GB

中国国家标准汇编

2012年 修订-16

中国标准出版社

中国国家标准汇编

2012年修订-16

中国标准出版社　编

中国标准出版社
北　京

图书在版编目(CIP)数据

中国国家标准汇编:2012 年修订.16/中国标准出版社编.—北京:中国标准出版社,2013.9

ISBN 978-7-5066-7248-1

Ⅰ.①中… Ⅱ.①中… Ⅲ.①国家标准-汇编-中国-2012 Ⅳ.①T-652.1

中国版本图书馆 CIP 数据核字(2013)第 186461 号

中国标准出版社出版发行
北京市朝阳区和平里西街甲 2 号(100013)
北京市西城区三里河北街 16 号(100045)

网址 www.spc.net.cn
总编室:(010)64275323 发行中心:(010)51780235
读者服务部:(010)68523946

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

*

开本 880×1230 1/16 印张 38.25 字数 1 185 千字
2013 年 9 月第一版 2013 年 9 月第一次印刷

*

定价 220.00 元

出 版 说 明

1.《中国国家标准汇编》是一部大型综合性国家标准全集。自1983年起，按国家标准顺序号以精装本、平装本两种装帧形式陆续分册汇编出版。它在一定程度上反映了我国建国以来标准化事业发展的基本情况和主要成就，是各级标准化管理机构，工矿企事业单位，农林牧副渔系统，科研、设计、教学等部门必不可少的工具书。

2.《中国国家标准汇编》收入我国每年正式发布的全部国家标准，分为“制定”卷和“修订”卷两种编辑版本。

“制定”卷收入上一年度我国发布的、新制定的国家标准，顺延前年度标准编号分成若干分册，封面和书脊上注明“20××年制定”字样及分册号，分册号一直连续。各分册中的标准是按照标准编号顺序连续排列的，如有标准顺序号缺号的，除特殊情况注明外，暂为空号。

“修订”卷收入上一年度我国发布的、被修订的国家标准，视篇幅分设若干分册，但与“制定”卷分册号无关联，仅在封面和书脊上注明“20××年修订-1，-2，-3，……”字样。“修订”卷各分册中的标准，仍按标准编号顺序排列（但不连续）；如有遗漏的，均在当年最后一分册中补齐。需提请读者注意的是，个别非顺延前年度标准编号的新制定的国家标准没有收入在“制定”卷中，而是收入在“修订”卷中。

读者配套购买《中国国家标准汇编》“制定”卷和“修订”卷则可收齐由我社出版的上一年度我国制定和修订的全部国家标准。

3. 由于读者需求的变化，自1996年起，《中国国家标准汇编》仅出版精装本。

4. 2012年我国制修订国家标准共2 101项。本分册为“2012年修订-16”，收入新制修订的国家标准30项。

中国标准出版社

2013年7月

目　　录

ICS 35.040
L 80

中华人民共和国国家标准

GB/T 15852.2—2012

信息技术　安全技术　消息鉴别码　第2部分：采用专用杂凑函数的机制

Information technology—Security techniques—Message Authentication Codes (MACs)—Part 2: Mechanisms using a dedicated hash-function

(ISO/IEC 9797-2:2002, MOD)

2012-12-31 发布　　2013-06-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

前　言

GB/T 15852《信息技术　安全技术　消息鉴别码》由如下部分组成：

——第1部分：采用分组密码的机制；

——第2部分：采用专用杂凑函数的机制。

本部分是GB/T 15852的第2部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分修改采用ISO/IEC 9797-2:2002《信息技术　安全技术　消息鉴别码　第2部分：采用专用杂凑函数的机制》。增加了基于专用杂凑函数WHIRLPOOL的MAC生成方法及例子，更新了附录和参考文献，并将ISO/IEC 9797-2:2002中计算常数的6.3条调整到本部分的第9章。

本部分由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本部分起草单位：中国科学院软件研究所、信息安全国家重点实验室。

