> 教材第十二章

- 解决消息有没有被篡改的问题
- 两类消息认证码：HMAC（基于哈希函数）、CMAC（基于对称加密）
- HMAC 是带密钥的哈希函数
- 基于**密码原语**（加密、哈希函数、数字签名）的组合
- 当无需保证机密性时，应该避免对整个消息进行加密
- 对 MAC 的穷举攻击由于需要知道 <消息 - MAC> 对，所以这种攻击会比对 Hash 函数的攻击更难

消息认证函数

- Hash 函数：它是将任意长的消息映射为定长的 Hash 值的函数，以该 Hash 值作为认证符
- 消息加密：对整个消息加密后的密文作为认证符
- 消息认证码（MAC）：它是消息和密钥的函数，它产生定长的值，以该值作为认证符

上面三种函数对安全性的保障不同。

利用对称加密进行消息认证存在一些问题：如果消息 M 可以是任何的位模式，那么接收方无法确定接收到的消息是合法明文的密文。（也就是说，随便伪造一个密文发给接收方，接收方也能解密，但解密出来的消息可能是没有意义的，所以只能说在一定程度上提供消息认证）。

基于哈希函数的 MAC：HMAC

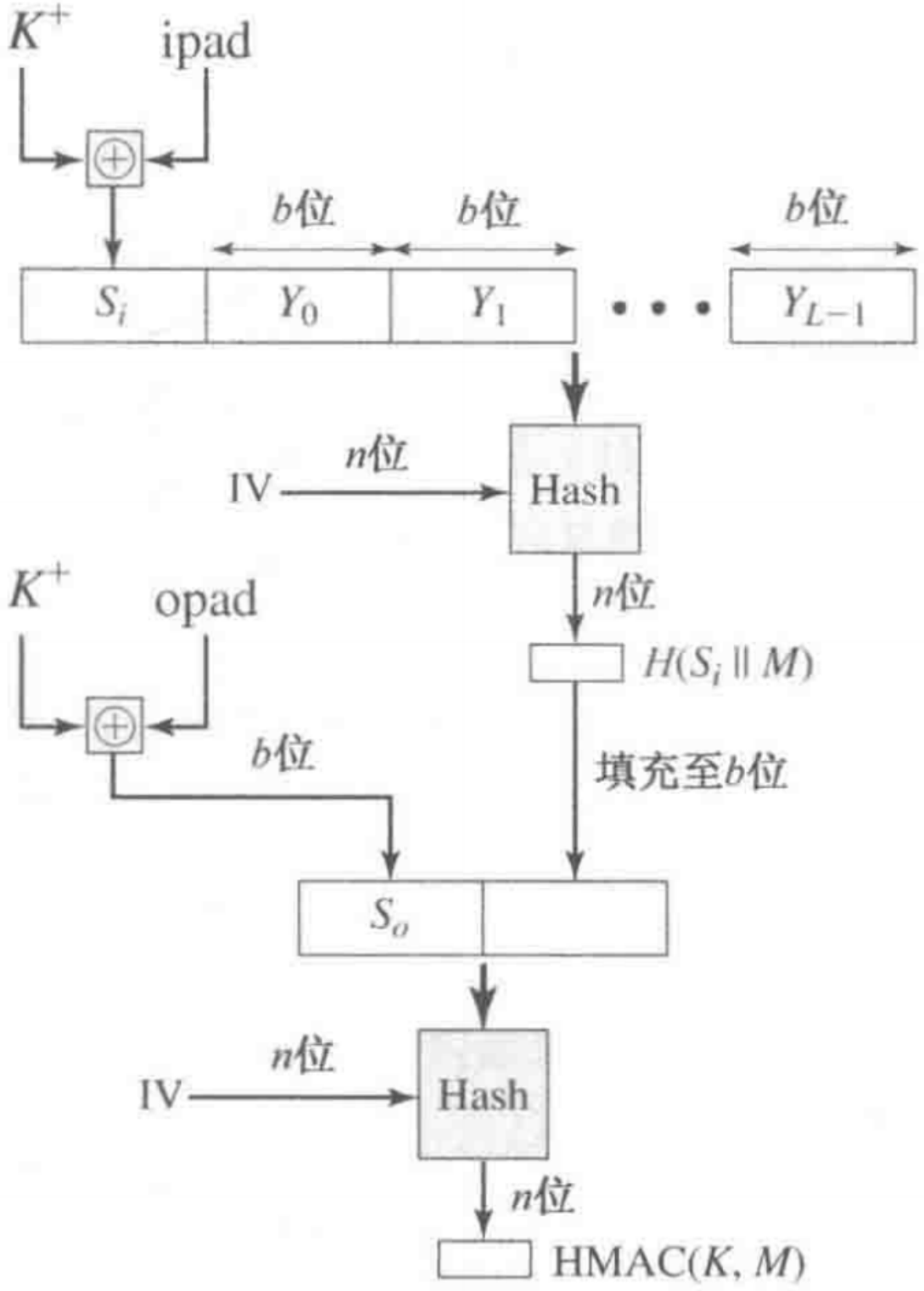


图 12.5　HMAC 的结构

IV 为 Hash 函数输入的初始值

$$\mathrm{HMAC}(K, M) = \mathrm{H}\left[(K^+ \oplus \mathrm{opad}) \| \mathrm{H}\left[(K^+ \oplus \mathrm{ipad}) \| M\right]\right]$$

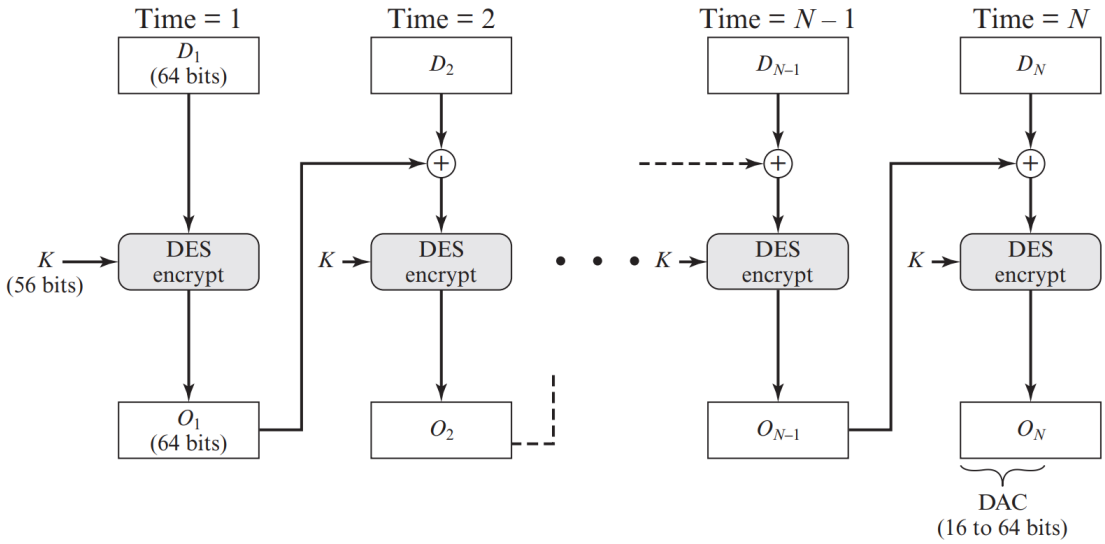
可以预计算以下两部分：

$$\mathrm{f}\left(IV, (K^+ \oplus \mathrm{ipad})\right)$$
$$\mathrm{f}\left(IV, (K^+ \oplus \mathrm{opad})\right)$$

