现代密码学

Modern Cryptography

张方国

中山大学计算机学院

Office: Room 305, IM School Building

E-mail: isszhfg@mail.sysu.edu.cn

HomePage: https://cse.sysu.edu.cn/content/2460

第十一讲 Hash函数（一）

- Hash函数与数据完整性
- Hash函数的安全性
- 迭代Hash函数
- 消息认证码
- 无条件安全消息认证码

Hash函数

密码学上的Hash函数可为数据完整性提供保障。可当作“指纹”来验证数据的完整性。

假设h是一个Hash函数，x是数据。设x是任意长度的二元串，相应的指纹为$y = h(x)$。通常指纹也被称为消息摘要，其通常为一个长度为160比特的二元串。

使用y，我们能判定x是否被改变了。

Hash函数在数字签名中有着重要的作用。

Hash函数

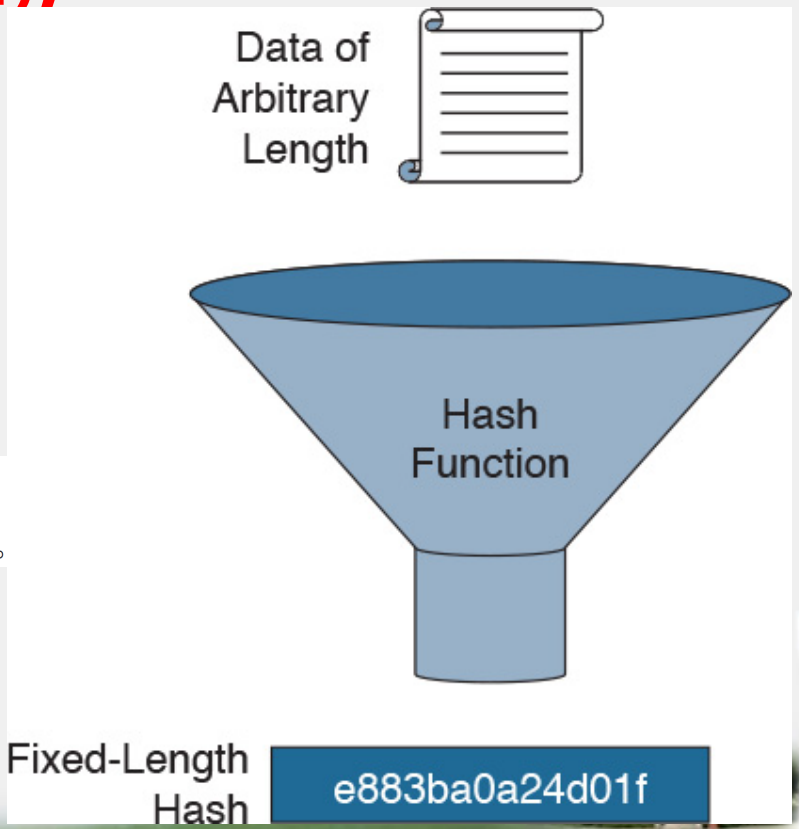
- Hash函数是将任意长度的消息映射成一个较短的定长输出消息的函数.

$$h = H(M),$$

M是变长的消息,
h是定长的Hash值.

- 带密钥的Hash:Hk(M)
- 不带密钥的Hash

不带密钥的Hash函数，消息摘要必须被安全地存放，不能被篡改。
带密钥的Hash函数，可以在不安全信道同时传送数据和认证标签。



Hash函数

定义4.1 一个Hash族是满足下列条件的四元组$(\mathbb{X},\mathbb{Y},\mathbb{K},\mathbb{H})$。

1.$\mathbb{X}$是所有可能的消息的集合。

2.$\mathbb{Y}$是所有可能的消息摘要或认证标签构成的有限集。

3.$\mathbb{K}$是密钥空间，是所有可能的密钥构成的有限集。

4.对每个$K\in\mathbb{K}$，存在一个Hash函数$h_K\in\mathbb{H}$，$h_K:\mathbb{X}\to\mathbb{Y}$。

