七、MAC与CRHF

哈尔滨工业大学

张宇

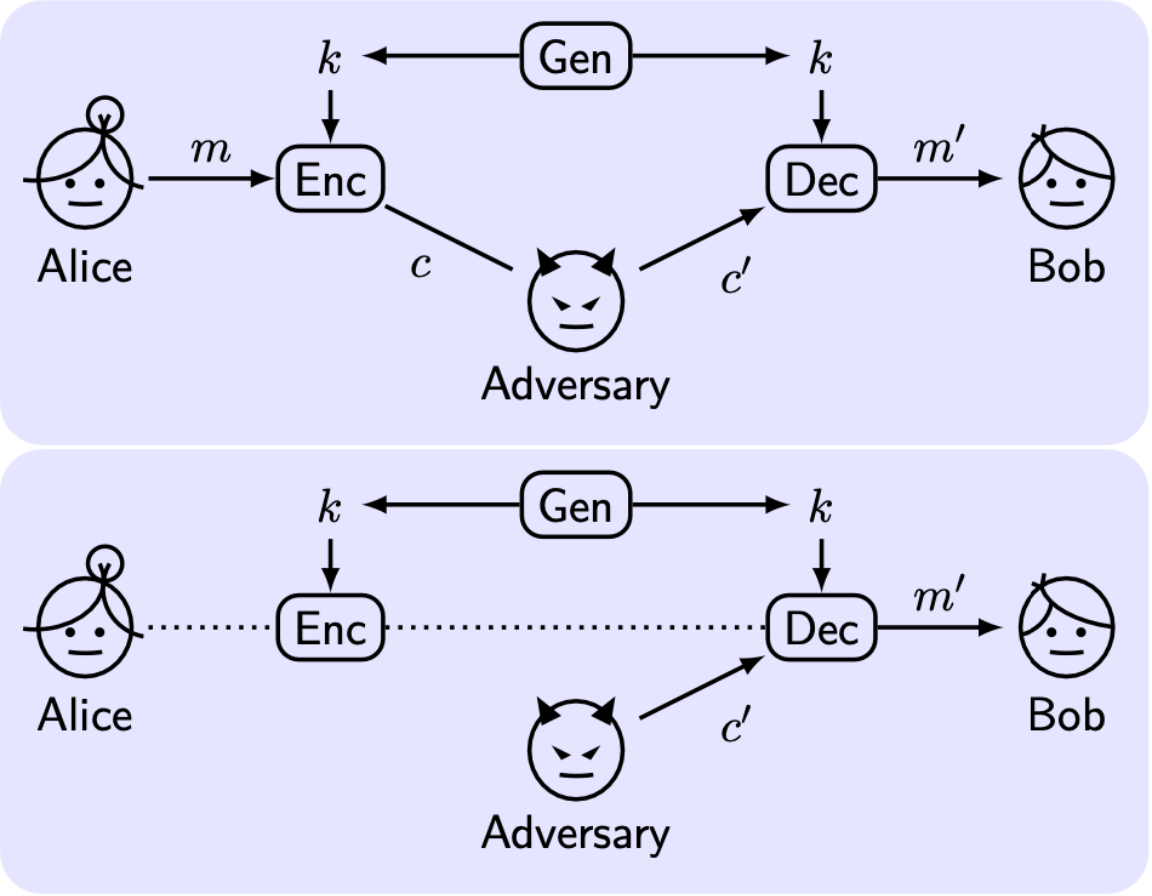
2024春

概览

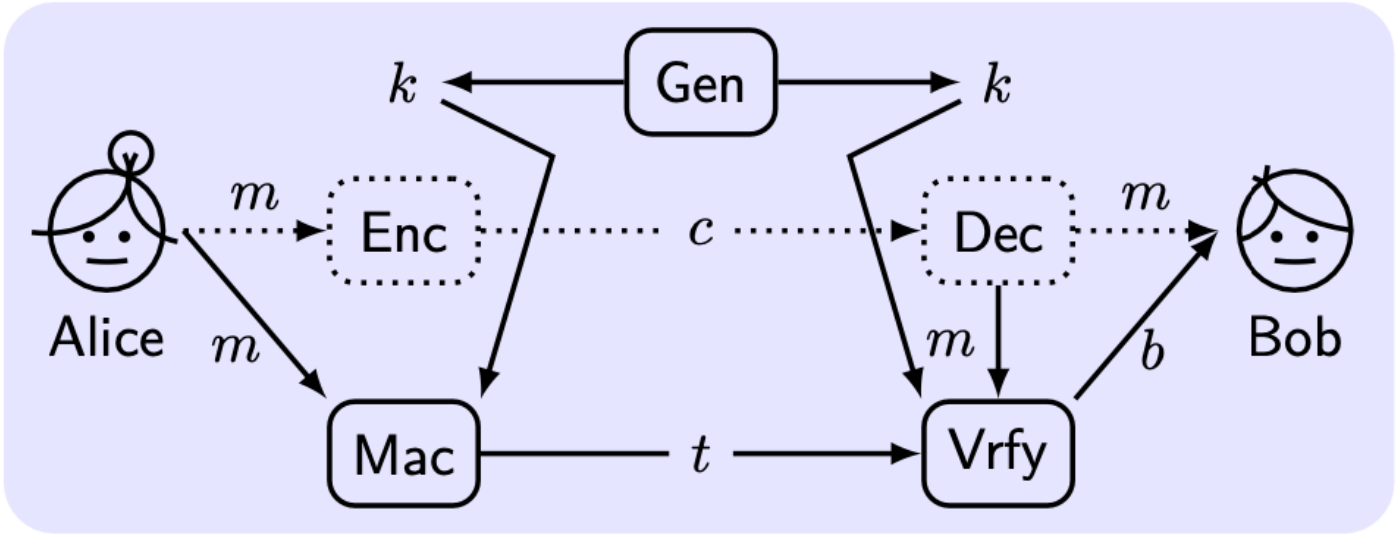
1. 消息认证码（MAC）安全
2. CBC-MAC
3. 抗碰撞哈希函数（CRHF）
4. 基于哈希的MAC（HMAC）
5. 信息论MAC（简介）

