Message Authentication Codes

今天梳理Katz&Lindell的《Introduction to Modern Cryptography》中《Message Authentication Codes》这一章，感觉看了几天没有深入理解？看完很快就忘了，必须整理一下。

4.1 Message Integrity 消息完整性

目的： 防止主动型敌手(active adversary)在消息传送得过程中注入信息或修改信息。虽然上一章讲到了加密，但是加密并不提供任何完整性，下面用两个例子说明：

- 流密码加密
    - 将伪随机串与明文异或得到密文，由于明文长度与密文长度相同且位置一一对应，所以如果将密文中的某一位翻转，如从0变为1，那么在解密后的明文m'中的相同位置与原m不同。例如Alice要从银行取钱，她向银行发送数额abcd，Eve从中截获了相应的密文c，对密文的最后一位进行翻转将改变1￥，对首位进行翻转将改变1000￥。如果Eve从某种途径知道了和取钱数额相关的信息，那么修改（modification）将产生巨大影响。
- 块密码加密
    - 对EBC的一个块进行比特翻转可以影响该块的解密内容，通过改变密文的顺序可以影响解密后的明文各块的顺序
    - 由于CBC只有第一块直接用到IV，没有用其他密文，因此改变IV的*j*th位会改变第一块对应的明文的jth位，CBC的第一块可以被任意修改

由于在加密中，对于一定长度的每个串都是有效密文，因此如果敌手伪造相同长度的密文冒充通信双方，即使敌手并不知道相应的明文是什么，这种攻击也应该被阻止。

总结：加密并不保证信息完整性，怎么办，再设计一套像加密一样的机制。

4.2 MACs的定义

由于加密机制并不解决消息完整性的问题，因此需要一个额外的机制来确保通信双方知道他们的消息并未被篡——MACs。

MAC的目的：阻止敌手在接收方监测不到收到的信息与发送方最初发送的信息不同的前提下篡改信息或者注入新的信息。只有通信双方有一些特有秘密的前提下，这种修改才能被发现。（如：双方约定第2、3位一定是0，敌手如果在这里修改了就会被发现）

在私钥加密的情况下，MAC有两种经典应用场景：保证双方通信的完整性；保证时空通信的完整性（取回之前在硬盘中存的东西）。

The Syntax of a MAC

先说明MAC是什么以及如何使用再给出具体定义。

1. 在通信之前，双方首先产生并共享密钥k
2. 发送方基于信息m和密钥k用**标签生成算法MAC**计算标签t，$t \leftarrow Mac_k(m)$；发送(m,t)给接收者
3. 接收方收到(m,t)运行**Vrfy算法**验证t

DEFINITION4.1 MAC包含(Gen,Mac,Vrfy)三个概率多项式算法，且满足以下条件:

1. **密钥产生算法Gen**以1^n作为输入，输出密钥k（均匀分布，随机选择），|k|≥n。
2. **标签产生算法MAC**以密钥k和消息$m \in \{0,1\}^*$作为输入，输出标签t。该算法可能是随机的，所以记作$t \leftarrow Mac_k(m)$。

3. **确定性验证算法Vrfy**以密钥k，消息m和标签t作为输入，输出比特b。b=1意味着标签t是有效的；b=0意味着t是无效的。将其记作$b := Vrfy_k(m,t)$
4. 正确性：对于每个n，k，m有$Vrfy_k(m, Mac_k(m)) = 1$.

canonical verification 对于确定性MAC算法，一般的验证方法是再计算一遍tag验证是否相等。

MAC的安全性

对信息验证码安全的直观定义是，不存在有效敌手能够对双方没有发送过的新的信息m产生一个有效的tag。正如其他的安全性定义一样，为了正式定义这个概念，需要定义敌手的能力以及什么是“break”。

- 敌手的能力。这里考虑PPT敌手，所以问题进一步变为：敌手与通信双方接触的模型是什么？是passive还是active？在信息验证的背景中，敌手可能会观察到通信双方所有的(m,t)，也可能会间接影响发送者要发送的内容。为了模型化以上情况，允许敌手可以查询他所选择的信息的相应tag：敌手可以不断地向oracle $Mac_k(\cdot)$查询，得到相应的tag，并且可以根据当前所得tag，选择下一个要查询的m(adaptive)。
- “break”。存在性伪造，只要敌手能够伪造新消息(任意消息)的tag，就算攻破。 不可攻破的MAC是说“在自适应选择信息攻击下存在不可伪造”：敌手不能伪造任何信息的有效tag。

信息认证实验$Mac - forge_{\mathcal{A},\Pi}(n)$：

1. 运行$Gen(1^n)$产生密钥k。
2. 给敌手$\mathcal{A}$输入1^n以及查询$Mac_k(\cdot)$的权利。敌手$\mathcal{A}$最终输出(m,b)。用$\mathcal{Q}$表示所有$\mathcal{A}$查询的信息集合。
3. 如果$Vrfy_k(m,t) = 1$且$m \notin \mathcal{Q}$，$\mathcal{A}$获得成功，实验输出1。

