哈希函数调研

# 一、Hash函数概述

Hash函数是重要的密码元件, 它不仅可以用于数字签名方案, 还可以用于验证信息来源的真实性及信息数据的完整性, 它可将任意长度的消息压缩到固定长度的信息摘要。在设计各种密码协议的时候, 可以灵活地用它来增强安全性或达到一些特定的要求, 因而是现代密码学最重要的部分。

标准Hash函数主要分为两大类:MD系列, 包括MD4, MD5, HAVAL, RIPEMD, RIPEMD-128等和SHA系列包括SHA-1, SHA-256等。其中MD5, SHA-1是当前国际通行的两大密码标准。MD5由国际著名密码学家R.L.Rivest设计, SHA-1是由美国专门制定密码算法的标准机构--美国国家标准与技术协会与美国国家安全局设计。两大算法是目前国际电子签名及许多其它密码应用领域的关键技术, 广泛应用于金融、证券等电子商务领域。其中, SHA-1早在1994年便被美国政府采纳, 目前是美国政府广泛应用的计算机密码系统。[8]

我们小组对Hash函数的调研以MD5和SHA256为主。（目前最流行的安全加密算法）

（1）MD5信息摘要算法（MD5 Message-Digest Algorithm）是一种密码散列函数，可以产生128位的散列值（hash value），用于保证信息传输完整一致。MD5由美国密码学家Ronald Linn Rivest 设计，于1992公开。MD5由MD2，MD3和MD4发展而来，其中，由于MD2是位8位机器设计的，而MD4、MD5是面向32位电脑，故MD2的设计与MD4、MD5完全不同。

（2）SHA256是SHA-2下细分出的一种算。SHA-2，名称来自于安全散列算法2（Secure Hash Algorithm 2）的缩写，一种密码散列函数算法标准，由美国国家安全局研发，属于SHA算法之一，是SHA-1的后继者。SHA-2下又可再分为六个不同的算法标准。包括了：SHA-224、SHA-256、SHA-384、SHA-512、SHA-512/224、SHA-512/256。这些变体除了生成摘要的长度 、循环运行的次数等一些微小差异外，算法的基本结构是一致的。

# 二、Hash函数原理

## 2.1 算法概述

sha256和MD5原理基本一致，这里简单介绍MD5算法原理。

对于任意长度的信息输入，MD5都将产生一个长度为128比特的输出。这一输出即原输入报文的“报文摘要（Message Digest）”。

MD5算法中，首先需要对信息进行填充，使其字节长度对512求余的结果等于488。所以，信息的字节长度会被扩展至N\*512+488，即N\*64+56个字节（N为正整数）。填充方式为：在信息后填充一个1和无数个0，直到满足上面的条件时才停止用0对信息的填充。然后再再结果后附加一个以64位二进制表示的填充前信息长度。经过这两步的处理，信息字节长度=N\*512+488+64=(N+1)\*512，即长度恰好时512的整数倍，满足了后面处理中对信息长度的要求。[1]

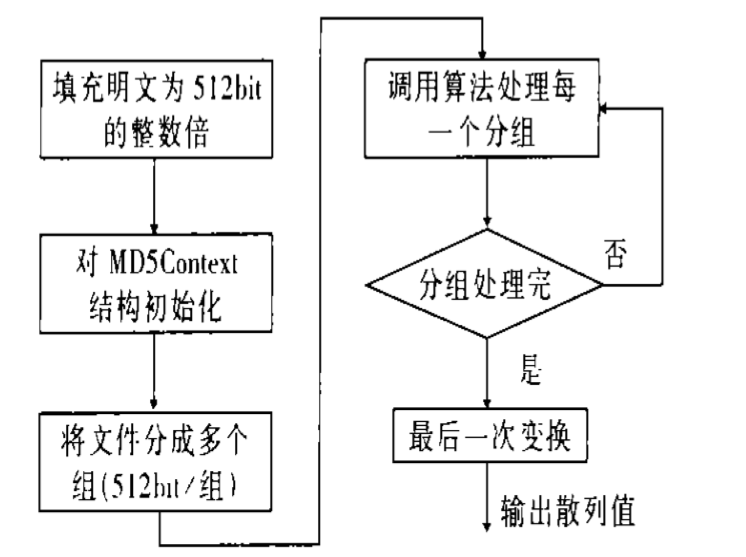


图2.1-1 MD5的实现过程[1]