完整性与真实性

- 敌手篡改传输中的密文（或明文）是对完整性的攻击；敌手伪装成Alice发送密文（或明文）是对真实性（认证）的攻击；归结为对真实性的攻击：
- 注意，真实性是指消息是来自预期的发送者，不是指内容的真假！



MAC（消息认证码）词法



- key k, tag t, a bit b means valid if $b = 1$; invalid if $b = 0$.
- **Key-generation** algorithm $k \leftarrow \mathsf{Gen}(1^n)$, $|k| \geq n$. 密钥生成
- **Tag-generation** algorithm $t \leftarrow \mathsf{Mac}_k(m)$. 标签生成
- **Verification** algorithm $b := \mathsf{Vrfy}_k(m, t)$. 验证
- **Message authentication code**: $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ 消息验证码
- **Basic correctness requirement**: $\mathsf{Vrfy}_k(m, \mathsf{Mac}_k(m)) = 1$.

安全MAC定义

- 直觉上，没有敌手能够伪造一个**从未被发送过的新消息**的有效标签。这里“新消息”是为了排除“重放攻击”。
- 重放攻击（Replay attack）：敌手记录并发送之前的消息和标签，从而发送了一个伪造的消息并带有有效的标签；为了避免重放攻击，可以通过两种非密码学的方法。
  - 序列号：接收方需要记录之前的序列号，从而发现序列号较小（或曾经接收过的）的旧消息；
  - 时间戳：双方维护时钟同步，从而发现晚与当前时钟的旧消息；
  - 这两种方法都不依赖于密码学，因此，防御重放攻击不需要在密码学的范畴内考虑。
- 存在性不可伪造（Existential unforgeability）：不能伪造任何消息的标签，一个都不能伪造。
- 适应性选择消息攻击（Adaptive chosen-message attack (CMA)）：敌手在攻击过程中始终具有获得任意消息的有效标签的能力，即访问标签生成预言机；

MAC应用

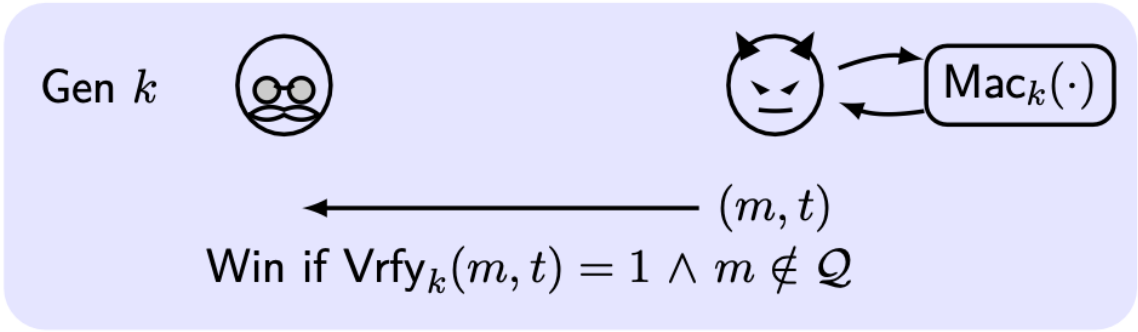
- Web cookie：Web服务器在发给浏览器的cookie中包含自己生成的MAC标签，来阻止攻击者伪造其他用户的cookie；
- TCP SYN cookie：在TCP三次握手中，服务器在其发给客户端的初始序列号中包含一个服务器生成的MAC标签，来避免保留握手状态，从而防御SYN Flooding DDoS攻击；
- 临时一次口令：用户发送给服务器的临时登录口令为一个MAC标签p=Mac_k(T)，其中k为原始口令，T为当前时间（按半分钟取整）；敌手窃听了之前的临时口令也无法伪造未来的临时口令；

