# HASH函数

**定义**

* 又称 哈希函数、杂凑函数、单向散列函数
* 是一个将任意长度的消息映射成固定长度输出的函数



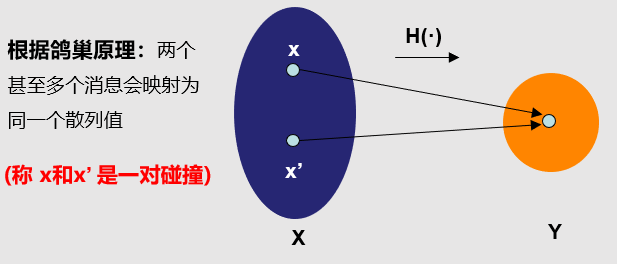
输入称为“消息”

输出称为“散列值” (Hash值、消息摘要)

n是散列值的长度

通常，Hash函数是一个具有压缩功能的函数

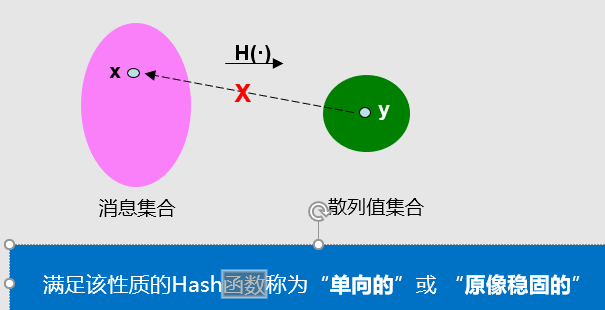
* 设X是消息的集合，Y是散列值的集合
* 我们总是假设|X|≥|Y|，并且经常假设更强的条件|X|≥2|Y|



