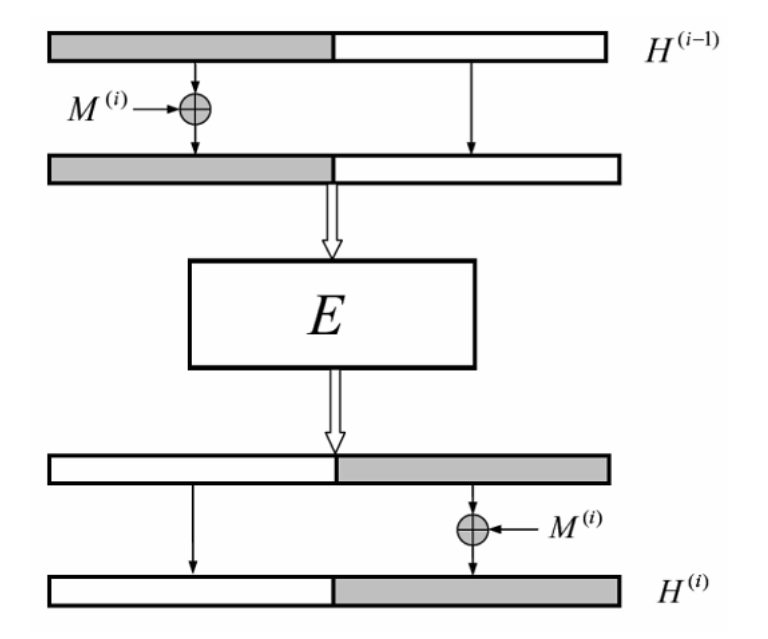
**5一种面向字的杂凑密码设计方案**

* 1. **整体结构和压缩函数**

杂凑函数是由压缩函数迭代构成的。首先对消息进行填充，填充后获得 比特倍数的消息，将其分割为若干个比特的消息块。初始值 是一个长度为比特，处理消息块过程为：

因此，压缩函数将一个长度为1024比特和一个长度为512比特的输入映射到一个1024比特的输出。处理完消息块后对最后一个链接块 做输出变换得到最终的哈希值。输出变换记为，因此。输出变换：输出变换定义为：。其中是一个截断操作，表示将中的前比特保留，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。