MAC安全定义

- 敌手成功：如果输出的消息和标签通过了验证，并且输出的消息是从未向预言机查询过的新消息。

The message authentication experiment $\text{Macforge}_{\mathcal{A},\Pi}(n)$:

1. $k \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access to $\text{Mac}_k(\cdot)$, and outputs (m, t). $\mathcal{Q}$ is the set of queries to its oracle.
3. $\text{Macforge}_{\mathcal{A},\Pi}(n) = 1 \iff \text{Vrfy}_k(m, t) = 1 \wedge m \notin \mathcal{Q}$.



Definition 1

A MAC Π is **existentially unforgeable under an adaptive CMA** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that: $\Pr[\text{Macforge}_{\mathcal{A},\Pi}(n) = 1] \leq \text{negl}(n)$.

真实案例

The 802.11b Insecure MAC[3]

Consider a variant of WiFi encryption in 802.11b WEP (Wired Equivalent Privacy). Let F be a PRF with a 32-bit length output. Let CRC32 be an error-detecting code outputting a 32-bit string. Define the following MAC scheme:

$$S(k,m) := (r \leftarrow \{0,1\}^n, t \leftarrow F(k,r) \oplus \text{CRC32}(m))$$

$$V(k,m,(r,t)) := 1 \quad \text{if} \quad t = F(k,r) \oplus \text{CRC32}(m)$$

- Different messages may have the same CRC32 output.
- Attacker can learn $F(k,r)$ from a valid tag, and then output $(m', (r, F(k,r) \oplus \text{CRC32}(m')))$.

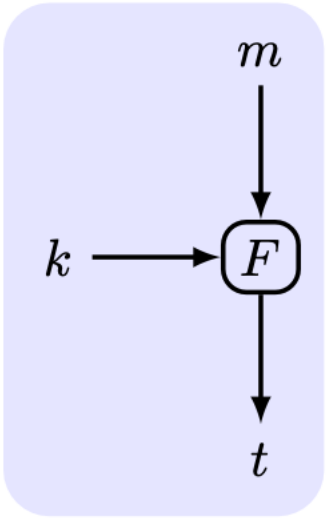
- 攻击MAC的两种常用手段:
  - 找到两个消息得到相同的中间结果，从而以一个消息的标签作为另一个新消息的标签;
  - 利用对一个/多个消息的标签来获得构造标签所需的信息，从而构造一个新消息的标签。

课堂练习

Suppose $\langle S, V\rangle$ are CMA-secure, are $\langle S', V'\rangle$ secure?

- $S'_k(m) = (S_k(m), m)$, $V'_k(m, (t_1, t_2)) = V_k(m, t_1) \wedge t_2 = m$
- $S'_{k_1,k_2}(m) = (S_{k1}(m), S_{k_2}(m))$
  $V'_{k_1,k_2}(m, (t_1, t_2)) = V_{k1}(m, t_1) \wedge V_{k_2}(m, t_2)$
- $S'_k(m) = (S_k(m), S_k(m))$
  $$V'_k(m, (t_1, t_2)) = \begin{cases} V_k(m, t_1) & \text{if } t_1 = t_2 \\ 0 & \text{otherwise} \end{cases}$$
- $S'_k(m) = (S_k(m), S_k(0^n))$
  $V'_k(m, (t_1, t_2)) = V_k(m, t_1) \wedge V_k(0^n, t_2)$
- $S'_k(m) = S_k(m)$, $V'_k(m, t) = \begin{cases} V_k(m, t) & \text{if } m \neq 0^n \\ 1 & \text{otherwise} \end{cases}$
- $S'_k(m) = S_k(m)$ without the LSB
  $V'_k(m, t) = V_k(m, t\|0) \ \vee \ V_k(m, t\|1)$

构造安全MAC



Construction 2

- F *is PRF.* $|m| = n$.
- $\mathsf{Gen}(1^n)$: $k \leftarrow \{0,1\}^n$ *u.a.r.*
- $\mathsf{Mac}_k(m)$: $t := F_k(m)$.
- $\mathsf{Vrfy}_k(m,t)$: $1 \iff t \stackrel{?}{=} F_k(m)$.

Theorem 3

If F is a PRF, Construction is a secure fixed-length MAC.

