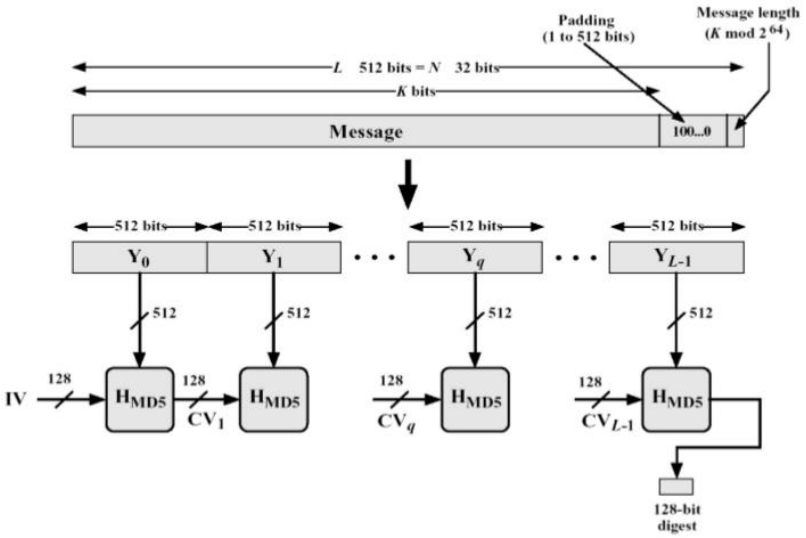
MD5 算法的程序设计和实现——实验报告

一、算法原理概述

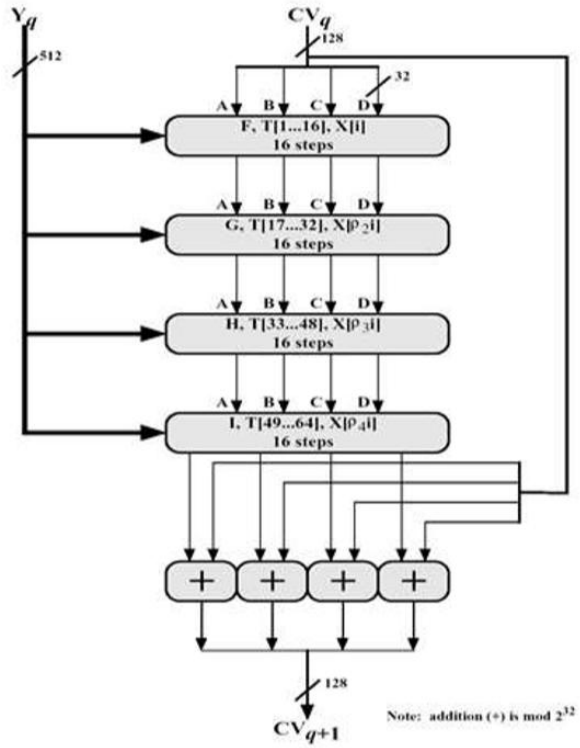
MD5 即 Message-Digest Algorithm5(信息-摘要算法 5)，由 Ron Rivest 发明，是广泛使用的 Hash 算法，用于确保信息传输的完整性和一致性。MD5 使用 little-endian(小端模式)，输入任意不定长度信息，以 512-bit 进行分组，每一分组又被划分为 16 个 32 位子分组，经过了一系列的处理后，生成四个 32-bit 的数据，最后联合输出固定 128-bit 的信息摘要。MD5 算法的基本过程为：填充、分块、缓冲区初始化、循环压缩、得出结果。

二、算法总体结构

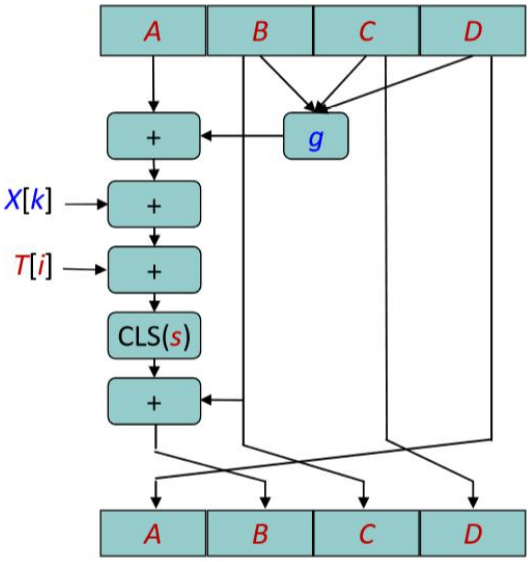
MD5 算法总流程



循环压缩总体结构



循环压缩每轮循环中一次迭代运算逻辑



4 轮循环所使用的生成函数

轮次	Function g	$g(b, c, d)$
1	$F(b, c, d)$	$(b\wedge c)\vee(\neg b\wedge d)$
2	$G(b, c, d)$	$(b\wedge d)\vee(c\wedge\neg d)$
3	$H(b, c, d)$	$b\oplus c\oplus d$
4	$I(b, c, d)$	$c\oplus(b\vee\neg d)$