本部分主要起草人：吴文玲、张立廷、王鹏、吴双、张文涛、陈华、眭晗。

引　言

本部分规定的第一个 MAC 算法通常被称作 MDx-MAC。它调用一次完整的杂凑函数，但是对其中的轮函数做了细微的修改，把一个密钥加到了轮函数的附加常数上。第二个 MAC 算法通常被称作 HMAC，它调用两次完整的杂凑函数。第三个 MAC 算法是 MDx-MAC 的一个变种，它限制输入长度不大于 256 比特。在只处理较短输入的情况下，它有更好的性能。

本部分规定的三种 MAC 算法采用四种专用杂凑函数；其中，专用杂凑函数 1、2、3 和 4 分别是 ISO/IEC 10118-3:2004 中规定的专用杂凑函数 1、2、3 和 7。使用的专用杂凑函数也可以为国家密码管理部门批准的相应专用杂凑函数。

信息技术　安全技术　消息鉴别码
第2部分：采用专用杂凑函数的机制

1　范围

GB/T 15852 的本部分规定了三种采用专用杂凑函数的消息鉴别码算法。这些消息鉴别码算法可用作数据完整性检验，检验数据是否被非授权地改变。同样这些消息鉴别码算法也可用作消息鉴别，保证消息源的合法性。数据完整性和消息鉴别的强度依赖于密钥的长度及其保密性、杂凑函数的算法强度及其输出长度、消息鉴别码的长度和具体的消息鉴别码算法。

本部分适用于任何安全体系结构、进程或应用的安全服务。

2　规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 1988—1998　信息技术　信息交换用七位编码字符集(eqv ISO/IEC 646:1991)

ISO/IEC 10118-3:2004　信息技术　安全技术　杂凑函数　第3部分：专用杂凑函数(Information technology—Security techniques—Hash-functions—Part 3:Dedicated hash-functions)

3　术语和定义

下列术语和定义适用于本文件。

3.1

消息鉴别码　message authentication code;MAC

利用对称密码技术，以密钥为参数，由消息导出的数据项。任何持有这一密钥的实体，都可利用消息鉴别码检查消息的完整性和始发者。

3.2

消息鉴别码(MAC)算法密钥　MAC algorithm key

一种用于控制消息鉴别码算法运算的密钥。

3.3

消息鉴别码算法　message authentication code algorithm

消息鉴别码算法简称 MAC 算法，其输入为密钥和消息，输出为一个固定长度的比特串，满足下面两个性质：

——对于任何密钥和消息，MAC 算法都能够快速地计算。

——对于任何固定的密钥，攻击者在没有获得密钥信息的情况下，即使获得了一些(消息，MAC)对，对任何新的消息预测其 MAC 在计算上是不可行的。

注：一个 MAC 算法有时被称作一个密码校验函数。计算不可行性依赖于使用者具体的安全要求及其环境。

3.4

输出变换　output transformation

应用在算法中，对迭代操作的输出所进行的变换。

3.5

抗碰撞杂凑函数 collision-resistant hash-function

满足如下性质的杂凑函数：

——寻找两个不同的输入，使得它们的输出相同，在计算上是不可行的。

3.6

消息比特串(数据) data string (data)

杂凑函数的输入比特串。

3.7

杂凑值 hash-code

杂凑函数的输出比特串。

3.8

杂凑函数 hash-function

将任意长消息比特串映射到定长比特串的函数，并且满足如下两个性质：

——对于任何输出，找到它所对应的输入在计算上是不可行的。

——对于任何输入，找到区别于它且和它具有相同输出的输入在计算上是不可行的。

3.9

初始值 initializing value

杂凑函数开始工作时用到的值。

3.10

填充 padding

在消息比特串后面附加额外比特串的操作。

3.11

分组 block

一种定义了长度的比特串。

3.12

轮函数 round-function

将两个长度为 L_1 和 L_2 的比特串映射到一个长度为 L_2 的比特串的函数 $\phi(\cdot,\cdot)$。

注：它被反复地用在杂凑函数中，将长度为 L_1 的比特串和前面长度为 L_2 的输出值相合并。

3.13

字 word

长度为 32 位的比特串。

4 符号和记法

下列符号和记法适用于本部分。

D、D'	将要被输入到 MAC 算法的消息比特串
m	MAC 值的比特长度
q	经过填充和分割操作后，消息比特串 D 的分组个数
$MSB_j(X)$	比特串 X 最左边的 j 比特
$X \oplus Y$	比特串 X 和 Y 的异或值
$X \| Y$	按顺序将比特串 X 和 Y 连接所构成的比特串
$:=$	MAC 算法定义中使用的赋值符号
$\overline{D}$	经过填充的消息比特串
h	杂凑函数