## 2.2 算法详细描述

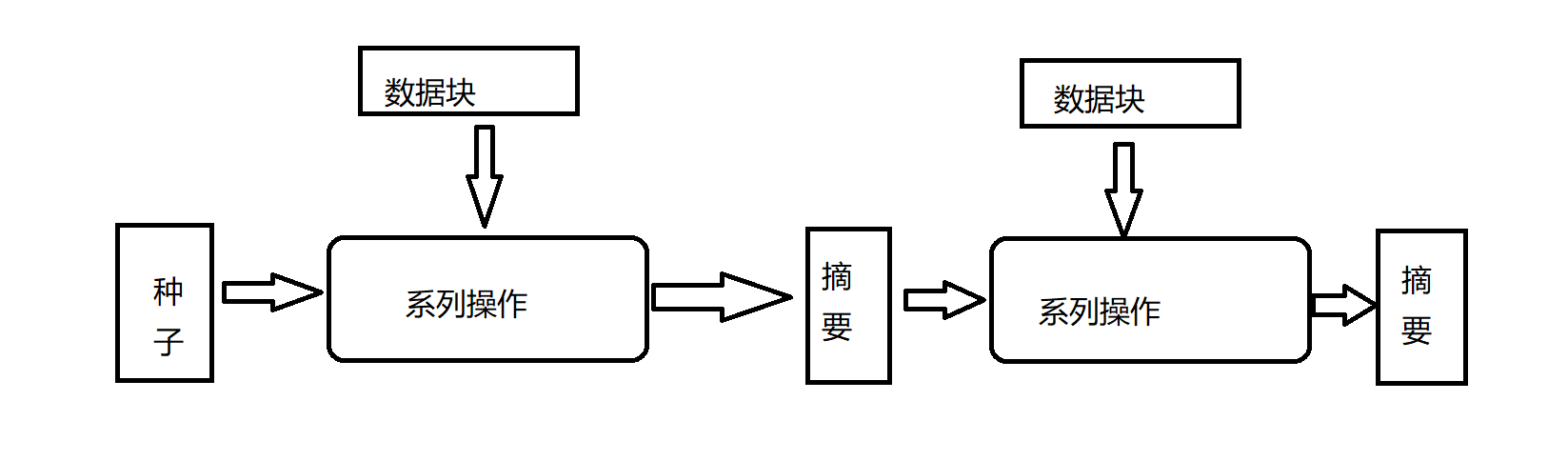


图2.2-1 MD5分组摘要算法[1]

### 2.2.1 数据准备

设输入信号为b（非负整数）位。数据为：m\_0, m\_1, …,m\_{b-1}

（1）补位

对输入的数据进行补位，使数据长度对512求余的结果是488。即数据长度为K\*512+488，即K\*64+56字节（K为整数）。即使数据长度对512求余结果已经是488，补位操作还要执行。

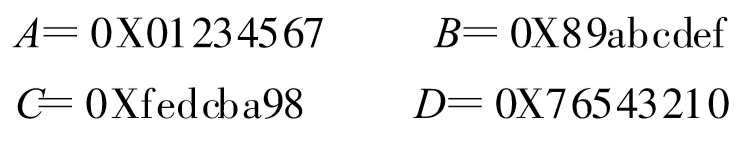
具体操作：补一个1，然后补0知道满足要求。最少补一位，最多补512位。

（2）补数据长度

将b用两个32位数表示，当b大于2^64时，数据被填补位长度位512的倍数。

（3）初始化MD缓冲器

用一个4个字的缓冲器A，B，C，D来计算报文摘要，A，B，C，D分别是32位寄存器，初始化使用十六进值表示的数字。MD5中这四个32位被称作链接变量的整数参数，分别为：[1]