基于分组密码的 MAC: DAA 和 CMAC

数据认证算法（DAA）

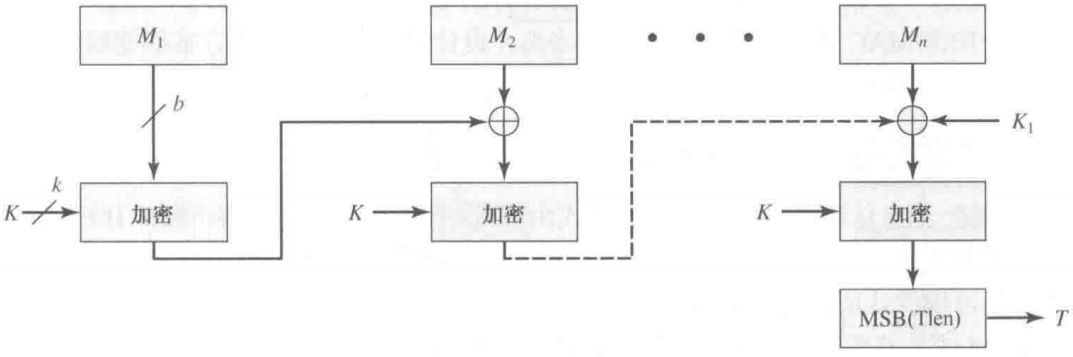
DAA 采用 DES 运算的密文块链接（CBC）方式。



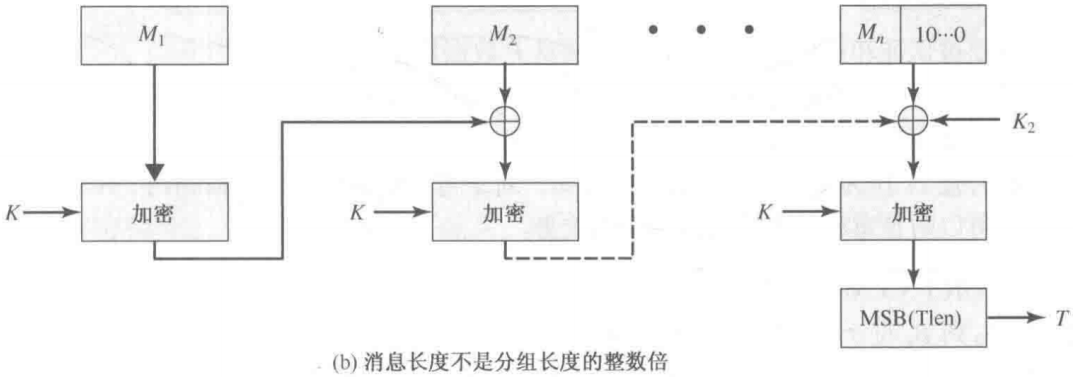
DAA 有如下的限制：仅能处理固定长度为 mn 的消息，其中 n 是密文分组的长度，m 是一个固定的正整数。例如，给定一个消息分组 X 的 CBC MAC 码，如 $T = MAC(K, x)$，则攻击者马上就知道两个消息分组 $X||(X \oplus T)$ 的 CBC MAC 码，因为这还是 T。

基于密码的消息认证码（CMAC）

DAA 的限制可以使用三个密钥来克服: 一个长度为 k 的密钥 K，用在密文分组链接的每一步，两个长度为 b 的密钥，其中 b 是密钥长度。



(a) 消息长度是分组长度的整数倍



(b) 消息长度不是分组长度的整数倍

图 12.8　基于密码的消息认证码（CMAC）

思考题

12.1

> 消息认证是为了对付哪些类型的攻击?

伪装，内容修改，序列修改，时序修改

12.2

> 消息认证或数字签名方法有哪两层功能?

1. 下层有产生认证符的函数（认证符是用来产生认证消息的值）
2. 上层协议将该函数作为原语，使接收方可以验证消息的真实性

12.3

> 产生消息认证有哪些方法?

消息加密，消息认证码，哈希函数。

12.4

> 对称加密和错误控制码（帧校验序列）一起用于消息认证时，这两个函数必须以何种顺序执行?

先附加错误控制码，再加密。若错误控制码在外部，那么攻击者可以构造具有正确错误控制码的消息，虽然攻击者不知道解密后的明文是什么，但他可造成混淆并破坏通信。

12.5

> 什么是消息认证码?

使用密钥和消息生成的一个固定长度的短消息块。

12.6

> 消息认证码和单向 Hash 函数之间的区别是什么?

Hash 函数是不带密钥的，Hash 函数本身不提供消息认证。MAC 使用密钥计算用于认证的码。

12.8

> 为了攻击 MAC 算法，必须要恢复密钥吗?

不需要。可以攻击 MAC 值，这种攻击的目的是对给定消息产生有效的 MAC 或者对给定 MAC 产生相应的消息。

其他

> 使用 $Hash(key||Message)$ 生成消息认证码安全吗？为什么?

不安全，会受到长度扩展攻击。