h'	被修改了常数和初始值的杂凑函数 h
$\overline{h}$	简化的杂凑函数 h，没有数据填充和长度附加
H'、H''	长度为 L_2 比特串，在 MAC 算法计算中被用来存储临时结果
IV、IV'、IV_1、IV_2	初始值
k	MAC 算法密钥的比特长度
K	MAC 算法的密钥
K'、K_0、K_1、K_2	MAC 算法 1 和 3 中的导出密钥
$\overline{K}$、$\overline{K_1}$、$\overline{K_2}$	MAC 算法 2 中的导出密钥
$\widetilde{L}$	MAC 算法 3 中表示消息长度的比特串
$OPAD$、$IPAD$	MAC 算法 2 中使用的常数比特串
R、S_0、S_1、S_2	MAC 算法 1 和 3 中，用来导出一系列常数的常数比特串
T_0、T_1、T_2	MAC 算法 1 和 3 中，用来导出子密钥的 128 比特常数
U_0、U_1、U_2	MAC 算法 1 和 3 中，用来导出子密钥的 768 比特常数
ϕ'	使用修改后常数的轮函数
$K_1[i]$	128 比特串 K_1 的第 i 个字，即：$K_1=K_1[0]\parallel K_1[1]\parallel K_1[2]\parallel K_1[3]$
H	杂凑值
L_X	比特串 X 的比特长度
C_i、C_i'	轮函数中用到的常数字
CC_i	专用杂凑函数 4 中用到的常数矩阵
L_1	输入到轮函数 ϕ 的两个比特串中，第一个比特串的比特长度
L_2	输入到轮函数 ϕ 的两个比特串中，第二个比特串的比特长度；轮函数 ϕ 输出值的比特长度；初始值 IV 的比特长度
ϕ	轮函数，即：若 X 和 Y 分别表示长度为 L_1 和 L_2 的比特串，则 $\phi(X,Y)$ 表示将 ϕ 作用到 X 和 Y 所得到的比特串
$+_{32}$	模 2^{32} 加法操作，即：若 A 和 B 是字，那么把 A 和 B 看作是整数的 2 进制表示，计算它们的和再模 2^{32}，所得到的结果在 0 和 $2^{32}-1$ 之间，把它看作为字，记作 $A+_{32}B$

注：$\overline{h}$ 只能被用来处理长度为 L_1 整数倍的输入比特串。

5 要求

采用本部分 MAC 算法的使用者应当选择：

1） 从第 6、7、8 章中选取一种 MAC 算法；

2） 从 ISO/IEC 10118-3:2004 中的专用杂凑函数 1、2、3 和 7 中选取一个杂凑函数；

3） MAC 的长度 m。

对于 MAC 算法 1 和 2，MAC 的长度 m 应该是一个正整数并且不大于杂凑值长度 L_H。对于 MAC 算法 3，MAC 的长度 m 应该是一个正整数并且不大于杂凑值长度的二分之一，即 $m\leqslant L_H/2$。

对于 MAC 算法 1 和 2，消息比特串 D 的比特长度不大于 $2^{64}-1$；对于 MAC 算法 3，消息比特串 D 的比特长度不大于 256。

对一个具体 MAC 算法、专用杂凑函数、m 值的选择超出了本部分所规定的范围。

注：上述选择将影响 MAC 算法的安全强度，具体请参考附录 B。

生成 MAC 和验证 MAC 应当使用同样的密钥。

6 MAC 算法 1

MAC 算法 1 计算 MAC 值要求调用一次杂凑函数，而且要求修改其中的轮函数常数。

杂凑函数应当从 ISO/IEC 10118-3:2004 中的专用杂凑函数 1、2、3 和 7 中选取。

密钥长度 k 不大于 128 比特。

注：本条款包括 MDx-MAC 的描述[5]。具体来讲，若采用专用杂凑函数 1，MAC 算法 1 也被称作 RIPEMD-160-MAC；若采用专用杂凑函数 2，MAC 算法 1 也被称作 RIPEMD-128-MAC；若采用专用杂凑函数 3，MAC 算法 1 也被称作 SHA-1-MAC；若采用专用杂凑函数 4，MAC 算法 1 也被称作 WHIRLPOOL-MAC。

