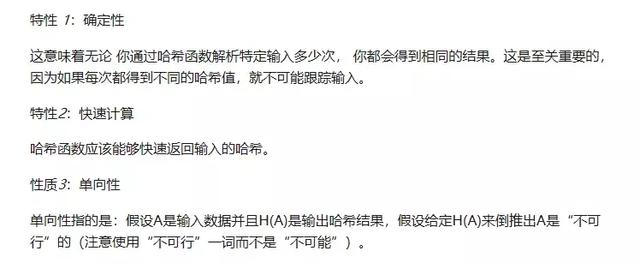
## 什么是HASH（哈希）

简单来说，哈希意味着输入任意长度的字符串通过密码运算来实现固定长度的输出。

## 加密哈希算法特性



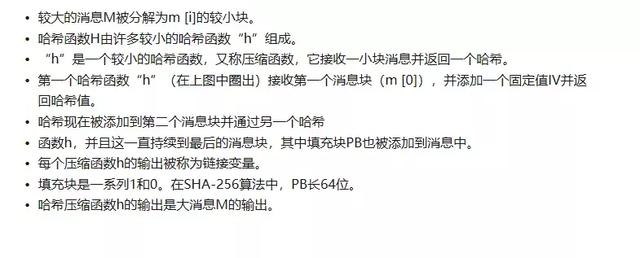
## 生日悖论

如果你在街上碰到任何陌生人，那么你们两个都有相同的生日的可能性非常低。事实上，假设一年中的所有日子都有生日的可能性，另一个人分享你的生日的机会是1/365，这是0.27％：概率非常低。

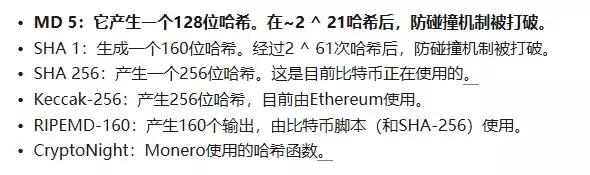
不过，如果你在一个房间里聚集20-30人，那么两个分享完全相同生日的人的几率会大幅度上升。

**什么是Merkle-Damgard结构？**

该结构非常简单，并且遵循以下原则：给定短消息的防碰撞哈希函数，我们可以为长消息构造防碰撞哈希函数。



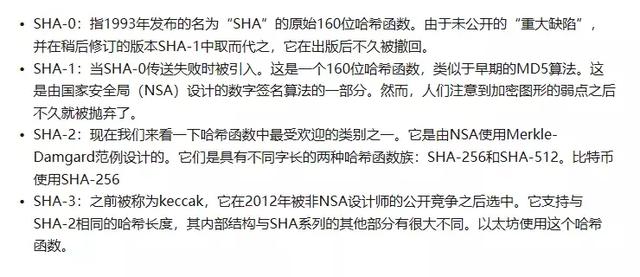
## 常用的加密哈希函数



## 安全哈希算法(SHA)

安全哈希算法是由美国国家标准与技术研究院（NIST）发布的美国联邦信息处理标准（FIPS）的一系列加密哈希函数。

SHA由以下算法组成：



# SHA-256

SHA-256是一个SHA-2函数，它使用32个单词，而不是使用64位单词的SHA-512。比特币在以下两种情况下使用SHA-256：

使用场景:采矿、创建地址

**采矿:**

比特币采矿涉及矿工解决复杂的计算难题，以找到一个块，然后将其附加到比特币区块链。就是我们经常称的工作量证明（proof-of-work），它涉及SHA-256哈希函数的计算。

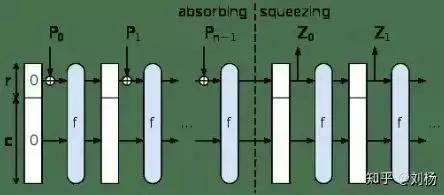
**创建地址:**

SHA-256哈希函数用于哈希比特币公钥以生成公共地址。哈希密钥为身份认证增加了一层额外的保护。此外，哈希地址的长度仅比存储更好的比特币公钥更短。

# SHA-3**:**

这种算法以前被称为keccak，并被以太坊使用。它是在non-NSA设计师的公开竞赛之后创建的。SHA-3使用“海绵（sponge）机制”。

## 什么是海绵机制



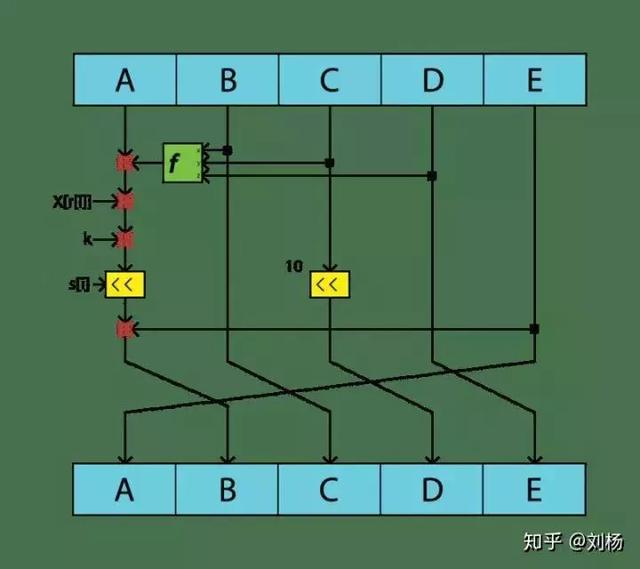
海绵功能是具有有限内部状态的一类算法，其获取任意长度的输入比特流并产生预定长度的输出比特流。

# RIPEMD-160哈希函数

RIPEMD是由比利时鲁汶的Hans Dobbertin，Antoon Bosselaers和Bart Preneel在鲁汶Katholieke大学的COSIC研究小组开发的一系列加密哈希函数，并于1996年首次发布。

虽然RIPEMD基于MD4的设计原则，但其性能与SHA-1非常相似。RIPEMD-160是这种哈希函数的160位版本，通常用于生成比特币地址。

比特币公钥首先通过SHA-256哈希函数运行，然后通过RIPEMD-160运行。这样做的原因是因为160位的输出比256位小很多，这有助于节省空间。



# CryptoNight哈希函数

现在我们拥有由Monero使用的CryptoNight哈希函数。与比特币不同，Monero希望他们的采矿尽可能地不利于GPU。他们可以做到这一点的唯一方法是让他们的哈希算法难以记忆。

CryptoNight是一种内存硬件哈希函数。它被设计为在GPU，FPGA和ASIC体系结构上无法有效计算

CryptoNight的工作原理如下：

1 该算法首先用伪随机数据初始化一个大暂存器。

2 大量的读/写操作发生在伪随机地址上包含在便签簿中。

3 整个暂存器被哈希以产生最终值。