Lemma 4

***Truncating MACs based on PRFs**: If F is a PRF, so is $F_k^t(m) = F_k(m)[1,\ldots,t]$.*

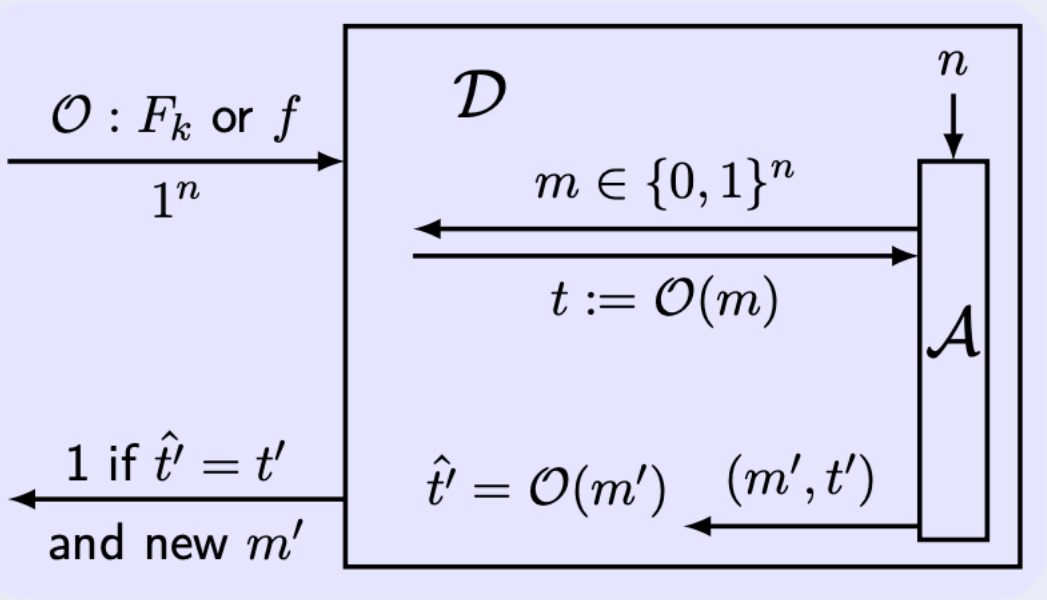
- 注：这个引理说明部分输出仍保留伪随机性。引理成立的原因在于，如果根据更短的输出可以区分出伪随机函数，那么根据原长度输出也可以区分出伪随机函数了。

MAC安全性证明

- 证明思路是从PRF的区分器算法D规约到伪造标签的敌手算法A。D作为A的挑战者，用D要区分的预言机作为A的标签生成预言机；当A伪造标签成功时，D输出1。

Proof.

D distinguishes F_k; $\mathcal{A}$ attacks Π.



□

MAC安全性证明

Proof.

(1) If true random f is used, $t = f(m)$ is uniformly distributed.

$$\Pr[D^{f(\cdot)}(1^n) = 1] = \Pr[\text{Macforge}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 2^{-n}.$$

(2) If F_k is used, conduct the experiment $\text{Macforge}_{\mathcal{A},\Pi}(n)$.

$$\Pr[D^{F_k(\cdot)}(1^n) = 1] = \Pr[\text{Macforge}_{\mathcal{A},\Pi}(n) = 1] = \varepsilon(n).$$

According to the definition of PRF,

$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \geq \varepsilon(n) - 2^{-n}.$$

□

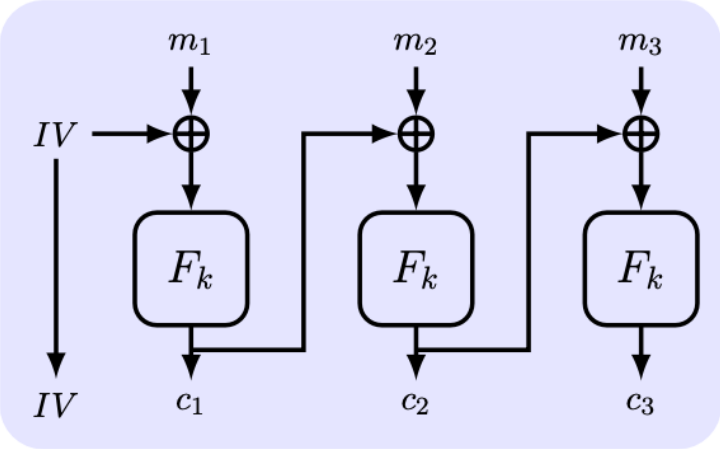
课堂练习

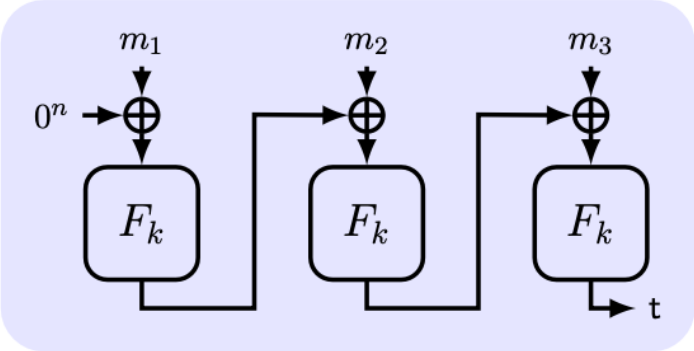
- 对于变长消息，下面的建议是安全的吗？

For variable-length messages, would the following suggestions be secure?