压缩函数是一个双射函数，其中是一个置换



定义为：

Generate\_hash函数：

函数功能：用于生成给定参数的杂凑密码；

函数的输入如下：

Len\_hash：杂凑值的长度（比特）；

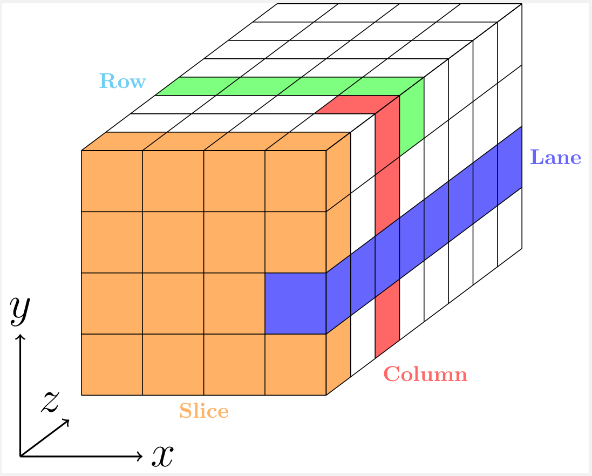
函数返回：给定参数的杂凑密码。

子模块函数：

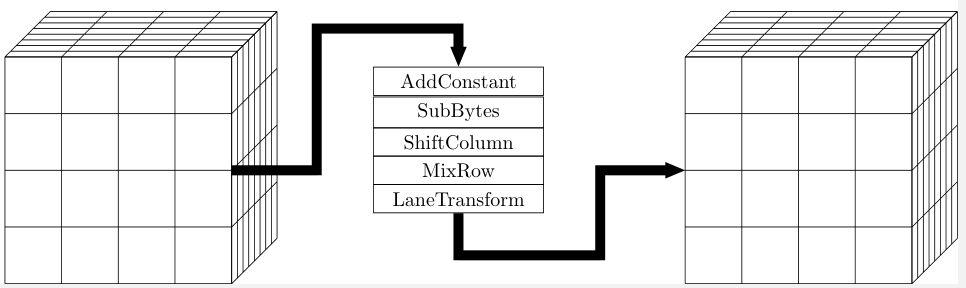
Generate\_permutation函数：生成用于杂凑密码压缩函数中使用的置换函数

**5.2置换函数**

置换函数的状态设置为 构成的立方体，每个方块大小为 比特，总体状态大小为 比特。将整个状态记为，那么()代表其中一个字节。状态中橙色的Slice用()来表示，绿色的Row用表示，红色的Column用表示，蓝色的Lane用表示。



置换由以下五个组件构成，分别是轮常数加、字节替代（S盒）、列移位、行混淆、Lane变换，我们将它们分别记为轮常数加：、字节替代：、列移位：、行混淆：和Lane变换：。因此输入状态经过置换得到输出定义为：

如下图所示：

Generate\_permutation函数：

函数功能：用于生成杂凑密码所需的置换；

函数返回：用于生成杂凑密码所需的置换。

子模块函数：

GenerateNonlinearlay：返回目前已有的性质良好的8比特S盒；

GenerateLinearlayer：返回所需要的线性组件；

GenerateRoundconstant：返回置换中所需的轮常数。

**5.2.1非线性组件**

变换是杂凑函数中唯一的非线性变换，定义为

选用了与ZUC杂凑函数中相同的S盒。

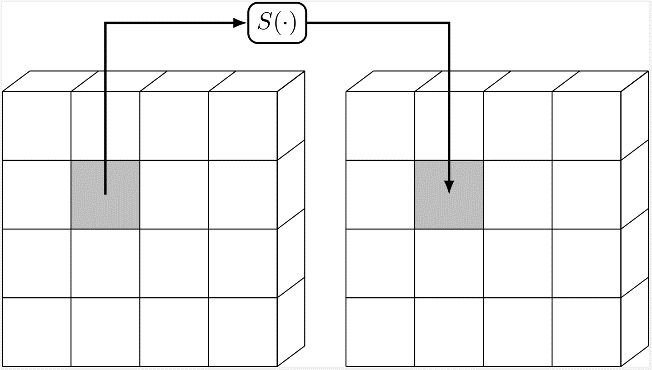
选择这个变换的原因如下：

• 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 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ZUC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。



GenerateNonlinearlay函数：

函数功能：选择目前已有的性质良好的S盒；

函数返回：返回目前已有的性质良好的S盒。

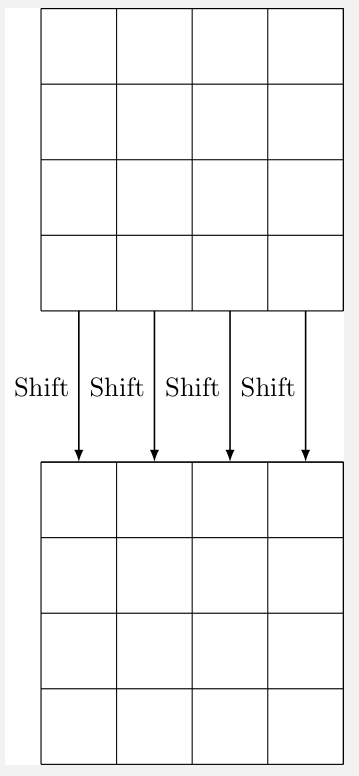
**5.2.2线性组件**

**列移位：**

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。

为了搜索最佳扩散的移位参数，使用自动化搜索技术来搜索最佳扩散的移位参数，将状态扩散之间的约束转化为MILP问题或者SAT问题，设置最大扩散数，或者设置目标函数使其求取最大扩散数，再对模型进行求解来获得最佳扩散的移位参数。