# Hash函数的安全性

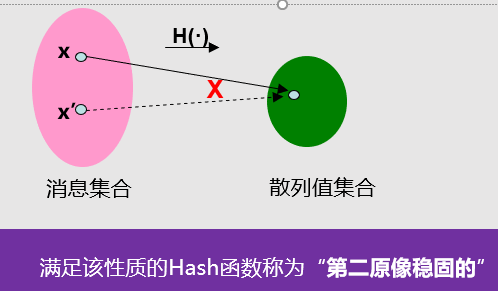
原像稳固

* 给定散列值y，要找到一个x，使得H(x)=y是计算上不可行的



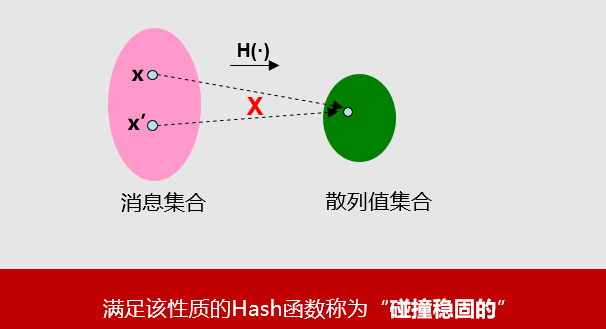
* **第二原像稳固**

给定消息x，找到另一个x’，使得H(x’)=H(x)是计算上不可行的

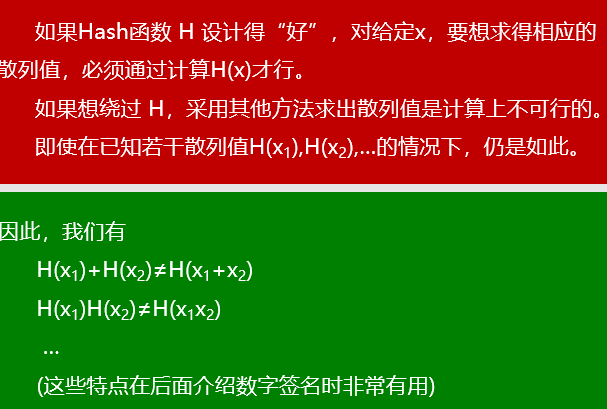


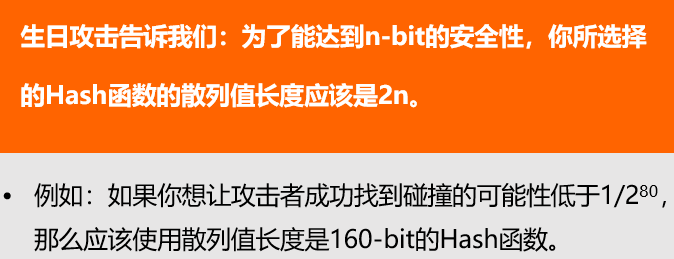
* **碰撞稳固**

找到两个不同的消息 x和x’ ，使得H(x)=H(x’)是计算上不可行的。



**对Hash函数的攻击实际就是 寻找一对碰撞 的过程**





**迭代法构造Hash函数的步骤**

1. **预处理**

用一个填充函数pad(·)在消息x右方追加若干比特，得到比特串y，使得y的长度为t的倍数。即有

y = x || pad(x) = y1 || y2 || … || yr, 其中 |yi| = t

典型的填充函数是先追加与x等长的值，再追加若干比特（如0）。

该阶段必须保证变换是单射

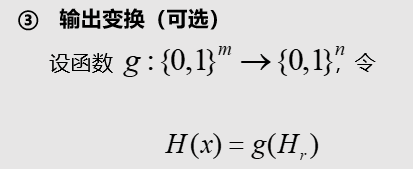
* + - 因为如果预处理变换不是单射，则存在x≠x’使得y=y’，从而H(x)=H(x’)，能够很容易找到碰撞。

