目 录

[1. 符号与约定 1](#_Toc193663624)

[1.1. 比特、字节和子块顺序 1](#_Toc193663625)

[1.2. 运算符列表 1](#_Toc193663626)

[2. 算法编制描述 2](#_Toc193663627)

[2.1. 上层结构 2](#_Toc193663628)

[2.1.1. 填充分割消息 2](#_Toc193663629)

[2.1.2. 初始IV 3](#_Toc193663630)

[2.1.3. 输出变换 3](#_Toc193663631)

[2.2. 压缩函数 3](#_Toc193663632)

[2.2.1. 轮函数E 4](#_Toc193663634)

[2.2.2. 轮常数加 4](#_Toc193663635)

[2.2.3. 字节代换 4](#_Toc193663636)

[2.2.4. 列位移 5](#_Toc193663637)

[2.2.5. 行混淆 5](#_Toc193663638)

[2.2.6. 道变换 5](#_Toc193663639)

[2.2.7. S盒及其性质 5](#_Toc193663640)

[2.3. 杂凑样本数据 7](#_Toc193663641)

[2.3.1. 第一组样本数据 7](#_Toc193663642)

[2.3.2. 第二组样本数据 7](#_Toc193663643)

[2.3.3. 第三组样本数据 7](#_Toc193663644)

[3. 活跃S盒测试结果 7](#_Toc193663645)

**杂凑密码算法描述**

# 符号与约定

## 比特、字节和子块顺序

所有数据变量用最高比特（或子块）在左边，最低比特（或子块）在右边表示，而且从左到右从0开始编号。例如，一个n比特数从左到右分别称为它的第0比特、第1比特、第2比特、…、第n-1比特。当一个变量被分成几个子块时，最左边（最高）子块由原始数据的最高部分组成，等等一直到最低。

当密码算法的输入（或输出）是字节流时，我们按地址小的是高位（即左边），地址大的是低位（即右边）的顺序把每4（或2）个字节组成一个32（或16）比特字。32（或16）比特字之间的顺序是先左后右，即存储时，左边地址小，右边地址大。

下面举例说明。根据以上规定，认为二进制表示的64比特数：

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

与如下4种16进制表示的64比特数互相等价：

0x0123456789abcdef

0x01234567||0x89abcdef

0x0123||0x4567||0x89ab||0xcdef

0x01||0x23||0x45||0x67||0x89||0xab||0xcd||0xef

## 运算符列表

0x 用前缀0x表示16进制数；

= 赋值，也可表示相等；

\* 表示矩阵乘法；

~ 逐比特取反；

# 表示集合中元素的数量；

i mod n 表示整数i除以n的余数，0≤ i mod n≤n-1；

|| 字符串连接；

|S| 表示字符串S的长度；

◦ 变换复合，即对于变换f1、f2和变量x：f2◦f1(x)= f2(f1(x))；

MSBn(S) 表示截取比特串S的高位n比特。

: 操作数循环左移n比特；

: 第i个消息块；

第i次迭代之后算法的内部状态。

算法内部状态A里位于(x,y,z)的元素。

一个字节的异或运算。

首对齐的异或运算。

尾对齐的异或运算。

AB 将运算A复合到运算B上。

使内部状态 x 经过 S 盒S。

置换。

压缩函数。

# 算法编制描述

本算法是杂凑密码算法，杂凑摘要长度为64字节。

## 上层结构

本算法的上层结构统一采用JH压缩函数结合MD迭代结构，如图2-1。JH压缩函数是由双射函数（具有固定密钥的分组密码）构建而成的。该分组密码的块大小为S字节。​在压缩函数中，S字节的哈希值H(i−1)与n字节的消息块M(i)被压缩成S比特的H(i)。这样的压缩一共会重复进行r次，也即轮数为r。​消息摘要的大小最多为n字节。本算法各项参数的取值如下表所示：

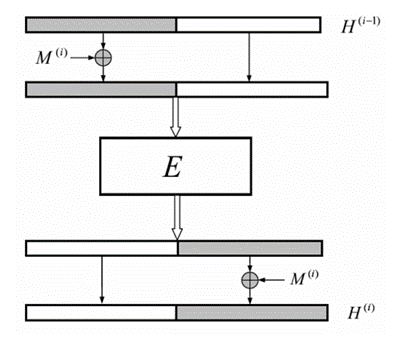


图2-1 JH压缩函数结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | n | S | m | r |
| hash-64 | 64 | 128 | 64 | 20 |