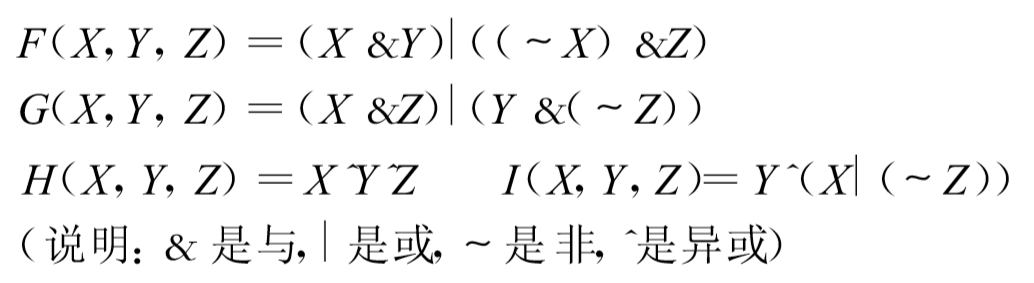


### 2.2.2 四轮循环运算

设置好四个链接变量后，就开始进入算法的四轮循环运算，循环次数是信息中512位信息分组的数目。将上面四个链接变量复制到另外四个变量中：A到a，B到b，C到c，D到d。

主循环由四轮，每轮循环都相似。第一轮进行16次操作，每次操作对a, b, c, d中的其中三个作一次非线性函数运算，然后将所得结果加上第四个变量，文本的一个子分组和一个常数。再将所得结果向右环移一个不定的数，并加上a,b,c,d中之一。最后用该结果取代a,b,c,d中之一。

每次操作中用到的四个非线性函数（每轮用一个）：



如果X，Y，Z的对应位是独立和均匀的，那么结果的每一位也是独立和均匀的。F是一个逐位运算函数，如果X，那么Y否则Z。H是逐位奇偶操作符。[1]

具体循环过程如图2.2-2 2.2-3：[2]

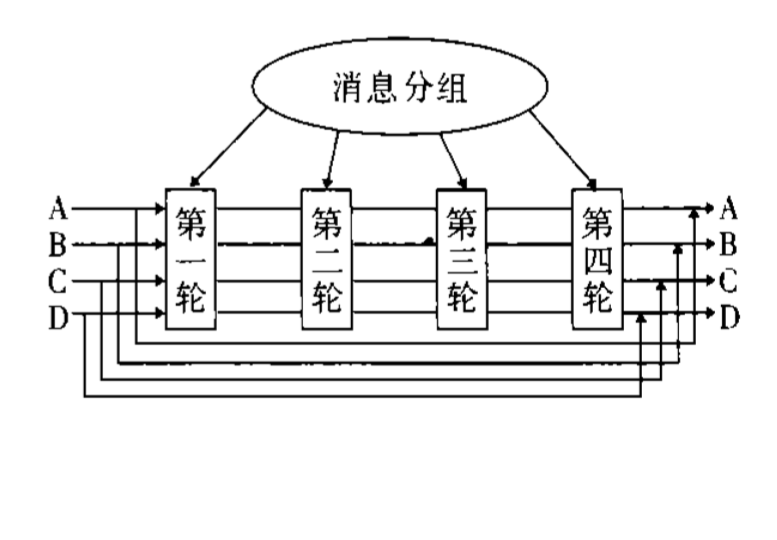


图2.2-2 MD5的主循环[2]