三、模块分解

①填充字符串函数：在长度为 K bits 的原始消息尾部填充长度为 P bits 的标识（1000....00），其中 1 <= P <= 512（即至少需要填充一个 bit），使得填充后的消息位数为 K+P ≡ 448（mod 512）。注意，当 K ≡ 448（mod 512）时，填充的字节数 P = 512 bit。填充得到上述消息后，在尾部附加 K 值的低 64 位，最后得到一个长度为 K+P+64 ≡ 0（mod 512）的消息。

②字符串分块函数：将填充好的字符串分割成 L 个长度为 512bit 的分组

③循环压缩函数：对每个 512-bit 分组进行 64 轮迭代运算

(1) 对分组（A,B,C,D）中的 A 进行迭代运算

公式为：A <= B + ((A + g(B,C,D) + X[k] + T[i])) << S[i]

其中：

- A,B,C,D 代表 MD5 缓冲区当前的数值
- g 为轮函数，1-16 轮迭代使用 F 函数，17-32 轮迭代使用 G 函数，33-48 轮迭　代使用 H 函数，49-64 轮迭代使用 I 函数
- X[k]代表当前处理消息分组的第 k 个 32 位字，X[k]由第 n 轮迭代对应的顺序表决定
- T[i]代表 T 表的第 i 项的值，T[i] = int(2^32 * |sin(i)|)
- S[i]对应第 i 轮的左循环移位的 s 值

(2) 对分组（A,B,C,D）作循环轮换

公式为：（B,C,D,A）<=（A,B,C,D）

④MD5 编码函数：用于调用前面的功能函数进行 MD5 编码

(1) 输入待加密的明文字符串

(2) 对明文字符串进行填充

(3) 对填充后的明文字符串进行分块（Y_q）

(4) 使用预设的初始值初始化 MD5 缓冲区间（IV）

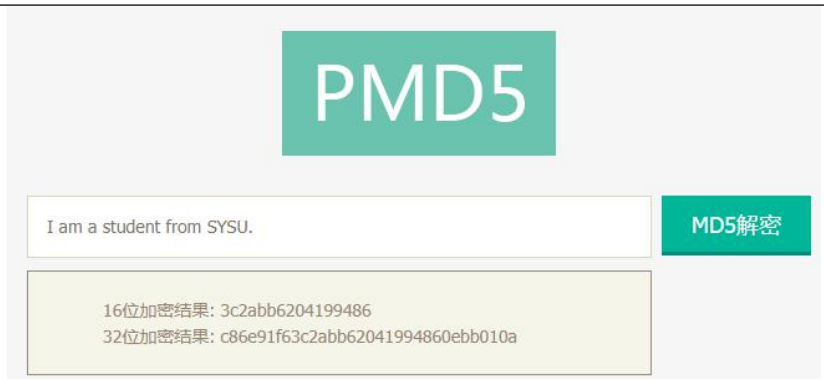
(5) 对各个分块字符串利用公式 $H_{MD5}(CV_{i-1}, Y_i)$进行循环压缩，运算结果作为下一块的输入（CV_i）

(6) 当所有的分块迭代完成后，输出结果 CV_L，L 表示最后一个分块的序号

四、数据结构

程序的输入使用 C++的 string 类型，在 MD5 的编码过程中将字符串转化成类型为 32 位 unsigned int 数组，每个分块使用长度为 16 的 unsigned int 数组，循环压缩运算过程中所有的操作都是基于 unsigned int 类型。最后输出时将 unsigned int 数组中的数据按照 16 进制格式输出成长度为 32 的字符串。

五、编译运行结果

<table>
<tr><td>本地测试</td><td> 
</td></tr>
<tr><td>网上比对</td><td> 
</td></tr>
<tr><td>本地测试</td><td> 
</td></tr>
<tr><td>网上比对</td><td> 
</td></tr>
<tr><td>结果分析</td><td>32 位加密结果输出完全一样，MD5 算法成功实现。</td></tr>
</table>