说明：$\mathbb{X}$可以是有限或无限集，$\mathbb{Y}$总是有限集。如果$\mathbb{X}$是有限集，则Hash函数称为压缩函数。总假定$|\mathbb{X}|\geq|\mathbb{Y}|$或$|\mathbb{X}|\geq 2|\mathbb{Y}|$。

如果$h_K(x)=y$，则对$(x,y)\in\mathbb{X}\times\mathbb{Y}$称为在密钥$K$下是有效的。

令$\mathbb{F}^{\mathbb{X},\mathbb{Y}}$为所有从$\mathbb{X}$到$\mathbb{Y}$的函数集合。假定$|\mathbb{X}|=N$和$|\mathbb{Y}|=M$。则显然$|\mathbb{F}^{\mathbb{X},\mathbb{Y}}|=M^N$。任何Hash函数族$\mathbb{F}\subseteq\mathbb{F}^{\mathbb{X},\mathbb{Y}}$被称为一个$(N,M)$-Hash族。

Hash函数安全性

假定h是一个不带密钥的Hash函数，则产生有效对(x,y)满足$y=h(x)$的方法如下：首先选择x，再把函数h作用于x，计算出$y=h(x)$。

我们说一个Hash函数是安全的，如果以下的三个问题都是难解的。

1：原像问题；

2：第二原像；

3：碰撞。

Hash函数安全性：原像问题

问题4.1(原像preimage)

实例：Hash函数$h : \mathbb{X} \to \mathbb{Y}$和$y \in \mathbb{Y}$。

找出：$x \in \mathbb{X}$使得$h(x) = y$。

不能解决原像问题的Hash函数通常称为单向的或者原像稳固的。

Hash函数安全性：第二原像

问题4.2(第二原像)

实例：Hash函数$h:\mathbb{X}\to\mathbb{Y}$和$x\in\mathbb{X}$。

找出：$x'\in\mathbb{X}$使得$x\neq x'$，并且$h(x)=h(x')$。

不能解决第二原像问题的Hash函数通常称为第二原像稳固的。

Hash函数安全性：碰撞

问题4.3(碰撞)

实例：Hash函数$h:\mathbb{X}\to\mathbb{Y}$。

找出：$x,x'\in\mathbb{X}$使得$x\neq x'$，并且$h(x)=h(x')$。

不能解决碰撞问题的Hash函数通常称为碰撞稳固的。

Hash函数安全性

一个理想的Hash函数的概念。

如果Hash函数h设计得好，对给定的x，求出函数h在点x的值应该是得到$h(x)$的唯一有效的方法，即使其他的值$h(x_1), h(x_2), \ldots$已经计算出来，这仍然是正确的。

(实例)假定函数$h : \mathbb{Z}_n \times \mathbb{Z}_n \to \mathbb{Z}_n$是一个线性函数，令

$$h(x,y) = ax + by \bmod n$$

$a, b \in \mathbb{Z}_n$且$n \geq 2$是正整数。假定已得到$h(x_1, y_1) = z_1$和$h(x_2, y_2) = z_2$，令$r, s \in \mathbb{Z}_n$，则有

$$h(rx_1 + sx_2 \bmod n, ry_1 + sy_2 \bmod n) = rh(x_1, y_1) + sh(x_2, y_2) \bmod n$$

因此，只要知道函数h在(x_1, y_1)和(x_2, y_2)两点的值，就可以知道其他各点的值。

Hash函数安全性

- 随机谕示模型（随机预言模型）

Bellare和Rogaway引入的随机喻示模型给出了一个理想的Hash函数的数学模型。在这个模型中，随机从$\mathbb{F}^{\mathbb{X},\mathbb{Y}}$中选出一个Hash函数，我们仅允许喻示器访问函数$h$。这意味着不会给出一个公式或算法来计算函数$h$的值。