6.1 MAC 算法 1 的描述

MAC 算法 1 要求如下五步操作：密钥扩展、修改常数和初始值、杂凑操作、输出变换和截断操作。

6.1.1 密钥扩展

若 K 长度小于 128 比特，那么将 K 重复足够多次数，从连接起来的比特串中选取最左边 128 比特作为 128 比特密钥 K'（若 K 的长度恰好为 128 比特，则 $K' := K$），即：

$$K' := MSB_{128}(K \| K \| \cdots \| K)$$

按照如下操作计算子密钥 K_0、K_1 和 K_2：

$$K_0 := \overline{h}(K' \| U_0 \| K')$$

$$K_1 := MSB_{128}(\overline{h}(K' \| U_1 \| K'))$$

$$K_2 := MSB_{128}(\overline{h}(K' \| U_2 \| K'))$$

其中，U_0、U_1 和 U_2 是 768 比特的常数，在第 9 章中有定义。$\overline{h}$ 表示简化的杂凑函数 h，即没有数据填充和长度附加。

注：数据填充和长度附加可以被省略，是因为在这里输入比特串的长度总是 $2L_1$ 比特。

导出的密钥 K_1 被分割成四个字，表示为 $K_1[i](0 \leqslant i \leqslant 3)$，即：$K_1 = K_1[0] \| K_1[1] \| K_1[2] \| K_1[3]$。

从比特串到字的转换，需要规定字节的排列顺序。在这里的转换中，采用 ISO/IEC 10118-3:2004 中对所有专用杂凑函数规定的字节排列顺序。

6.1.2 修改常数和初始值

轮函数中采用的附加常数，被修改为它与 K_1 四个字中的一个进行模 2^{32} 加的结果，比如说：

$$C_0 := C_0 +_{32} K_1[0]$$

在条款 9 中具体规定了 K_1 中的哪个字与哪个常数相加。用 $IV' := K_0$ 取代杂凑函数的初始值 IV，所得的杂凑函数记作 h'，其中的轮函数记作 ϕ'。

6.1.3 杂凑操作

用 D 表示输入到被修改的杂凑函数 h' 中的比特串，即：

$$H' := h'(D)$$

6.1.4 输出变换

再一次应用被修改的轮函数 ϕ'，其中输入的第一个参数为 $K_2 \| (K_2 \oplus T_0) \| (K_2 \oplus T_1) \| (K_2 \oplus T_2)$，第二个参数为 H'（杂凑操作的结果），即：

$$H'' := \phi'(K_2 \| (K_2 \oplus T_0) \| (K_2 \oplus T_1) \| (K_2 \oplus T_2), H')$$

这里 T_0、T_1 和 T_2 都是长度为 128 的比特串，在条款 9 中对所有专用杂凑函数均有定义。

注：输出变换对应于处理一个额外的数据分组，这个额外的数据分组是在数据填充和长度附加操作之后，由 K_2 导出。

6.1.5 截断操作

取比特串 H'' 最左边 m 比特，作为 MAC 值，即：

$$\text{MAC} := MSB_m(H'')$$

6.2 MAC 算法 1 的效率

假定填充后的消息比特串包括 q 个分组（这里填充方法由具体的杂凑函数决定），那么 MAC 算法 1 调用轮函数 $q+7$ 次。

通过预计算 K_0、K_1 和 K_2，并且在杂凑函数的应用中用 IV' 取代 IV，MAC 算法 1 调用轮函数的次数可以降低到 $q+1$ 次。

处理长的消息比特串时，MAC 算法 1 和相应杂凑函数的性能相当。

7 MAC 算法 2

MAC 算法 2 计算 MAC 值要求调用两次杂凑函数。

杂凑函数应当从 ISO/IEC 10118-3:2004 中的专用杂凑函数 1、2、3 和 7 中选取，并且要求 L_1 是 8 的正整数倍，$L_2 \leqslant L_1$。