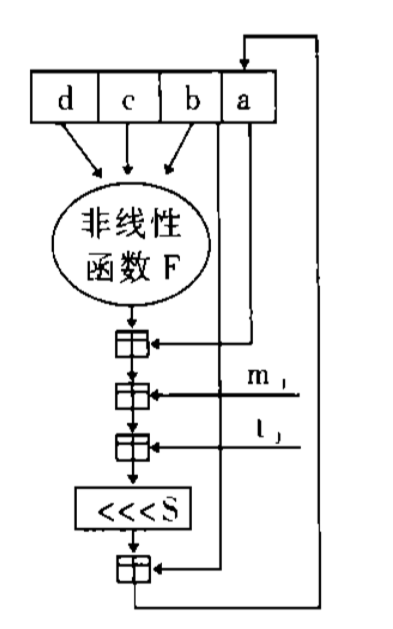


图 2.2-3 MD5的一次执行过程[2]

## 2.3 MD5 SHA256比较

安全性方面，MD5生成序列长度为128，而SHA256生成序列长度为256，长度越长越不易产生碰撞，所以SHA256的安全性更高；在效率方面，SHA256耗时更长。

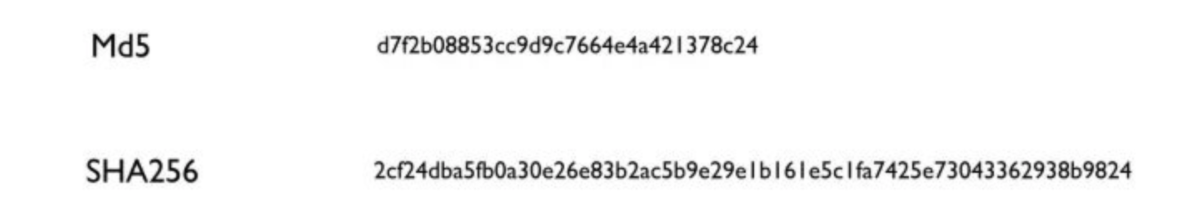


图2.3-1 两种哈希生成的序列对比

# 三、应用

## 3.1 消息完整性认证

Hash的典型应用是对一段信息产生信息摘要，以防止被篡改。比如再UNIX下右很多软件在下载时都有一个文件名相同，文件扩展名为.md5的文件，在这个文件中通常只有一行文本结构如：

MD5 ( tanajiya. tar. gz) = 0ca175b9c0f726a831d895 e269332461

这就是该文件的数字签名。MD5将整个文件当作一个大文本信息，通过其不可逆的字符串变换算法，产生了这个唯一的MD5信息摘要。在以后传播这个文件的过程中，若文件内容改变，只要对文件重新计算MD5时就会发现信息摘要不对，就能确定得到了一个不正确的文件。若还有第三方认证机构，用MD5能防止文件作者“抵赖”，这就是数字签名的应用。[1]

## 3.2 用户密钥管理

Hash函数被广泛应用于加密技术。加密方式分为单向加密和双向加密，MD5和SHA256属于单向加密，双向机密算法是机密算法中最常用的，可以加密和解密。但单向加密算法只能加密不能解密，这种特性可以应用在用户密钥管理中。[3]

大多数系统软件和应用软件中，用户口令是用DES或3DES算法加密后存放在机器中的。由于DES算法对被加密的数据的长度有一定的要求，因而一般的系统对用户密码有限制，即用户不能输入过长的密钥，给系统安全造成隐患。同时，由于计算能力的提高，DES的56bit密钥已经难以满足用的的需求，所以引入MD5的使用。[2]

用户密码以Hash算法经加密后存储在文件系统中。当用户登陆的时候，系统把用户输入的密码计算成hash值。然后再去和保存在文件系统中的hash值进行比较，进而确定输入密码是否正确。这样系统在不知道用户密码明码的情况下就能确定用户登陆系统的合法性。

这种方法不但在一定程度上增加密码被破解的难度，还可以避免用户的密码被具有系统管理员权限的用户知道。[1]

## 3.3 密码学中的应用

（1）保密增强邮件PEM（Privacy Enhanced Mail）是因特网保密性增强邮件标准。它提供的消息完整性检测中的鉴别使用的单向散列函数是MD2和MD5。