- **Suggestion 1**: XOR all the blocks together and authenticate the result. $t := \mathsf{Mac}'_k(\oplus_i m_i)$.
- **Suggestion 2**: Authenticate each block separately. $t_i := \mathsf{Mac}'_k(m_i)$.
- **Suggestion 3**: Authenticate each block along with a sequence number. $t_i := \mathsf{Mac}'_k(i \| m_i)$.

构造固定长度的CBC-MAC





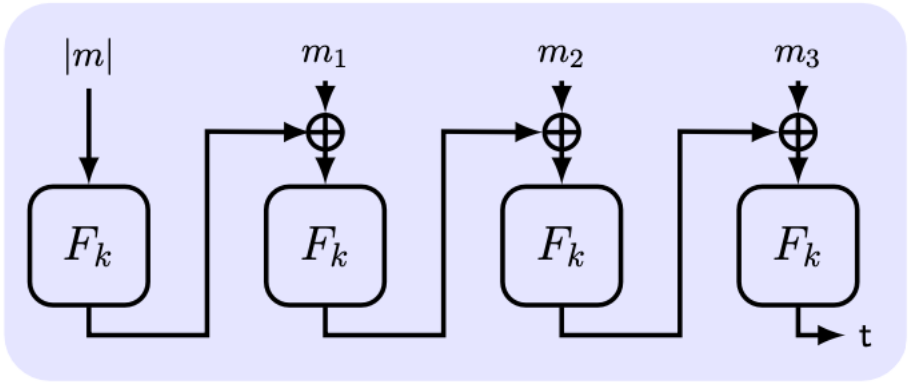
Modify CBC encryption into CBC-MAC:

- Change random IV to encrypted fixed 0^n,*otherwise*:
  Q: query m_1 and get (IV, t_1); output $m_1' = IV' \oplus IV \oplus m_1$ and $t' =$ ____.
- Tag only includes the output of the final block,*otherwise*:
  Q: query m_i and get t_i; output $m_i' = t_{i-1}' \oplus t_{i-1} \oplus m_i$ and $t_i' =$ ____.

- 改动1：将初始向量IV改为0；如果不这样改动，则敌手查询 m_1 并获得 (IV, t_1)；然后，输出 $m_1' = IV' \oplus IV \oplus m_1$ 并且 $t' = (IV', t_1)$，一个有效的标签。
- 改动2：标签只包括最后一个块的输出；如果不这样改动，则敌手查询 m_i 并得到 t_i；然后，输出 $m_i' = t_{i-1}' \oplus t_{i-1} \oplus m_i$ 以及 $t_i' = t_i$，一个有效的标签。

安全变长MAC

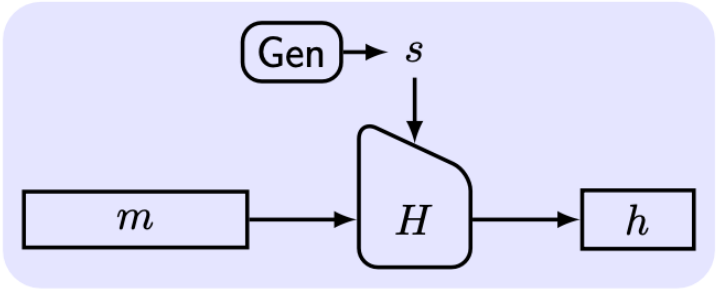
- 有三种方法可以将CBC-MAC改造为用于变长消息的MAC，都可以防御上面在结尾添加新块的攻击。
  - 输入长度密钥分离：$k_\ell := F_k(\ell)$, 用 k_ℓ 作为 CBC-MAC 的密钥。不同长度下采用不同密钥，追加新块后长度变化，之前的标签无法利用。
  - 在开头添加长度：在CBC-MAC的明文m前添加一个长度块$|m|$。不同长度下消息有不同的初始块，追加新块后长度变化，之前的标签无法利用。
  - 加密末块输出（ECBC-MAC）：采用两个密钥k_1, k_2。用k_1和CBC-MAC计算出 t, 然后输出 $\hat{t} := F_{k_2}(t)$。输出结果被加密，之前的标签无法利用。



MAC填充（padding）

- 与加密类似，为了将消息长度与块长度对齐，MAC中也需要在消息中填充。为了安全性，需要保证填充是可逆的，即不同的消息在填充后也应该不同！
- $m_0 \neq m_1 \Rightarrow \mathsf{pad}(m_0) \neq \mathsf{pad}(m_1)$.
- ISO的填充标准：用“100...00”填充，并按需填充哑块。
- 如果不填充哑块，则会导致什么?
- CMAC（Cipher-based MAC）：不填充哑块，不加密最后一块的输出，密钥包括三个 k, k_1, k_2
  - k用于CBC-MAC；
  - k_1 和 k_2 与最后一块消息异或来阻止利用最后一块输出；
  - 用k_1 和 k_2 来区分是否添加了哑块。

哈希函数定义



Definition 7

A **hash function (compression function)** is a pair of PPT algorithms (Gen, H) satisfying:

- a key $s \leftarrow \mathsf{Gen}(1^n)$, s is **not kept secret**.
- $H^s(x) \in \{0,1\}^{\ell(n)}$, where $x \in \{0,1\}^*$ and ℓ is polynomial.