注：ISO/IEC 10118-3:2004 中的杂凑函数 1、2、3 和 7 满足这些条件。

密钥长度 k 不小于 L_2 比特（L_2 是杂凑值的比特长度），不大于 L_1 比特（L_1 为输入到轮函数的比特串的比特长度），即：$L_2 \leqslant k \leqslant L_1$。

7.1 MAC 算法 2 的描述

MAC 算法 2 要求如下四步操作：密钥扩展、杂凑操作、输出变换和截断操作。

7.1.1 密钥扩展

在密钥 K 的右侧填充 L_1-k 个 0，所得的长度为 L_1 的比特串记作 $\overline{K}$。

按照如下的方法，将 $\overline{K}$ 扩展为两个子密钥 $\overline{K_1}$ 和 $\overline{K_2}$：

- 将 16 进制的值“36”（二进制表示为“00110110”）重复 $L_1/8$ 次连接起来，所得比特串记作 $IPAD$。然后将 $\overline{K}$ 和比特串 $IPAD$ 相异或，记作 $\overline{K_1}$。即：

$$\overline{K_1} := \overline{K} \oplus IPAD$$

- 将 16 进制的值“5C”（二进制表示为“01011100”）重复 $L_1/8$ 次连接起来，所得比特串记作 $OPAD$。然后将 $\overline{K}$ 和比特串 $OPAD$ 相异或，记作 $\overline{K_2}$。即：

$$\overline{K_2} := \overline{K} \oplus OPAD$$

7.1.2 杂凑操作

将 $\overline{K_1}$ 和 D 相连接，作为输入到杂凑函数的比特串，即：

$$H' := h(\overline{K_1} \| D)$$

7.1.3 输出变换

将$\overline{K_2}$和H'相连接，作为输入到杂凑函数的比特串，即：

$$H'' := h(\overline{K_2} \| H')。$$

7.1.4 截断操作

取比特串H''最左边m比特，作为MAC值，即：

$$\text{MAC} := MSB_m(H'')。$$

7.2 MAC算法2的效率

假定填充后的消息比特串包括q个分组(这里填充方法由具体的杂凑函数决定)，那么采用专用杂凑函数1、2和3时，MAC算法2调用轮函数$q+3$次；采用专用杂凑函数4时，MAC算法2调用轮函数$q+4$次。

通过修改杂凑函数代码，MAC算法2调用轮函数的次数可以降低2次。

使用者可以预计算$IV_1 := \phi(\overline{K_1}, IV)$和$IV_2 := \phi(\overline{K_2}, IV)$，并且在第一次调用杂凑函数时用$IV_1$取代$IV$，在输出变换中(第二次调用杂凑函数)用$IV_2$取代$IV$。同时，这也要求对填充方法进行修改。事实上，对杂凑函数实际输入的比特长度少了L_1，这样必须把L_1的值加到L_D上。

处理长的消息比特串时，MAC算法2和相应杂凑函数的性能相当。

8 MAC算法3

注：本条款包括MAC算法1的一个变种，对短的输入(不大于256比特)做了优化。

MAC算法3计算MAC值，要求调用7次简化的轮函数；但是通过预计算，可以降低到调用一次简化的轮函数。

杂凑函数应当从ISO/IEC 10118-3:2004中的专用杂凑函数1、2、3和7中选取。

密钥长度k不大于128比特，MAC值长度m不大于$L_H/2$比特。

8.1 MAC算法3的描述

MAC算法3要求如下五步操作：密钥扩展、修改轮函数的常数、数据填充、应用轮函数和截断操作。

8.1.1 密钥扩展

若K长度小于128比特，那么将K重复足够多次数，从连接起来的比特串中选取最左边128比特作为128比特密钥K'(若K的长度恰好为128比特，则$K' := K$)，即：

