哈希函数

1. 哈希函数是一个**确定性的函数**，即，给它相同的输入会得到相同的输出；它也是一个具**有压缩功能的函数**，即当你给它一个很长的输入的时候，它能进行压缩，输出固定长度的输出。
2. 哈希函数通常用 H 进行表示，输入空间用 M（消息） 表示，输出空间用 Y（输出值、消息摘要、散列值）表示，通常要求 **M 空间要远远大于 Y 空间**。
3. 假设哈希函数的输出是长度为 n 的 {0,1} 比特串，不论输入长度是多少，输出始终为 n 。不同的哈希函数输出的散列值长度不同，MD5 散列值长度为 128。

Hash函数的定义

定义（Hash函数，Hash function）（又称 哈希函数、杂凑函数、单向散列函数等）是一个高效可计算的确定性函数，它将一个长消息映射成固定长度的输出

$$H: M \rightarrow Y \quad (\ |M| >> |Y|\ )$$

- 其输入称为“消息”
- 其输出称为“散列值”（又称Hash值、消息摘要等）
- 设 $Y=\{0,1\}^n$，称n为散列值的长度

4. 与哈希函数相关的很重要的概念是碰撞，我们知道，输入空间 M 要比输出空间 Y 要大，不可避免会出现两个消息 m0 和 m1 经过哈希函数运算以后得到的结果相同，这种情况就说 m0 和 m1 是一对碰撞。对于哈希函数来说，碰撞必然是存在的，因为抽屉原理（两个甚至更多个消息会映射到同一个散列值）。

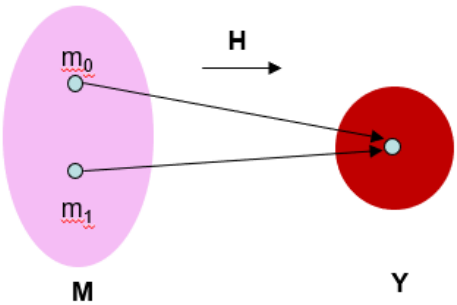
碰撞的定义

定义（碰撞，collision） Hash函数 H 的碰撞是指一对消息 $m_0, m_1 \in M$，使得

$$H(m_0) = H(m_1) \quad 但 \ m_0 \neq m_1$$

抽屉原理：两个甚至多个消息会映射为同一个散列值

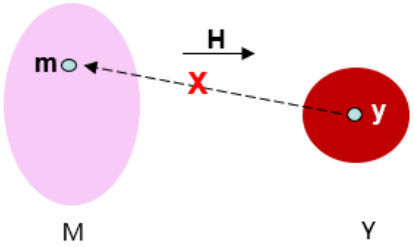
（m_0 和 m_1 是一对碰撞）



5. 哈希函数的安全性有三个方面，①原像稳固。给定了一个散列值 y，去**寻找这个散列值对应的是哪一个消息**，因为 y 可能对应很多个消息，只需找到一个就可以了，使得 H(m)=y。如果说这样的 m 在计算上不可行，那么我们就说这个哈希函数具有原像稳固这个性质，满足这个性质的哈希函数就称为是单向的或原像稳固的。所谓单向就是说，给了 m 计算 H(m)很容易，给了 H(m)去计算 m 在计算上不可行。

Hash函数的安全性

定义（原像稳固，preimage resistance）给定散列值 $y \in Y$，要找到一个消息 $m \in M$，使得H(m)=y是计算上不可行的

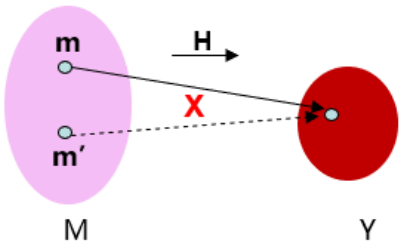


满足该性质的Hash函数称为“**单向的**”或“**原像稳固的**”

6. ②第二原像稳固。**给了一个消息 m，去寻找 m'**，使得 m 和 m'是一对碰撞，如果说找到 m'是计算上不可行的，我们就说这个哈希函数具有第二原像稳固这个性质，满足这个性质的哈希函数就成为是第二原像稳固的。

Hash函数的安全性

定义（第二原像稳固，second preimage resistance）给定一个消息$m \in M$，要找到另一个消息$m' \in M\ (m \neq m')$，使得H(m)=H(m') 是计算上不可行的