1. 迭代处理

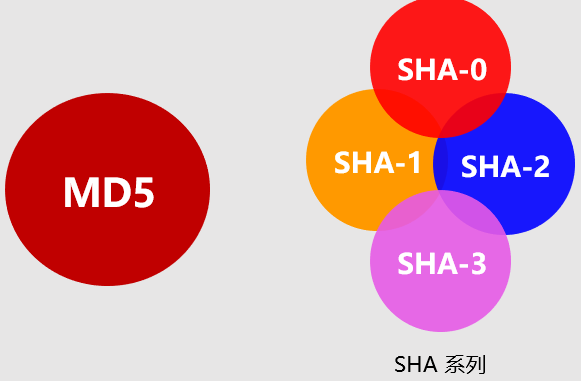
设H0=IV是一个长度为m的初始比特串，重复使用压缩函数f，依次计算

 直到计算出Hr为止。

1. 输出变换



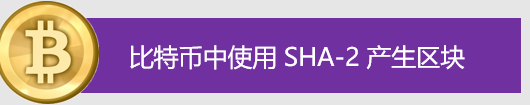
## 几个著名的Hash函数



MD5加密的说法不对

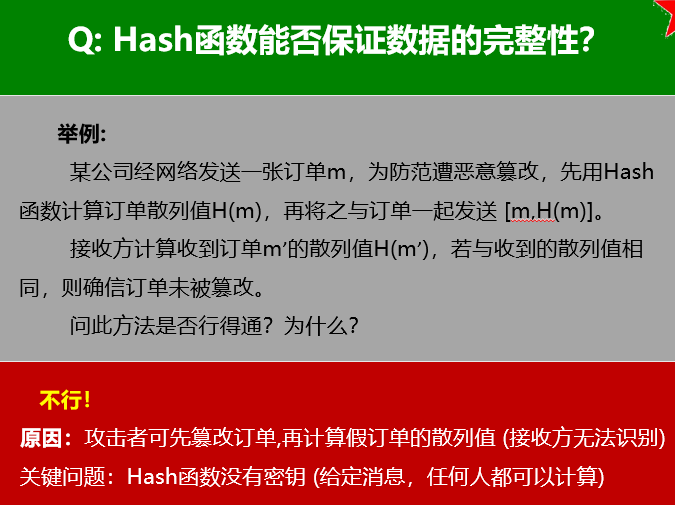
对MD5,SHA-0，SHA-1的破译（不建议再使用这些算法）

**SHA-2系列：易遭到“长度扩展攻击”，建议使用:SHA-512/224、SHA-512/256**



**Q: 消息在传输过程中是否发生改变或被恶意篡改，我们怎么能知道？**

使用“数据完整性技术”当消息发生改变时，我们可以检查出来。



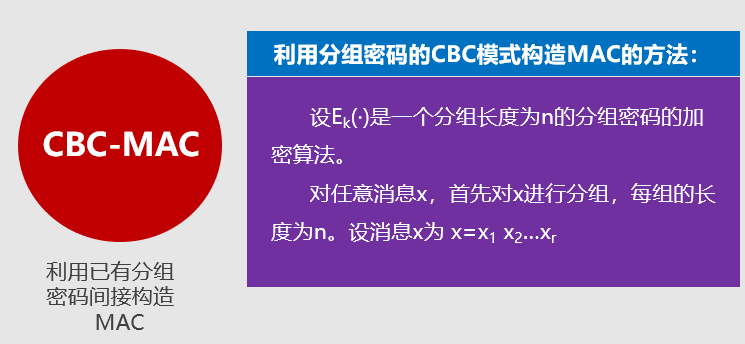


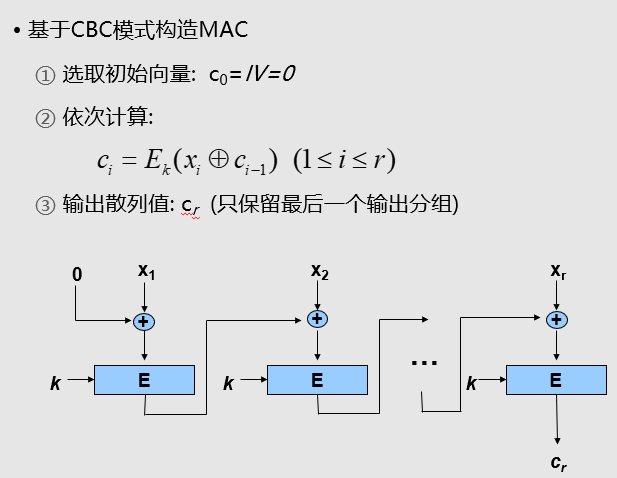
**消息认证码MAC** 安全性

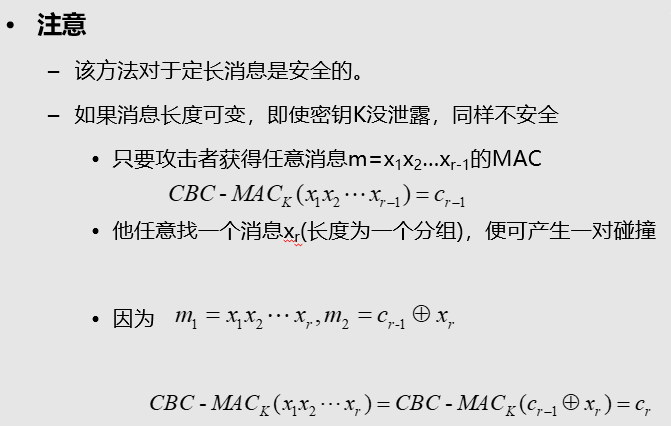
**MAC的安全性要求——抗伪造：**

1. 在不知道密钥的情况下，给定任何消息，产生相应的MAC是计算上不可行的

2. 即使已知很多消息及对应的MAC，对新消息产生MAC仍是计算上不可行的

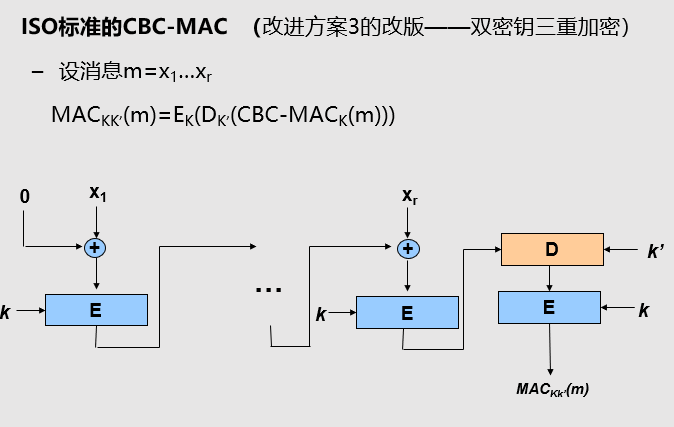






**CBC-MAC只有用于定长消息才能抗伪造**

**Q: 如何让CBC-MAC对于变长消息也能抗伪造？**



4.3认证消息

**消息认证，又称数据源认证，它的重要目标：**

1. 保证传输消息的完整性（检查消息是否被改动过）
2. 保证消息是由指定发送者发来的

主要技术： 基于MAC的认证