$$K' := MSB_{128}(K \| K \| \cdots \| K)$$

按照如下操作计算子密钥K_0、K_1和K_2：

$$K_0 := \overline{h}(K' \| U_0 \| K')$$

$$K_1 := MSB_{128}(\overline{h}(K' \| U_1 \| K'))$$

$$K_2 := MSB_{128}(\overline{h}(K' \| U_2 \| K'))$$

其中，U_0、U_1和U_2是768比特的常数，在条款9中有定义。$\overline{h}$表示简化的杂凑函数h，即没有数据填充和长度附加。

注：数据填充和长度附加可以被省略，是因为在这里输入比特串的长度总是$2L_1$比特。

导出的密钥 K_1 被分割成四个字，表示为 $K_1[i](0 \leqslant i \leqslant 3)$，即：$K_1 = K_1[0] \| K_1[1] \| K_1[2] \| K_1[3]$。

从比特串到字的转换，需要规定字节的排列顺序。在这里的转换中，采用 ISO/IEC 10118-3:2004 中对所有专用杂凑函数规定的字节排列顺序。

8.1.2 修改轮函数的常数

轮函数中采用的附加常数，被修改为它与 K_1 四个字中的一个进行模 2^{32} 加的结果，比如说：

$$C_0 := C_0 +_{32} K_1[0]$$

在第 9 章中具体规定了 K_1 中的哪个字与哪个常数相加。用 $IV' := K_0$ 取代杂凑函数的初始值 IV，所得的轮函数记作 ϕ'。

8.1.3 数据填充

对原始消息填充的比特串只用来计算 MAC，所以这些填充比特串（如果有）不必随原始消息存储或发送。MAC 的验证者应当知道填充比特串是否已经被存储或发送。

对要输入到 MAC 算法的消息比特串 D，在其右侧填充尽可能少（可能没有）的“0”以使得填充后比特串 $\overline{D}$ 的长度是 256 比特。

注：如果消息比特串 D 是空串，那么规定填充后的比特串 $\overline{D}$ 为 256 个“0”。

8.1.4 应用轮函数

消息比特串 D 的长度记作 L_D，其二进制表示记作 $\widetilde{L}$。在 $\widetilde{L}$ 最左边填充足够少的“0”使得 $\widetilde{L}$ 的长度为 128 比特，$\widetilde{L}$ 最右边的比特和 L_D 最低位相对应。

将 K_2、$\overline{D}$ 和 K_2 与 $\widetilde{L}$ 的异或值相连接，作为轮函数 ϕ'（使用修改过的常数）的输入，即：

$$H' := \phi'(K_2 \| \overline{D} \| (K_2 \oplus \widetilde{L}), IV')。$$

8.1.5 截断操作

取比特串 H' 最左边 m 比特，作为 MAC 值，即：

$$\mathrm{MAC} := MSB_m(H')。$$

8.2 MAC 算法 3 的效率

MAC 算法 3 需要调用 7 次轮函数，通过预计算 K_0、K_1 和 K_2，可以降低到一次。

9 常数的计算

本条款中规定的常数，将被用在 MAC 算法 1 和条款 8 的 MAC 算法 3 中。

比特串 T_i 和 U_i 是 MAC 算法中固定的元素，它们通过杂凑函数计算得到（只计算一次），并且在四个专用杂凑函数中各不相同。

128 比特的 T_i 和 768 比特的 U_i 按照如下的方法定义：

$$T_i := MSB_{128}(\overline{h}(S_i \| R)), \qquad i=0,1,2$$

$$U_i := T_i \| T_{i+1} \| T_{i+2} \| T_i \| T_{i+1} \| T_{i+2}, \qquad i=0,1,2$$

其中下标的加法是模 3 加。$R =$ "$ab\cdots yzAB\cdots YZ01\cdots 89$" 是 496 比特的常数，$S_0$、$S_1$ 和 S_2 都是 16 比特的常数，其中 S_i 通过重复两次数字 i 的 16 进制 ASCII 编码得到（比如说，S_1 的表示为 3131）。R 和 S_i 都采用 ASCII 编码，ASCII 编码等同于 GB/T 1988—1998 所使用的编码。