计算$h(x)$的唯一方法是询问喻示器。

真正的随机喻示器不存在，但是我们希望一个精心设计的Hash函数具有一个随机喻示器的性质。

Hash函数安全性

假设存在随机喻示模型，则：

定理4.1 假定$h \in \mathbb{F}^{\mathbb{X},\mathbb{Y}}$是随机选择的，令$\mathbb{X}_0 \subseteq \mathbb{X}$。假定当且仅当$x \in \mathbb{X}_0$时，$h(x)$被确定（通过查询$h$的喻示器）。则对所有的$x \in \mathbb{X} \setminus \mathbb{X}_0$和$y \in \mathbb{Y}$，都有$Pr[h(x) = y] = 1/M$。

我们考虑随机喻示模型下，原像问题、第二原像以及碰撞问题的复杂性，即在随机喻示模型下，解决三个问题的困难程度。

Hash函数安全性

我们介绍的算法都是随机算法：它们在执行过程中做出随机选择。

Las Vegas算法是一个不一定给出答案的随机算法（即以失败而终止），但是一旦该算法返回一个答案，那么这个答案就是正确的。

成功概率

对每个问题实例，一个随机算法能返回一个正确答案的概率至少是$\varepsilon \in [0,1)$，那么该算法具有最差情况成功率ε。

对每个问题，一个随机算法平均能返回一个正确答案的概率至少是ε，那么该算法具有平均成功率ε。

(ε, Q)算法表示一个具有平均情况成功率ε的Las Vegas算法，其中该算法向喻示器查询的次数最多为Q。如果x和/或y被指定为问题实例的一部分，则成功率ε是对所有的$h \in \mathbb{F}^{\mathbb{X},\mathbb{Y}}$和$x \in \mathbb{X}$和$y \in \mathbb{Y}$的可能出现的随机选择的平均。

Hash函数安全性

分析那些在随机喻示模型中计算Q个$x \in \mathbb{X}$的$h(x)$之值的一般算法。实际上是对所有函数$h \in \mathbb{F}^{\mathbb{X},\mathbb{Y}}$取平均值，这就说明了这个算法的复杂性独立于$Q$个$x$值的选择。

考虑下算法4.1，该算法企图通过计算在Q个点的h值来解决原像问题。

算法4.1 Find-preimage(h, y, Q)

选择任意的$X_0 \subseteq X$，$|X_0| = Q$

For each $x \in \mathbb{X}_0$

return (x) if $h(x) = y$

return（failure)

定理4.2 对任意的$\mathbb{X}_0 \subseteq \mathbb{X}$，且$|\mathbb{X}_0| = Q$，算法4.1平均情况成功概率是

$$\varepsilon = 1 - (1 - \frac{1}{M})^Q$$

Hash函数安全性

现在介绍和分析一个类似企图解决第二原像问题的算法.

算法4.2 Find-Second-Preimage(h,x,Q)

$y \leftarrow h(x)$

选择$\mathbb{X}_0 \subseteq \mathbb{X} \setminus \{x\}$，$|\mathbb{X}_0| = Q-1$

For each $x_0 \in \mathbb{X}_0$

return (x_0) if $h(x_0) = y$

return（failure)

定理4.3 对任意的$\mathbb{X}_0 \subseteq \mathbb{X} \setminus \{x\}$，且$|\mathbb{X}_0| = Q-1$，算法4.2的平均情况成功概率是

$$\varepsilon = 1 - (1 - \frac{1}{M})^{Q-1}$$

Hash函数安全性

针对碰撞问题的基本算法.

算法4.3 Find-Collision(h, Q)

选择$\mathbb{X}_0 \subseteq \mathbb{X}$，$|\mathbb{X}_0| = Q$

For each $x \in \mathbb{X}_0$

do $y_x \leftarrow h(x)$

if 对某一个$x' \neq x$，有$y_x = y_{x'}$，then return (x, x')

return（failure)

