word \rightarrow Word) \mid \#g = 80 \}$$

$$
\begin{array}{|l}
g : Indexed\_g \\
\hline
\forall X_0, X_1, X_2 : Word \bullet \\
\quad (\forall i : 1 \mathinner{..} 16 \bullet \\
\qquad g(i)(X_0, X_1, X_2) = X_0 \text{ XOR } X_1 \text{ XOR } X_2) \wedge \\
\quad (\forall i : 17 \mathinner{..} 32 \bullet \\
\qquad g(i)(X_0, X_1, X_2) = (X_0 \text{ AND } X_1) \text{ OR } (\text{NOT } X_0 \text{ AND } X_2)) \wedge \\
\quad (\forall i : 33 \mathinner{..} 48 \bullet \\
\qquad g(i)(X_0, X_1, X_2) = (X_0 \text{ OR NOT } X_1) \text{ XOR } X_2) \wedge \\
\quad (\forall i : 49 \mathinner{..} 64 \bullet \\
\qquad g(i)(X_0, X_1, X_2) = (X_0 \text{ AND } X_2) \text{ OR } (X_1 \text{ AND NOT } X_2)) \wedge \\
\quad (\forall i : 65 \mathinner{..} 80 \bullet \\
\qquad g(i)(X_0, X_1, X_2) = X_0 \text{ XOR } (X_1 \text{ OR NOT } X_2))
\end{array}
$$

#7.2.4 Constants

$x00000000 == 0$
$x5A827999 == 1518500249$
$x6ED9EBA1 == 1859775393$
$x8F1BBCDC == 2400959708$
$xA953FD4E == 2840853838$

$x50A28BE6 == 1352829926$
$x5C4DD124 == 1548603684$
$x6D703EF3 == 1836072691$
$x7A6D76E9 == 2053994217$

$Constants == \{ c : StringWord \mid \#c = 80 \}$

$$
\begin{array}{|l}
C, C' : Constants \\
\hline
(\forall\, i : 1 \,..\, 16 \bullet \\
\quad C(i) = I\_to\_W(x00000000)) \land \\
(\forall\, i : 17 \,..\, 32 \bullet \\
\quad C(i) = I\_to\_W(x5A827999)) \land \\
(\forall\, i : 33 \,..\, 48 \bullet \\
\quad C(i) = I\_to\_W(x6ED9EBA1)) \land \\
(\forall\, i : 49 \,..\, 64 \bullet \\
\quad C(i) = I\_to\_W(x8F1BBCDC)) \land \\
(\forall\, i : 65 \,..\, 80 \bullet \\
\quad C(i) = I\_to\_W(xA953FD4E)) \land \\
\\
(\forall\, i : 1 \,..\, 16 \bullet \\
\quad C'(i) = I\_to\_W(x50A28BE6)) \land \\
(\forall\, i : 17 \,..\, 32 \bullet \\
\quad C'(i) = I\_to\_W(x5C4DD124)) \land \\
(\forall\, i : 33 \,..\, 48 \bullet \\
\quad C'(i) = I\_to\_W(x6D703EF3)) \land \\
(\forall\, i : 49 \,..\, 64 \bullet \\
\quad C'(i) = I\_to\_W(x7A6D76E9)) \land \\
(\forall\, i : 65 \,..\, 80 \bullet \\
\quad C'(i) = I\_to\_W(x00000000))
\end{array}
$$

$$
\begin{aligned}
t == \\
&\langle 11, 14, 15, 12, 5, 8, 7, 9, 11, 13, 14, 15, 6, 7, 9, 8, \\
&7, 6, 8, 13, 11, 9, 7, 15, 7, 12, 15, 9, 11, 7, 13, 12, \\
&11, 13, 6, 7, 14, 9, 13, 15, 14, 8, 13, 6, 5, 12, 7, 5, \\
&11, 12, 14, 15, 14, 15, 9, 8, 9, 14, 5, 6, 8, 6, 5, 12, \\
&9, 15, 5, 11, 6, 8, 13, 12, 5, 12, 13, 14, 11, 8, 5, 6\rangle
\end{aligned}
$$

$$
\begin{aligned}
t' == \\
&\langle 8, 9, 9, 11, 13, 15, 15, 5, 7, 7, 8, 11, 14, 14, 12, 6, \\
&9, 13, 15, 7, 12, 8, 9, 11, 7, 7, 12, 7, 6, 15, 13, 11, \\
&9, 7, 15, 11, 8, 6, 6, 14, 12, 13, 5, 14, 13, 13, 7, 5, \\
&15, 5, 8, 11, 14, 14, 6, 14, 6, 9, 12, 9, 12, 5, 15, 8, \\
&8, 5, 12, 9, 12, 5, 14, 6, 8, 13, 6, 5, 15, 13, 11, 11\rangle
\end{aligned}
$$

Note values for a and a' are one greater than the normative text as sequences in Z start at 1.