对于所有的常数 C_i、C'_i和所有的字 $K_1[i]$，最高位和最左边的比特相对应。常数 C_i 和 C'_i用 16 进制表示。

9.1 专用杂凑函数 1

专用杂凑函数 1 中的 128 比特常数 T_i 定义如下：(用 16 进制表示)

$$T_0 = 1CC7086A046AFA22353AE88F3D3DACEB$$

$$T_1 = E3FA02710E491D851151CC34E4718D41$$

$$T_2 = 93987557C07B8102BA592949EB638F37$$

专用杂凑函数 1 的轮函数中用到两个常数字序列 $C_0, C_1, \cdots, C_{79}$ 和 $C'_0, C'_1, \cdots, C'_{79}$，它们定义如下：

$$C_i = K_1[0] +_{32} 00000000, (0 \leqslant i \leqslant 15),$$

$$C_i = K_1[1] +_{32} 5A827999, (16 \leqslant i \leqslant 31),$$

$$C_i = K_1[2] +_{32} 6ED9EBA1, (32 \leqslant i \leqslant 47),$$

$$C_i = K_1[3] +_{32} 8F1BBCDC, (48 \leqslant i \leqslant 63),$$

$$C_i = K_1[0] +_{32} A953FD4E, (64 \leqslant i \leqslant 79),$$

$$C'_i = K_1[1] +_{32} 50A28BE6, (0 \leqslant i \leqslant 15),$$

$$C'_i = K_1[2] +_{32} 5C4DD124, (16 \leqslant i \leqslant 31),$$

$$C'_i = K_1[3] +_{32} 6D703EF3, (32 \leqslant i \leqslant 47),$$

$$C'_i = K_1[0] +_{32} 7A6D76E9, (48 \leqslant i \leqslant 63),$$

$$C'_i = K_1[1] +_{32} 00000000, (64 \leqslant i \leqslant 79)$$

9.2 专用杂凑函数 2

专用杂凑函数 2 中的 128 比特常数 T_i 定义如下：(用 16 进制表示)

$$T_0 = FD7EC18964C36D53FC18C31B72112AAC$$

$$T_1 = 2538B78EC0E273949EE4C4457A77525C$$

$$T_2 = F5C93ED85BD65F609A7EB182A85BA181$$

专用杂凑函数 2 的轮函数中用到两个常数字序列 $C_0, C_1, \cdots, C_{63}$ 和 $C'_0, C'_1, \cdots, C'_{63}$，它们定义如下：

$$C_i = K_1[0] +_{32} 00000000, (0 \leqslant i \leqslant 15),$$

$$C_i = K_1[1] +_{32} 5A827999, (16 \leqslant i \leqslant 31),$$

$$C_i = K_1[2] +_{32} 6ED9EBA1, (32 \leqslant i \leqslant 47),$$

$$C_i = K_1[3] +_{32} 8F1BBCDC, (48 \leqslant i \leqslant 63),$$

$$C'_i = K_1[0] +_{32} 50A28BE6, (0 \leqslant i \leqslant 15),$$

$$C'_i = K_1[1] +_{32} 5C4DD124, (16 \leqslant i \leqslant 31),$$

$$C'_i = K_1[2] +_{32} 6D703EF3, (32 \leqslant i \leqslant 47),$$

$$C'_i = K_1[3] +_{32} 00000000, (48 \leqslant i \leqslant 63)$$

9.3 专用杂凑函数 3

专用杂凑函数 3 中的 128 比特常数 T_i 定义如下：(用 16 进制表示)

$$T_0 = 1D4CA39FA40417E2AE5A77B49067BBCC$$

$$T_1 = 9318AFEF5D5A5B46EFCA6BEC0E138940$$

$$T_2 = 4544209656E14F97005DAC76868E97A3$$

专用杂凑函数 3 的轮函数中用到一个常数字序列 $C_0, C_1, \cdots, C_{79}$，它定义如下：

$$C_i = K_1[0] +_{32} 5A827999, (0 \leqslant i \leqslant 19),$$

$$C_i = K_1[1] +_{32} 6ED9EBA1, (20 \leqslant i \leqslant 39),$$