定理4.4 对任意的$\mathbb{X}_0 \subseteq \mathbb{X}$，且$|\mathbb{X}_0| = Q$，算法4.3的成功概率是

$$\varepsilon = 1 - \frac{M-1}{M}\frac{M-2}{M}\cdots\frac{M-Q+1}{M}$$

生日悖论

1、在一个房间里，至少需要多少人，才可以找到一个与Alice的生日为同一天的概率大于1/2？

2、在一个房间里，至少需要多少人，才可以找到两个人的生日为同一天的概率大于1/2？

Answer：1. $P=1-(364/365)^{t-1}$ $t=183$

2, 23

Collision Search-1

For collision search, select distinct inputs x_i for i=1, 2, ... , n, where n is the number of hash bits and check for a collision in the $h(x_i)$ values

The prob. that no collision is found after selecting k inputs is

$$p_{\text{no collision}} = \left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot\left(1-\frac{3}{n}\right)\cdot\cdot\left(1-\frac{(k-1)}{n}\right)$$

(In the case of the birthday paradox k is the number of people randomly selected and the collision condition is the birthday of the people and n=365.)

Collision Search-2

For large *n*

$$p_{\text{no collision}} = \left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{(k-1)}{n}\right) \approx e^{-k^2/(2n)}$$

$$1-x \approx e^{-x} \text{ when x is small}$$

$$\left(1-\frac{1}{n}\right) \approx e^{-1/n}$$

$$p_{\text{no collision}} = e^{-1/n}\cdot e^{-2/n}\cdots e^{-(k-1)/n}$$

$$= e^{-\left((1+2+3+\ldots+(k-1))/n\right)}$$

$$= e^{-k\cdot(k-1)/2n}$$

Collision Search-3

When *k* is large, the percentage difference between *k* and *k-1* is small, and we may approximate *k-1* ≈ *k*.

$$p_{\text{no collision}} = e^{-k\cdot(k-1)/2n} = e^{-k^2/2n}$$

$$p_{\text{at least one collision}} = 1 - e^{-k^2/2n}$$

Collision Search-4

$$p = 1 - e^{-k^2/2n}$$

$$e^{-k^2/2n} = 1 - p$$

$$e^{k^2/2n} = \frac{1}{1-p}$$

$$\frac{k^2}{2n} = \ln\left(\frac{1}{1-p}\right)$$

$$k = \sqrt{2n * \ln\left(\frac{1}{1-p}\right)}$$

- For the birthday case, the value of *k* that makes the probability closest to 1/2 is 23