**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，那么行混淆操作定义为

行混淆变换的主要设计目标是遵循广泛的传播策略。同样使用自动化搜索技术来搜索出使得分支数最大的MDS矩阵，将行状态之间的约束以及MDS矩阵的计算量进行约束，之后将搜索MDS矩阵转化为MILP问题或者SAT问题，设置目标函数为最大差分、线性分支数或者将最大差分、线性分支数设置为约束，接着对模型进行求解来获得最大分支数的MDS矩阵参数。

**Lane变换**

在Lane上做循环移位操作，定义为：

其中表示每一个Lane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。

GenerateLinearlayer函数：

函数功能：生成所需要的线性组件；

函数返回：生成所需要的线性组件。

子模块函数：

ColumnShift：生成用于列移位的参数；

RowMix：生成用于行混淆的MDS矩阵；

LaneTransformation：生成用于Lane变换的参数。

**5.3 安全性评估**

**5.3.1碰撞分析**

杂凑密码设计的一个关键目标是抵抗碰撞分析，即防止找到两个不同的输入，它们产生相同的杂凑值。

1. 标准碰撞：这是最基本的碰撞类型，在这种情况下，分析者寻找两个不同的消息 和 ，使得它们在相同的初始向量（IV）下产生相同的杂凑值。
2. 半自由启动碰撞：在这种攻击中，分析者拥有更多的灵活性。他们可以选择一个不同于标准 IV 的初始值 ，并找到两个消息 和 ，使得 。这种攻击对杂凑密码的安全性构成更大的威胁，因为它表明即使在变更初始状态的情况下，也可能找到碰撞。
3. 自由启动碰撞：这是最灵活的碰撞分析形式。分析者寻找两对不同的初始值和消息 和 ，使得 。这种分析显示了杂凑密码在面对更复杂场景时的潜在脆弱性。

碰撞分析评估模块：根据不同参数生成的杂凑密码，我们定义Collision函数来评估其抵抗碰撞分析的能力，输入为生成的杂凑密码算法C或Python代码，输出其抵抗碰撞分析的能力，如碰撞分析轮数，复杂度等。

**5.3.2原像分析与第二原像分析**

在原像分析中，分析者试图找到一个消息，它的杂凑值与给定的杂凑值相匹配。这种攻击针对的是杂凑密码的基本属性，特别是其原像抗性。原像分析的成功意味着杂凑密码的安全性受到威胁。

**完整原像分析**：目标是对于一个特定的杂凑值，找到任意一个消息，使得其杂凑值与给定值相匹配。完整原像分析评估模块：根据不同参数生成的杂凑密码，我们定义FullPreimage函数来评估其抵抗完整原像分析的能力，输入为生成的杂凑密码算法C或Python代码，输出其抵抗完整原像分析的能力，如完整原像分析轮数，复杂度等。

**部分原像分析**：在这种分析中，只需要匹配杂凑值的一部分。这通常比完整原像分析更容易。部分原像分析评估模块：根据不同参数生成的杂凑密码，我们定义PartialPreimage函数来评估其抵抗部分原像分析的能力，输入为生成的杂凑密码算法C或Python代码，输出其抵抗部分原像分析的能力，如部分原像分析轮数，复杂度等。

**第二原像分析**：在此分析中，分析者找到一个与已知输入不同的输入，但这两个不同的输入产生相同的杂凑值。部分原像分析评估模块：根据不同参数生成的杂凑密码，我们定义PartialPreimage函数来评估其抵抗部分原像分析的能力，输入为生成的杂凑密码算法C或Python代码，输出其抵抗部分原像分析的能力，如部分原像分析轮数，复杂度等。