If H^s is defined only for $x \in \{0,1\}^{\ell'(n)}$ and $\ell'(n) > \ell(n)$, then (Gen, H) is a **fixed-length** hash function.

- 一个哈希函数 (压缩函数) 是一对PPT算法 (Gen, H) 满足以下条件:
  - 一个密钥 $s \leftarrow \mathsf{Gen}(1^n)$， s 不保密.
  - $H^s(x) \in \{0,1\}^{\ell(n)}$， 其中 $x \in \{0,1\}^*$ 且 ℓ 为多项式。
- 若 H^s 只在 $x \in \{0,1\}^{\ell'(n)}$ 上定义并且 $\ell'(n) > \ell(n)$，那么 (Gen, H) 是固定长度的哈希函数。
- 上面的定义说明，哈希函数将长消息转变为短消息。

安全哈希函数定义

碰撞

- **Collision** in H: $x \neq x'$ and $H(x) = H(x')$.
- **Collision Resistance**: infeasible for any PPT alg. to find.

The collision-finding experiment $\text{Hashcoll}_{\mathcal{A},\Pi}(n)$:

1. $s \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given s and outputs x, x'.
3. $\text{Hashcoll}_{\mathcal{A},\Pi}(n) = 1 \iff x \neq x' \wedge H^s(x) = H^s(x')$.

Definition 8

Π (Gen, H^s) is **collision resistant** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr[\text{Hashcoll}_{\mathcal{A},\Pi}(n) = 1] \leq \text{negl}(n).$$

哈希函数应用

- 文件指纹和去重（Fingerprinting 和 Deduplication）：识别一个文件，用于病毒指纹识别，去重复，P2P文件共享；
- 默克尔树（Merkle Tree）：构造多个文件或一个文件多个部分的指纹，从而定位有问题的文件或者文件中的部分；
- 口令哈希（Password Hashing）：$(salt, H(salt, pw))$，缓解明文口令泄漏风险；
- 密钥派生（Key Derivation）：从一个高熵（但不必均匀随机）的共享秘密中派生一个密钥；
- 承诺方案（Commitment Scheme）：将一个承诺与一份信息绑定，隐藏承诺的信息；例如，互联网上掷硬币。

课堂练习

H is CRHF. Is H' CRHF?

- $H'(m) = H(m) \oplus H(m)$
- $H'(m) = H(m) \| H(0)$
- $H'(m) = H(m \| 0)$
- $H'(m) = H(m[0, \ldots, |m| - 2])$
- $H'(m) = H(m) \oplus H(m \oplus 1^{|m|})$
- $H'(m) = H(m)[0, \ldots, |H(m)| - 2]$

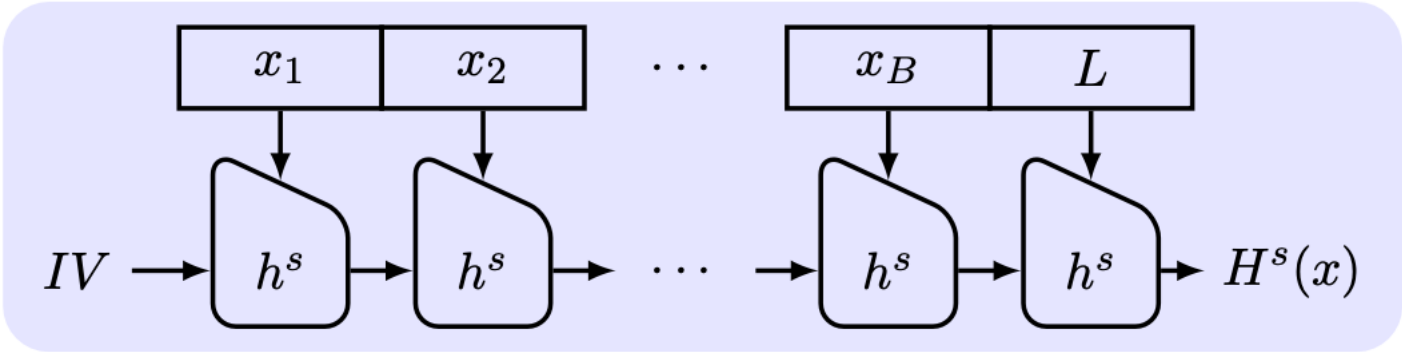
生日攻击

- 生日问题："如果一群人中有两个人的生日是同一天的概率有1/2，这群人数有多少？"。答案是23。这与我们平时的认知差异，也被称作"生日悖论"。具体计算见教材附件。
- 这个问题意味着哈希函数的输出需要足够长，否则敌手可能通过蛮力枚举来发现碰撞。
- 在现实攻击中，找到有意义的消息的碰撞对于攻击者来说更有价值。这对攻击者来说并不是难题，可以很容易的构造足够数量的、有意义的消息来实施攻击。对消息中一个单

How many different meaningful sentences are below?