（2）由 Philip Zimmermann 设计的著名的邮件加密软件PGP( Pretty Good Privacy)的 2. 6. 3 版本中使用的单向散列函数是 MD5。[2]

# 四、改进

## 4.1 MD5改进

由于MD5算法的密码和加密后的密文是一一对应的，黑客可以收集密文和对应的密码形成一张表。当获取了数据库的密文之后，通过枚举法匹配密文即可推出相应的密码。密文和密码对应表（彩虹表）如图所示。[4]

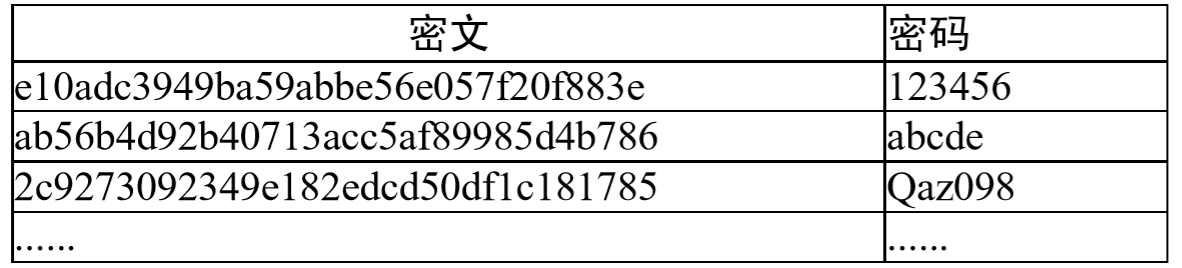


图 4.1-1密文密码对应表[4]

所以我们需要对MD5进行改进增强其安全性。

### 4.1.1 MD5加随机数算法

MD5加随机数算法也称MD5加盐算法。在密码加上随机数之后再进行MD5运算，最终的MD5值就会不一样，这个随机数也会保存在数据库中，随机数被称为Salt值。有时为了减轻开发压力，程序员会统一使用一个Salt值，存储在某个地方，而不是每个用户都生成私有的Salt值。

下图为MD5加随机数算法流程。[4]

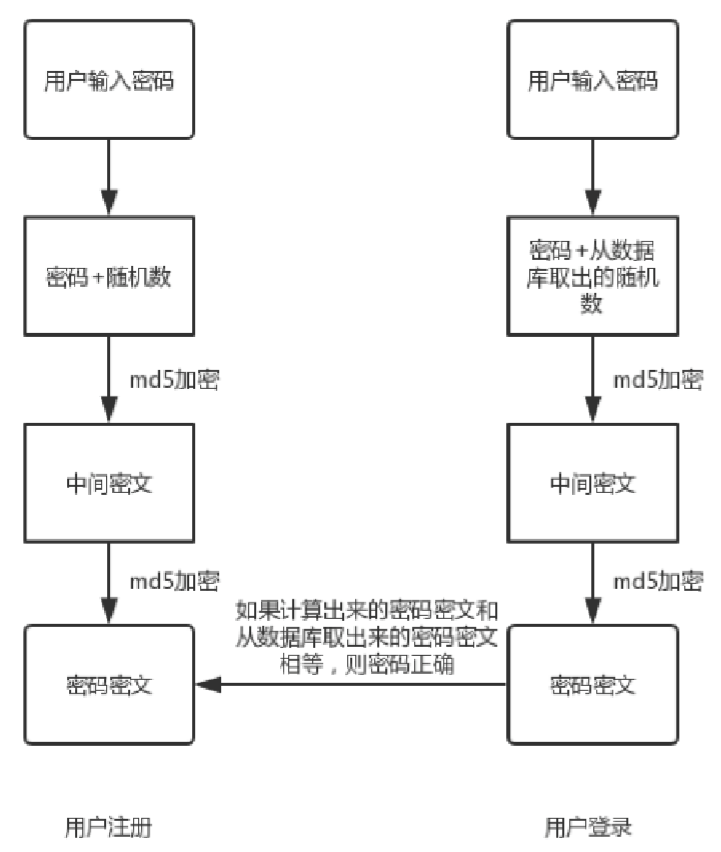


图 4.1-2 MD5加随机数算法流程图[4]