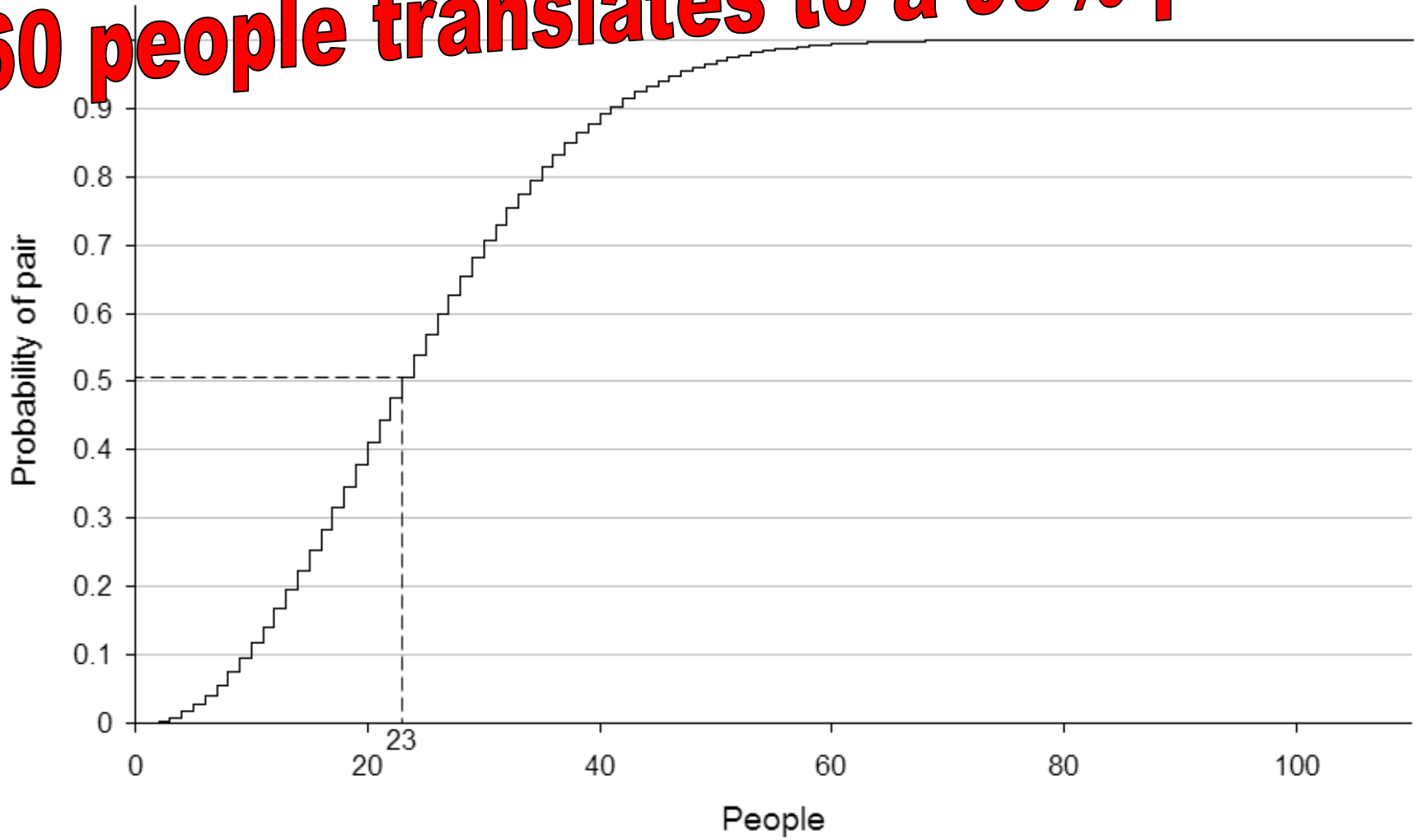
$$k = \sqrt{2n * \ln 2}$$

$$= 1.1774\sqrt{n}$$

$$= 1.1774 * \sqrt{365} = 22.49$$

Birthday Paradox Problem

60 people translates to a 99% probability



Attack Prevention

The important property is the length in bits of the message digest produced by the hash function.

If the number of *m* bit hash , the cardinality *n* of the hash function is

$$n = 2^m$$

The 0.5 probability of collision for m bit hash, expected number of operation *k* before finding a collision is very close to

$$k \approx \sqrt{n} = 2^{m/2}$$

m should be large enough so that it's not feasible to compute hash values!!!

Hash函数安全性

从定理4.2，4.3，4.4我们发现，解决碰撞问题比解决原像问题和第二原像问题要容易。

一个相关问题是是否在可应用于任意Hash函数的这三个问题中存在归约。利用算法4.4可以把碰撞问题归约为第二原像问题。

算法4.4 Collision-To-Second-Preimage(h)

External Oracle-2nd-Preimage

均匀地随机选择$x \in \mathbb{X}$

if oracle-2nd-Preimage$(h, x) = x'$

then return (x, x')

return (failure)

Hash函数安全性

碰撞问题是否可归约为原像问题？我们讲证明，在一些特殊情况下，任何能解决原像问题且概率为1的算法也能解决碰撞问题。

算法4.5 Collision-To-Preimage(h)

External Oracle-Preimage

均匀地随机选择$x \in \mathbb{X}$

$y \leftarrow h(x)$

if oracle-Preimage$(h, y) = x'$ 且$(x \neq x')$

then return (x, x')

return (failure)

定理4.5 假定$h : \mathbb{X} \to \mathbb{Y}$是一个Hash函数，$|\mathbb{X}|$和$|\mathbb{Y}|$是有限的，且$|\mathbb{X}| \geq 2|\mathbb{Y}|$。假定Oracle-preimage对固定的Hash函数h是原像问题的一个$(1, Q)$算法。则Collision-to-Preimage对固定的Hash函数h是碰撞问题的一个$(1/2, Q+1)$算法。