It is **hard/difficult/challenging/impossible** to **imagine/believe** that we will **find/locate/hire** another **employee/person** having similar **abilities/skills/character** as Alice. She has done a **great/super** job.

MD变换（Merkle-Damgård Transform）



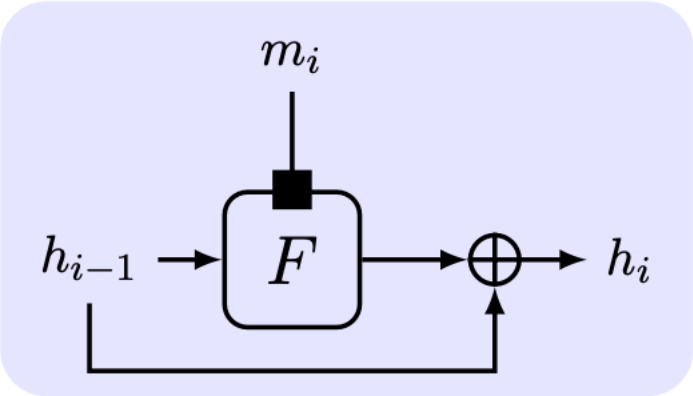
- 从定长哈希函数(Gen, h) (2ℓ bits $\to \ell$ bits, $\ell = \ell(n)$)构造变长哈希函数 CRHF (Gen, H) ：
  - Gen: 不变
  - H: 密钥 s 与串 $x \in \{0,1\}^*$, $L = |x| < 2^\ell$:
    - $B := \lceil \frac{L}{\ell} \rceil$ (块数)。 用0填充。 ℓ-位的块 $x_1, \ldots, x_B$。最后一块是长度 $x_{B+1} := L$, L 以 ℓ 位编码，这是必要的，因为只用0填充会导致不同消息的输入是一样的。
    - $z_0 := IV = 0^\ell$。 对于 $i = 1, \ldots, B+1$， 计算 $z_i := h^s(z_{i-1} \| x_i)$。

MD变换的安全性

- 定理：如果h是定长CRHF，那么H也是CRHF。
- 证明：思路是H上的碰撞意味着h上的碰撞，而h是不会被找到碰撞的。两个消息 $x \neq x'$ ，长度分别为 L 和 L' ，块数分别为B 和 B'，使得 $H^s(x) = H^s(x')$。 有两种情况：
  - $L \neq L'$: $z_B \| L \neq z_{B'} \| L'$；长度不同，意味着最后一个哈希函数$h$的输入不同，但输出相同，发现碰撞。
  - $L = L'$: $z_{i^*-1} \| x_{i^*} \neq z'_{i^*-1} \| x'_{i^*}$；长度相同，意味着中间某一块的输入不同，但输出相同，发现碰撞。
  - 因此，必定有 $x \neq x'$ 使得 $h^s(x) = h^s(x')$。
- 作业中有关于MD变换的变体的安全性分析问题。

通过PRP构造CRHF

Davies-Meyer (SHA-1/2, MD5)



$$h_i = F_{m_i}(h_{i-1}) \oplus h_{i-1}$$

- 定理：如果F是一个理想的加密方案，那么Davies-Meyer构造得到一个CRHF。
  - *注：理想的加密方案参考后面要学习的随机预言机模型。目前，没有找到F是强伪随机排列下该方法是CRHF的证明。*

- 对于这个定理不做严格证明，而是回答两个问题：
  - 如果 $h_i = F_{m_i}(h_{i-1})$ ，不与 h_{i-1} 异或，会如何？敌手尝试以相同的h_i和不同的m_i对F求逆。
  - 如果 F 不是理想的，而是 $\exists x, F_k(x) = x$，会如何？敌手输入不同m_i，但都得到0；

哈希MAC

- 有了CRHF，一个自然的想法是：先将任意长度消息哈希，然后通过PRF对哈希值做MAC，实现任意长度消息MAC。$F_k(H(m))$
- 这个方案的安全性分两种情况分析：当不同消息得到相同哈希值时，这意味着碰撞发生；否则，意味着MAC标签被伪造。

Construction 13

$(\widetilde{\mathsf{Gen}}, H)$ *is a CRHF.* $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *is a fixed-length MAC.*

- $\mathsf{Gen}'(1^n)$: (k, s). $s \leftarrow \widetilde{\mathsf{Gen}}, k \leftarrow \mathsf{Gen}$.
- $\mathsf{Mac}'_{s,k}(m)$: $t := \mathsf{Mac}_k(H^s(m))$.
- $\mathsf{Vrfy}'_{s,k}(m, t)$: $1 \iff \mathsf{Vrfy}_k(H^s(m), t) = 1$.