如果不存在有效敌手能够以不可忽略的概率在上述实验中取得成功，MAC就是安全的。

DEFINITION4.2 如果对于任意PPT敌手$\mathcal{A}$，存在可忽略函数negl使得$Pr[Mac - forge_{\mathcal{A},\Pi}(n) = 1] \leq negl(n)$MAC(Gen,Mac,Vrfy)在自适应选择信息攻击的条件下是存在不可伪造的（安全的）。

replay attacks.虽然MAC保证了不能伪造对新信息的标签，但是存在一种重复攻击。攻击者可以将一方曾经发送过的(m,t)直接发送给另一方，而且可以通过验证。比如Alice向BOB支取1000￥，对应的(m,t)被eve截获，之后eve将该(m,t)重复10次发送给bob，，bob会给Alice10000￥。

更强的不可伪造 一个安全的MAC只是保证了敌手不能产生未查询过信息的tag，但是不能保证对于查询过得信息mi，敌手可以产生一个不同于之前的有效tag，ti'≠ti。

$Mac - sforge_{\mathcal{A},\Pi}(n)$：

1. 运行$Gen(1^n)$产生密钥k。
2. 给敌手$\mathcal{A}$输入1^n以及查询$Mac_k(\cdot)$的权利。敌手$\mathcal{A}$最终输出(m,b)。用$\mathcal{Q}$表示所有$\mathcal{A}$查询的信息标签对集合。
3. 如果$Vrfy_k(m,t) = 1$且$(m,t) \notin \mathcal{Q}$，$\mathcal{A}$获得成功，实验输出1。

如果不存在有效敌手能够以不可忽略的概率在上述实验中取得成功，MAC就是安全的。

DEFINITION4.3 如果对于任意PPT敌手$\mathcal{A}$，存在可忽略函数negl使得$Pr[Mac - sforge_{\mathcal{A},\Pi}(n) = 1] \leq negl(n)$MAC(Gen,Mac,Vrfy)在自适应选择信息攻击的条件下是strongly secure。

- 如果一个安全的MAC使用经典的验证，那么它也是强安全的。证明

验证查询 验证也可以像查询的$Mac_k(\cdot)$一样有oracle。

潜在的时间攻击　　敌手将m/t对发送给接收者，可以将接收者当做验证oracle，不仅可以知道接收者对m/t接收还是拒绝，还可以消耗接收者用来判断的时间。假如验证像C中的strncmp一样，对t和t'一位一位的进行比较，只要出现一次不同就不再比较，视为t'无效，因此拒绝所花费的时间依赖于第一个不相同的比特位的位置。　　假设攻击者知道m对应的t的前i位，通过不断发送(m,t0),(m,t1)...(m,t255)，最终攻击者可以得到m对应的t。因此MAC的验证应该考虑使用与时间无关的比较方式：比较所有比特。

总结：定义了MAC的机制以及MAC应该满足的安全性要求(模型化敌手能力以及什么是攻破)

4.3 构造安全的MAC

伪随机函数是构造MAC的一个自然的工具。如果通过应用伪随机函数到m来得到t，那么伪造一个之前没有验证过的消息意味着能够正确猜测出伪随机函数在一个新的点的值。对于输出长度为n的随机函数来说，这个概率是2^{-n}，伪随机函数的概率只比这大negl。

4.3.1 固定长度的MAC

根据伪随机函数，构造4.5给出了一个安全的固定长度的MAC：

Consrtuction4.5

让F是一个(length preserving)的伪随机函数，定义对于消息长度为n的MAC：

- **MAC**：输入密钥$k \in \{0,1\}^n$，消息$m \in \{0,1\}^n$，输出标签$t := F_k(m)$。
- **Vrfy**：输入密钥$k \in \{0,1\}^n$，$m \in \{0,1\}^n$，$t \in \{0,1\}^n$，如果$t = F_k(m)$输出1，否则输出0.

Theroem 4.6

如果F是伪随机函数，构造4.5是一个消息长度为n的安全的固定长度的MAC。

证明：关于伪随机函数的证明模板：首先用真随机函数代替伪随机函数，然后证明这对于攻击者成功的概率影响是可忽略的；再具体分析使用真随机函数的方案。

4.3.2 定义域扩展的MAC(可以处理任意长度的消息)

仅仅处理固定长度消息的MAC在实际中的应用有很多限制，接下来根据固定长度(n)的MAC来构造任意长度的MAC。接下来的构造并不是实用和有效的，在这里仅以教学的目的展示。

Constuction 4.7

令$\Pi' = (Mac', Vrfy')$是对于长度为n的消息的固定长度的MAC，定义MAC:

- **MAC**：输入密钥$k \in \{0,1\}^n$，消息$m \in \{0,1\}^*$长度$l < 2^{n/4}$。将m分为$m_1, \ldots, m_d$共d块每块长度为$n/4$(如果必要，最后一块以0s填充)。随机选择信息标识符$r \in \{0,1\}^{n/4}$。

  对于i=1,...,d，计算$t_i := Mac'_k(r||l||i||m_i)$，其中r和l以n/4的长度编码。输出标签集合$t :=< r, t_1, \ldots, t_d >$
- **Vrfy**：输入密钥$k \in \{0,1\}^n$，$m \in \{0,1\}^*$长度$l < 2^{n/4}$，$t =< r, t_1, \ldots, t_{d'} >$，将m分为$m_1, \ldots, m_d$共d块每块长度为$n/4$(如果必要，最后一块以0s填充)。当且仅当$d = d'$且$Very((r||l||i||m_i), t_i) = 1, 1 \le i \le d$，时，输出1.

Theroem 4.8

如果$\Pi' = (Mac', Vrfy')$是对于长度为n的消息的固定长度的安全的MAC，构造4.7就是对于任意长度的安全的MAC。

4.4 CBC-MAC