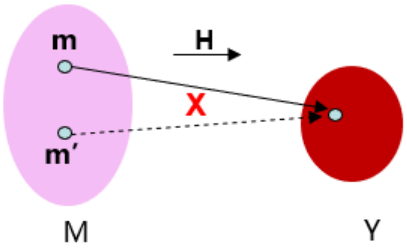


满足该性质的Hash函数称为“**第二原像稳固的**”

7. ③碰撞稳固。**寻找这样一对消息 m 和 m'**，使得 m 和 m'是一对碰撞，如果说找到这样的 m 和 m'是计算上不可行的，我们就称满足这样性质的哈希函数是碰撞稳固的。

Hash函数的安全性

定义（碰撞稳固，collision resistance）找到一对消息m, m' ∈ M (m ≠ m')，使得H(m)=H(m') 是计算上不可行的



满足该性质的Hash函数称为 **"碰撞稳固的"**

8. 要注意第二原像稳固和碰撞稳固的区别，第二原像稳固是已经给了 m，去寻找另外一个 m'；碰撞稳固是随便去找，只要能够找到任意的一对 m 和 m'使得它俩是碰撞就可以。所以说，随便找碰撞要比找第二原像容易很多。
9. 综上所述，一个安全的哈希函数要满足以下三个性质。

Hash函数的安全性

一个安全的Hash函数应该满足以下三个性质：

- **原像稳固的（单向性）**
- **第二原像稳固的（弱抗碰撞性）**
- **碰撞稳固的（强抗碰撞性）**

10. 注意！对于攻击者来说，找碰撞比找第二原像要容易，所以说，哈希函数在设计的时候要想 抵抗找第二原像 要比 抵抗找碰撞 要容易实现，因为攻击者攻击的越容易，在设计的时候要想不让他成功反而越难，所以说把第二原像稳固称作是弱抗碰撞性，而把碰撞稳固称作强抗碰撞性。（**找碰撞更容易就更难抵抗**）。
11. 在哈希函数里，还有一种哈希函数叫做抗碰撞哈希函数，假设 H 是一个哈希函数，如果说对于所有的高效的攻击者它们能够找到碰撞的概率是可忽略的，那么我们就称 H 是一个抗碰撞的哈希函数。即，如果一个哈希函数是抗碰撞的哈希函数，那么它具有两个性质，①碰撞稳固，②第二原像稳固。**在这个概念里，不考虑原像稳固**。
12. 哈希函数和抗碰撞哈希函数的区别：哈希函数考虑原像稳固，抗碰撞哈希函数不考虑原像稳固。

13. 虽然在抗碰撞哈希函数的定义里没有说明是否要满足原像稳固，但在实际应用中，人们去设计抗碰撞哈希函数的时候，同时也会把它设计成一个原像稳固的，即，**真正设计抗碰撞哈希函数时，这三个性质都要考虑**。

Hash函数的安全性

定义（抗碰撞Hash函数，collision-resistant hash function）

设 H 是 Hash函数，如果所有高效的攻击者 A 能够找到碰撞的概率是可忽略的，则称 H 为 抗碰撞Hash函数

（具有的性质：碰撞稳固、第二原像稳固）

虽然定义中并未说明，但实际应用中的抗碰撞Hash函数也同时设计成原像稳固的

例如：SHA-256，SHA3-256 等

14. 哈希函数设计好之后，对于给定消息 m，要想求 m 的散列值，就**必须通过调用这个哈希函数 H 本身才能获得**，如果攻击者**想绕过 H 求出 m 的散列值，必须应该是计算上不可行的**。即使攻击者获得一堆 m1 的散列值、m2 的散列值，想要通过 m1、m2 以及它们的散列值拼凑出新的消息 m 的散列值（m 可能与 m1、m2 相关）在计算上是不可行的。
15. H(m1) + H(m2) ≠ H(m1+m2) ≠ H(m1*m2) ≠ H(m1/m2)
    H(m1) * H(m2) ≠ H(m1*m2) 等，即它们不会存在简单的数学上的关系

生日问题：假设每个人生日是等概率的，每年有 365 天，问 k 最小应是多少，才能使 k 人中至少有 2 人生日相同的概率大于 1/2？可以先求 k 个人生日都不同的概率，第一个人为 1/365

k人生日都不同的概率是多少?

$$\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{k-1}{365}\right)$$

k人中至少有2人生日相同的概率为:

$$p = 1-\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{k-1}{365}\right)$$
$$\approx 1-e^{-\frac{k(k-1)}{2\times 365}}$$
$$p>\frac{1}{2}\text{时，}\ k\approx 1.1774\sqrt{365}\approx 23$$

只需 23 人就至少有两个生日相同的概率大于 1/2，比直觉上小得多，因而称为**生日悖论**。

如何利用生日悖论的原理去攻击抗碰撞的哈希函数?