### 填充分割消息

首先对*l*比特消息M进行填充，先在其后面联接1位的“1”，然后联接最少个数的位“0”，使得填充后的消息Mpad的长度为S的倍数。即：

Mpad=M||1||0m-1-(*l* mod m)

然后将其分割为若干个128字节的消息块。

### 初始IV

记函数的初始状态为IV，即H。。如图2-2，IV表示为一个长宽高为4\*4\*8的三维立方体。

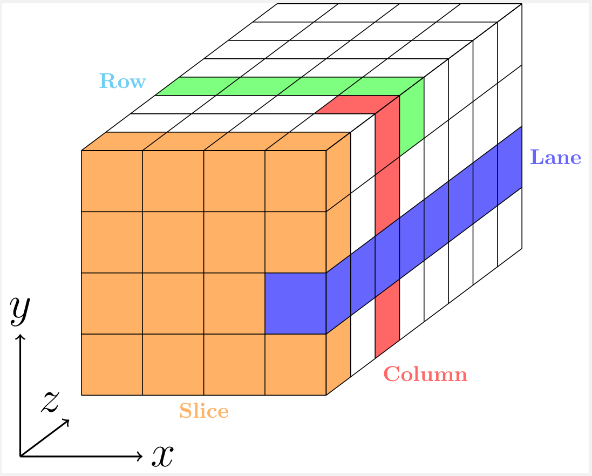


图2-2 IV的结构

其中x,y,z坐标均从0开始。x轴和y轴张成的面成为切片（slice），每个切片根据Z坐标取值初始化为以下对应的矩阵：

### 输出变换

输出变换定义如下：

。

其中是一个截断操作，表示将x中的前n位保留，其余部分丢弃。需要注意的是，由于使用置换函数作为压缩函数，所以。

## 压缩函数

JH结构压缩函数非常简单，其核心是一个{0,1}t→{0,1}t的置换P。P的设计采用了分组密码的设计思想，其结构类似广义Feistel结构.记当前轮数为i，1≤i≤r，输入状态为H(i-1)，满足|H(i-1)|=l。将H(i-1)分为2部分，每部分64字节，即H(i-1)= h0(i-1)|| h1(i-1).压缩函数计算流程如下：

1. 计算h0(i-1) = h0(i-1) ⊕M(i)
2. 计算H(i-1)=E(H(i-1) )
3. 计算H(i)=h0||(h1(i-1) ⊕M(i )

最后，压缩函数返回H(i)，作为一轮压缩的输出。



### 轮函数E

图2-3给出了轮函数y=E(x)的图示。

y=E(x)的输入是一个128字节的状态x，输出是一个128字节的状态y，y=F(x)=E(D(C(B(A(x)))))，即F=A◦B◦C◦D◦E，其中，A、B、C、D、E分别是轮常数加(AddConstant)、字节代换(SubBytes)、列位移(ShiftColumn)、行混淆(MixRow)、道变换(LaneTransform)

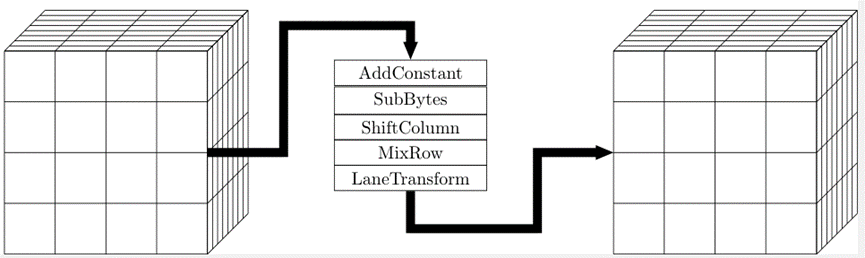


图2-3 轮函数E

### 轮常数加

轮常数加在每个字上进行，定义为：

其中c表示轮常数，其表示如下：

。

### 字节代换

字节替代是杂凑函数中唯一的非线性变换，定义为：

其中S表示的是应用S盒映射。

### 列位移

列移位是在每一个 Slice 上进行的，定义为：

其中表示的是每一列所要循环移位的参数。其表示如下：

### 行混淆

行混淆参考Keccak 的设计思路，对不同维度的行进行混淆。将MDS 矩阵记为𝑀，那么行混淆操作定义为:

其中M为一个4×4的MDS矩阵。其表示如下

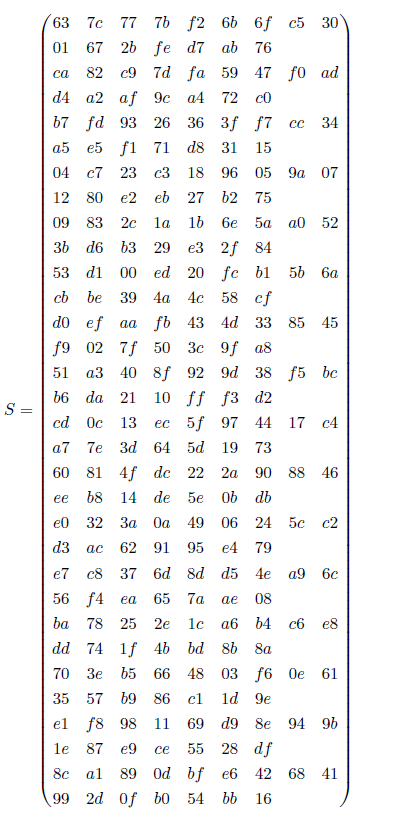
### 道变换

在道上做循环移位操作，定义为：

其中表示每一个道的循环移位参数。移位参数同样使用类似于列移位使用自动化搜索技术获得。其表示如下：

### S盒及其性质

S 盒中的每个元素是一个十六进制整数。其表示如下：



选用了与 AES加密算法中相同的 S 盒。选择这个变换的原因如下：

* **大小：**8 位 S 盒在实现方面是方便的权衡选择（在流行平台上是最小的字长），同时也考虑了密码分析方面的因素。另一方面，可以选择$2^8!$种不同的置换。
* **单一：** S 盒而不是许多不同的 S 盒：这再次是实现和密码分析方面的权衡选择。
* **没有随机 S 盒**：结构化的 S 盒 比随机 S 盒 允许更高效的硬件实现。
* 由于 S 盒是从 AES 继承而来，实现方面的特点（特别是在硬件方面）得到了深入研究。

## 杂凑样本数据

### 第一组样本数据

消息：

0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00

杂凑值:

0xfea147263fe660d2dd53bcebb69f484aa9474e130446a8a99767342f0e9b6655feb127b60ae87980f321d36777adb2469430ab8ef4fc5bd851ec82a44ce59ad6

### 第二组样本数据

消息：0x00，0x01，0x02，0x03，0x04，0x05，0x06，0x07，0x08，0x09，0x0a，0x0b，0x0c，0x0d，0x0e，0x0f

杂凑值：

0x2bdbe460c36f0f4fe204e139802c8e1a09203027afde4eafdb9c66738c60a250e6269b05d6cba3934d9220526365077ca7b72171a97c4d3fd2f0f3e6c1bf7cd5

### 第三组样本数据

消息：0x00，0x00，0x00，0x00

杂凑值：

0xfea147263fe660d2dd53bcebb69f484aa9474e130446a8a99767342f0e9b6655feb127b60ae87980f321d36777adb2469430ab8ef4fc5bd851ec82a44ce59ad6

# 活跃S盒测试结果

|  |  |
| --- | --- |
| 运行轮数 | 活跃S盒个数 |
| 1 | 1 |
| 2 | 5 |
| 3 | 9 |
| 4 | 25 |
| 5 | 41 |
| 6 | 50 |
| 7 | 57 |
| 8 | 65 |
| 9 | 69 |
| 10 | 85 |
| 11 | 93 |
| 12 | 105 |
| 13 | 111 |
| 14 | 120 |
| 15 | 128 |
| 16 | 140 |
| 17 | 146 |
| 18 | 155 |
| 19 | 163 |
| 20 | 173 |