$$
\begin{aligned}
a == \\
&\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, \\
&8, 5, 14, 2, 11, 7, 16, 4, 13, 1, 10, 6, 3, 15, 12, 9, \\
&4, 11, 15, 5, 10, 16, 9, 2, 3, 8, 1, 7, 14, 12, 6, 13, \\
&2, 10, 12, 11, 1, 9, 13, 5, 14, 4, 8, 16, 15, 6, 7, 3, \\
&5, 1, 6, 10, 8, 13, 3, 11, 15, 2, 4, 9, 12, 7, 16, 14\rangle
\end{aligned}
$$

$$a' ==$$
$$\langle 6, 15, 8, 1, 10, 3, 12, 5, 14, 7, 16, 9, 2, 11, 4, 13,$$
$$7, 12, 4, 8, 1, 14, 6, 11, 15, 16, 9, 13, 5, 10, 2, 3,$$
$$16, 6, 2, 4, 8, 15, 7, 10, 12, 9, 13, 3, 11, 1, 5, 14,$$
$$9, 7, 5, 2, 4, 12, 16, 1, 6, 13, 3, 14, 10, 8, 11, 15,$$
$$13, 16, 11, 5, 2, 6, 9, 8, 7, 3, 14, 15, 1, 4, 10, 12\rangle$$

#7.2.5 Initializing Value

$$x67452301 == 1732584193$$
$$Y_0 == I\_to\_W(x67452301)$$
$$xEFCDAB89 == 4023233417$$
$$Y_1 == I\_to\_W(xEFCDAB89)$$
$$x98BADCFE == 2562383102$$
$$Y_2 == I\_to\_W(x98BADCFE)$$
$$x10325476 == 271733878$$
$$Y_3 == I\_to\_W(x10325476)$$
$$xC3D2E1F0 == 3285377520$$
$$Y_4 == I\_to\_W(xC3D2E1F0)$$

$$IV = Y_0 \frown Y_1 \frown Y_2 \frown Y_3 \frown Y_4$$

#7.3 Padding method

$$\forall D : String \bullet$$
$$pad(D) =$$
$$(\textbf{let } L_D == \#D \bullet$$
$$(\textbf{let } Zeros == \{ n : 1 \,..\, ((447 - L_D) \bmod 512) \bullet n \mapsto 0 \} \bullet$$
$$(\textbf{let } Length\_D\_LSH == I\_to\_W(L_D \bmod (2 \uparrow 32)) \bullet$$
$$(\textbf{let } Length\_D\_MSH == I\_to\_W(L_D \operatorname{div} (2 \uparrow 32)) \bullet$$
$$D \frown \langle 1 \rangle \frown Zeros \frown Length\_D\_LSH \frown Length\_D\_MSH))))$$

#7.4 Description of the round-function

$$StringWord == \text{seq } Word$$

$$Split\_String\_to\_StringWord : String \longrightarrow StringWord$$
$$Split\_String\_to\_StringWord =$$
$$\{ s : String; sw : StringWord \mid s = \frown / sw \bullet s \mapsto sw \}$$

$$
\begin{array}{l}
L80 : StringWord \times StringWord \times \\
\quad Indexed\_g \times \mathrm{seq}\,\mathbb{N} \times \mathrm{seq}\,\mathbb{N} \times \\
\quad Constants \times 1\,..\,80 \longrightarrow StringWord \\
\hline
\forall\, Z, X : StringWord;\ g : Indexed\_g;\ t, a : \mathrm{seq}\,\mathbb{N}; \\
\quad C : Constants;\ i : 1\,..\,80 \mid \#Z - 16 \wedge \#X = 5 \bullet \\
\quad L80(Z, X, g, t, a, C, i) = \\
\qquad (\textbf{let}\ X0 == X(1); X1 == X(2); X2 == X(3); X3 == X(4); X4 == X(5) \bullet \\
\qquad (\textbf{let}\ W == S^{t(i)}(X0 \uplus g(i)(X1, X2, X3) \uplus Z(a(i)) \uplus C(i)) \uplus X4 \bullet \\
\qquad (\textbf{let}\ Y == \langle X4, W, X1, S^{10}(X2), X3\rangle \bullet \\
\qquad\quad \textbf{if}\ (i = 80) \\
\qquad\quad \textbf{then}\ Y \\
\qquad\quad \textbf{else}\ L80(Z, Y, g, t, a, C, i + 1))))
\end{array}
$$

$$
\begin{array}{l}
\forall\, sx : String\_L_1;\ sy : String\_L_2 \bullet \\
\quad \phi(sx, sy) = \\
\qquad (\textbf{let}\ Z == Split\_String\_to\_StringWord(sx) \bullet \\
\qquad (\textbf{let}\ Y == Split\_String\_to\_StringWord(sy) \bullet \\
\qquad (\textbf{let}\ X == L80(Z, Y, g, t, a, C, 1) \bullet \\
\qquad (\textbf{let}\ X' == L80(Z, Y, rev\, g, t', a', C', 1) \bullet \\
\qquad (\textbf{let}\ Y0 == Y(2) \uplus X(3) \uplus X'(4) \bullet \\
\qquad (\textbf{let}\ Y1 == Y(3) \uplus X(4) \uplus X'(5) \bullet \\
\qquad (\textbf{let}\ Y2 == Y(4) \uplus X(5) \uplus X'(1) \bullet \\
\qquad (\textbf{let}\ Y3 == Y(5) \uplus X(1) \uplus X'(2) \bullet \\
\qquad (\textbf{let}\ Y4 == Y(1) \uplus X(2) \uplus X'(3) \bullet \\
\qquad\quad Y0 \frown Y1 \frown Y2 \frown Y3 \frown Y4))))))))))
\end{array}
$$

B.1.1 Auxiliary functions

$$
\begin{array}{l}
\_ \uparrow \_ : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N} \\
\hline
\forall\, p : \mathbb{N} \bullet \\
\quad p \uparrow 0 = 1 \wedge \\
\quad (\forall\, n : \mathbb{N}_1 \bullet p \uparrow n = p * (p \uparrow (n - 1)))
\end{array}
$$