对抗碰撞Hash函数的生日攻击

- **利用生日悖论原理攻击抗碰撞Hash函数**
  - **目的**：设散列值是n比特，构造两个不同的消息m和M，使得m和M具有相同的散列值。

m：明天8点电影院见，约吗?

M：请转账100万到我的银行账户

对哈希函数找碰撞，方法很简单，假设现在有一个哈希函数，它的散列值是 n 比特，构造两个不同的消息 m 和 M，使得 m 和 M 具有相同的散列值，如果它们具有相同的散列值，我们就找到了一对碰撞。我们需要对 m 和 M 进行一些修改，使得修改后的消息是一对碰撞。

对抗碰撞Hash函数的生日攻击

- **攻击步骤（1）：消息变形**
  - 分别改变 m 和 M 的内容，变成 r 和 R 个新消息
  - 要求保证语义不变，例如 增加空格、使用缩写、使用意义相同的单词、去掉不必要的单词等。

m：明天8点电影院见，约吗?

m_1：明天8点电影院见，好吗?

m_2：明天8点电影院见，OK?

m_3：明天八点电影院见吧?

…

M：请转账100万到我的银行账户

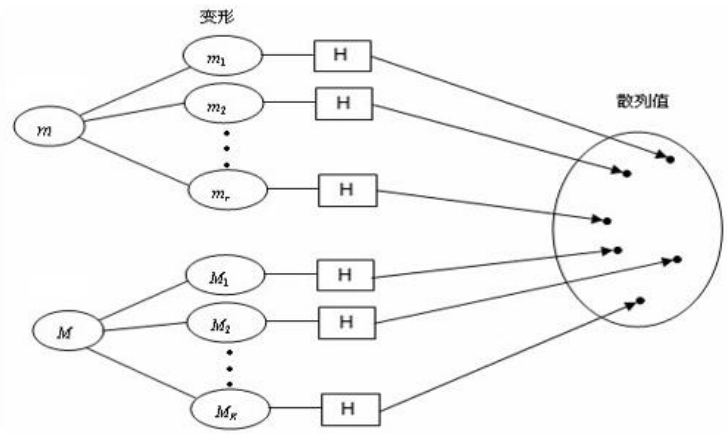
M_1：请转账100万到我的账户

M_2：请转账10,000,000到我的账户

M_3：请转账一百万到我的银行账户

….

- **攻击步骤（2）：计算散列值**
  - 分别计算变形后消息的散列值



对抗碰撞Hash函数的生日攻击

- **攻击步骤（3）：找碰撞**
  - 查找散列值相同的消息m_i和M_j

对抗碰撞Hash函数的生日攻击

发生碰撞的概率是多少?

- n比特长的输出共有2^n个散列值
- 对于给定m的一个变形m_i和M的一个变形M_j，m_i与M_j不碰撞的概率是$1-1/2^n$

M有R个变形，所以M的全部变形都不与m_i碰撞的概率：$\left(1-1/2^n\right)^R$

m有r个变形，因此m的变形与M的变形都不碰撞的概率：$\left(1-1/2^n\right)^{rR}$

m的变形与M的变形发生碰撞的概率：$P(n)=1-\left(1-\frac{1}{2^n}\right)^{rR}\approx 1-e^{-\frac{rR}{2^n}}$

对抗碰撞Hash函数的生日攻击

安全性分析

- $r=R=2^{n/2}$时，$P(n)\approx 1-e^{-1}=0.63$
- 当散列值长度n=64时，只需分别构造$r=2^{32}$个m的变形和$R=2^{32}$个M的变形，便可以0.63的概率找到一对碰撞，完全可行。

对于散列值长度为64 bit 的Hash函数，
生日攻击的时间复杂度约为$O(2^{32})$，所以是不安全的。

生日攻击告诉我们：要想达到n-bit安全性，选择的抗碰撞Hash函数的散列值长度应该是2n。

例如：如果想让攻击者成功找到碰撞的可能性低于$1/2^{80}$，应该使用散列值长度至少是160-bit的抗碰撞Hash函数。