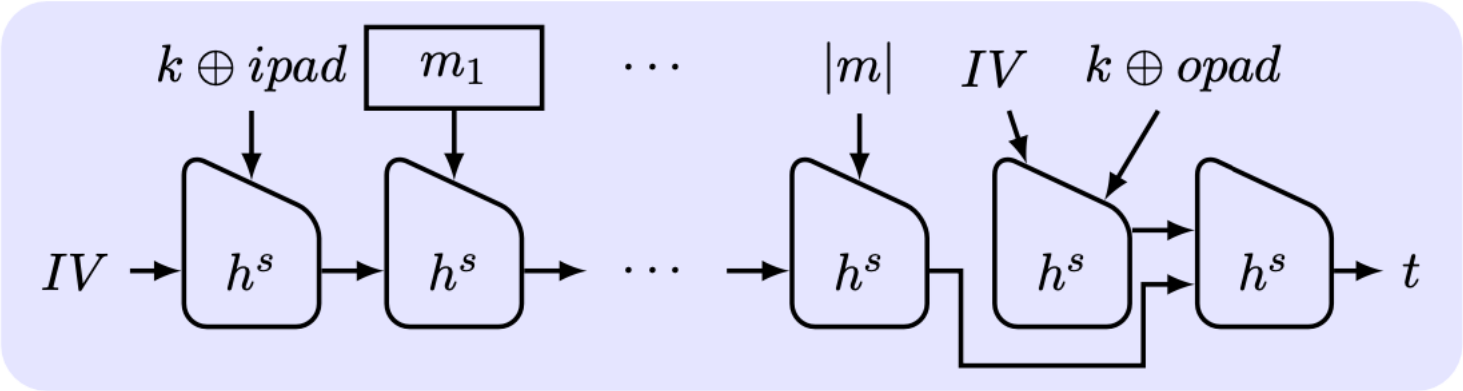
Theorem 14

The construction is a secure MAC for arbitrary-length messages.

Idea of proof: if the adversary has forged a tag on the "new message" m^*, then

- Case 1: If there is a queried messages m such that $H^s(m) = H^s(m^*)$, then there is a collision in H^s.
- Case 2: If there is no queried messages m such that $H^s(m) = H^s(m^*)$, then the adversary has forged a valid tag on the "new message" $H^s(m^*)$ for MAC.

HMAC



- 以MD变换为基础构造一个安全的MAC。在开头和结尾以两个不同密钥作为哈希函数输入。
- 不需要修改哈希函数。

HMAC

- HMAC是基于NMAC的改进，是工业标准（RFC2104），HMAC比CBC-MAC更快；
- 但不要自己实现！比较按字节匹配，通过观察函数返回时间可以判断相同字节的数量，从而按字节猜测标签内容。
  - 在Xbox 360中，相邻字节上被验证拒绝的时间差有2.2毫秒.

Verification timing attacks

Keyczar crypto library (Python):

```
def Verify(key, msg, sig_bytes):
        return HMAC(key, msg) == sig_bytes
```

The problem: implemented as a byte-by-byte comparison

In Xbox 360, a difference of 2.2 milliseconds between rejection times of i vs. i + 1 bytes.

Don't implement it yourself

信息论安全的MAC

- 不可能达到“完美的、不可伪造的”MAC，因为算力无限制的敌手可以至少以$1/2^{|t|}$的概率输出一个有效的标签。为此，对敌手查询MAC预言机的次数需要加以限制，下面分析只允许敌手查询一次MAC预言机的情况。

Definition 16

A MAC Π is **one-time ε-secure** if $\forall$ PPT $\mathcal{A}$:

$$\Pr[\mathsf{Macforge}^{1-\mathsf{time}}_{\mathcal{A},\Pi} = 1] \leq \varepsilon.$$

- 任意 ℓ次 2^{-n}-安全 MAC 需要密钥长度至少为 $(\ell+1)\cdot n$.
- 定理：令 Π 为一次 2^{-n}-安全 MAC，其中所有密钥长度相同。那么，密钥必须具有$2n$长度。
- 证明：直觉上，每对消息和标签成立需要2^n个密钥，才能保证 2^{-n}-安全。一共2对，需要2^{2n}

本节小结

- 认证意味着存在不可伪造
- 用PRF来实现安全MAC
- 用带密钥的CRHF来实现安全MAC
- 信息论MAC安全需要非常、非常长的密钥