**5关键技术分析**

**怎么样设计合适的MDS矩阵。**

现有的构造 MDS 矩阵的方法可概括为 3 种. 一是从有限域上的 MDS 码中提取 MDS 矩阵。二是从循环矩阵, 如 Cauchy 矩阵和 Hadamard 矩阵等特殊矩阵中搜索 MDS 矩阵。三是利用迭代的方法计算 MDS 矩阵。

本项目选择从循环矩阵中搜索 MDS 矩阵，在选择最轻量级的循环矩阵并检查其MDS属性的方法中，主要存在两个挑战。首先，对于一个阶数为的通用循环矩阵（不考虑条目的值），，有种排列条目的方式，这可能很快变得难以处理。其次，选择个轻量级非零条目的方式并不需要是独特的，这可能导致搜索空间远大于仅选择个不同条目并排列它们。

首先引入一个等价关系来将个循环矩阵划分为等价类，其中同一等价类内的循环矩阵共享相同的分支数。这使我们能够通过检查每个等价类中的一个代表来减少搜索空间。接下来我们分析了循环结构，并展示了对于阶数，最多有5种类型的MDS循环矩阵，即其第一行有个不同条目的循环矩阵，1、2或3对重复条目，或3个重复条目。这表明任何MDS循环矩阵必须属于这5种类型之一。本项目主要关注循环矩阵的一个特殊情况，即左循环矩阵。通过利用矩阵的性质，包括排列等价关系，来搜索MDS左循环矩阵。为了有效地判断一个左循环矩阵是否为 MDS，我们提前收集其所有子矩阵的行列式的符号表达式，并使用它们来计算行列式的值。一旦检测到行列式值为 0，就可以确认该矩阵不是 MDS；否则，它就是 MDS。

一般构造 MDS 矩阵的方法依赖于程序的遍历, 因此确定一个较小的可行解空间可极大地节约遍历耗时并提高搜索得 MDS 矩阵的概率。对于嵌套式 MDS 矩阵的构造, 若能确定各嵌套元素间的代数关系, 则搜索算法的时间复杂度甚至可以指数级地得到消减。考虑到 MDS 矩阵实际上就是任意阶子矩阵均满秩的矩阵, 一个可行的策略是使得所有子矩阵的行列式都是关于某个嵌套矩阵元素 的多项式, 只要这些多项式唯一分解后的因式不含 的极小多项式, 则这样的矩阵必为 MDS 矩阵。要达成上述目标同时使得构造方法更具一般性, 不妨考虑 MDS 矩阵中的其他元素, 设为, 令是的 某个幂次。即 。则这样构造的 MDS 矩阵中各级子矩阵的行列式都是关于 的某个指数多项式. 为尽量扩充参数的可选空间, 使 取上阶最大为15 的元素。

通过引入形式化 MDS 矩阵的概念，一个矩阵被称作是形式化 MDS 矩阵, 若它满足以下两个条件:

(1) 构成的元素部分已知, 且它们在中的位置确定;

(2) 满足在一个嵌套式 MDS 矩阵中, 单位矩阵的个数至多为9 且至少有 3 个元素不同。

简而言之, 形式化 MDS 矩阵就是满足以上条件但只有部分构成元素已知的矩阵。为获得异或总数尽可能少的 MDS 矩阵，可以进一步限制部分元素为单位矩阵，通过结合MILP、SAT等自动化搜索技术来搜索满足要求的MDS矩阵。

**MDS矩阵的设计准则：**

1. 安全性方面：在安全性方面需要考虑差分、线性分支数

设 ，是上的的矩阵，则的差分、线性分支数定义为：

其中为的转置，计算输入中比特分块非零个数。

1. 对于可逆矩阵，其差分分支数达到最大值，等价于线性分支数

达到最大值。

1. 当差分分支数达到最大值时，则称为MDS矩阵，L为MDS变换。
2. 效率方面：在效率方面需要考虑实现效率，称为线性扩散层的计算量评估，我们将矩阵中的 矩阵记为 表示为：

记。则的计算量定义为：。