用户注册：

1）用户输入账号密码以及其他用户信息；

2）系统为用户生成salt值；

3）系统将salt值和用户密码连接到一起；

4）对连接后的值进行散列（md5加密），得到Hash值，即密码密文；

5）将Hash值和Salt值分别存放到不同的数据库中。

用户登陆：

1）用户输入账号和密码；

2）系统通过用户名找到与之对应的Hash值和Salt值；

3）系统将Salt值和用户输入的密码连接到一起；

4）对连接后的值进行散列，得到Hash值；

5）对比数据库中存储的Hash值和运算得到的Hash值是否相等，相等则表示密码正确，否则表示密码错误。

### 4.1.2与其他加密算法混合使用

比如，MD5和Base64混合的加密算法，将名文MD5加密得到32位16进制密文，再将32位密文拆分成16个2位16进制数组，并转换成对应的二进制数，最后将16分二进制数组链接成一个128位的2进制数，采用Base64加密原理进行加密。该算法可以避免查询MD5散列值字典获取用户明文密码，更加有效保证了用户的密码安全。[5]

## 4.2 SHA256改进

由于MD5和SHA256都为单向哈希算法，上文提到对MD5算法的改进，对SHA256同样适用。另外，为提高SHA256的效率，以及降低SHA256的成本，有以下几种方法。

### 4.2.1 基于硬件的快速实现

为满足应用对算法运算速度的要求，有一种快速实现SHA-256安全散列算法的的硬件新结构。该方法通过对算法进行推导，使得能在硬件上实现平衡的四级流水线设计，提高了运算速度.综合结果表明,采用中芯国际0.18μm工艺，数据吞吐量可达2.89 Gb/s，优于目前发表的同类设计结果。[6]

### 4.2.2 高频率、低成本的SHA-256算法VLSI实现

SHA-256安全散列算法广泛应用于数据完整性校验及数字签名等领域。为满足安全SoC系统对SHA-256高工作频率和低硬件成本的设计需求，提出了一种新颖的SHA-256 VLSI实现方法，通过分解算法实现步骤，进而缩短关键路径，节省硬件资源。

传统的SHA-256硬件实现电路中, 需要用到多组寄存器阵列, 以保存中间计算结果及消息摘要, 消耗了大量的硬件资源, 越来越不适于大规模SoC系统的低硬件成本发展需求。为满足高工作频率以及面积敏感的安全SoC系统设计要求，对基本的SHA-256计算过程进行分解, 以实现寄存器、加法器的共享, 缩短关键路径；为节省面积, 传统的寄存器阵列用Register File代替, 算法实现的硬件架构相应地做出变更。

采用SMIC 0.13μm CMOS工艺综合实现，结果表明其最高工作频率达334.5MHz，资源消耗减少了70%。[7]

**参考文献：**

[1]张裔智,赵毅,汤小斌.MD5算法研究[J].计算机科学,2008(07):295-297.

[2]辛运帏,廖大春,卢桂章.单向散列函数的原理、实现和在密码学中的应用[J].计算机应用研究,2002(02):25-27.

[3]李霞.MD5加密算法浅析及应用[J].运城学院学报,2005(05):56-57.

[4]曾诗亮,周琪云.MD5加随机数算法的研究与应用[J].网络安全技术与应用,2019(09):26-27.

[5]罗江华.基于MD5与Base64的混合加密算法[J].计算机应用,2012,32(S1):47-49.

[6]陈华锋.高速SHA-256算法硬件实现[J].浙江大学学报(理学版),2009,36(06):675-678.

[7]应建华,罗柳平.高频率、低成本的SHA-256算法VLSI实现[J].微电子学与计算机,2011,28(05):28-31.

[8]柯品惠,郑秋鸿.Hash函数研究综述[J].福建电脑,2008(12):1-4+35.