



ch21-网络安全(4)

—— 密码散列函数

南京大学计算机系黄皓教授

2007年12月18日 星期五



参考文献

1. W. Stallings (杨明等译), 编码密码学与网络安全-原理与实践, 电子工业出版社。2001年4月。
2. Wenbo Mao(毛文波), 现代密码学理论与实践, 电子工业出版社, 2004年7月。



内容

1. 数据完整性的基本概念;
2. 散列函数;
3. 密码散列报文鉴别码

1. 基本概念



数据完整性需求

- 我们关于开放通信网络的脆弱性做了一个理想的、标准的假设：

所有的通信都经受一个称为**Malice**的攻击者，他可以随意地窃听、截取、重发、修改、伪造或插入消息。

当**Malice**插入了修改过的或伪造的消息，他将试图欺骗目标接收者，使其相信该消息来自某个其他合法的主体。

我们需要一种机制，使得消息的接收者可以验证该消息确实是来自所声称的消息源，且在传输的过程中未受到未授权方式修改。数据完整性就是抗击对消息未授权修改的安全服务。



差错控制与数据完整性

■ 检错码

- 检测由于通信的缺陷而导致消息发生错误的方法。
- 通过这种编码加入的冗余度使得消息的接收者可以采用极大似然检测器来判定接收到的码字应该译为哪条消息。

■ 数据完整性

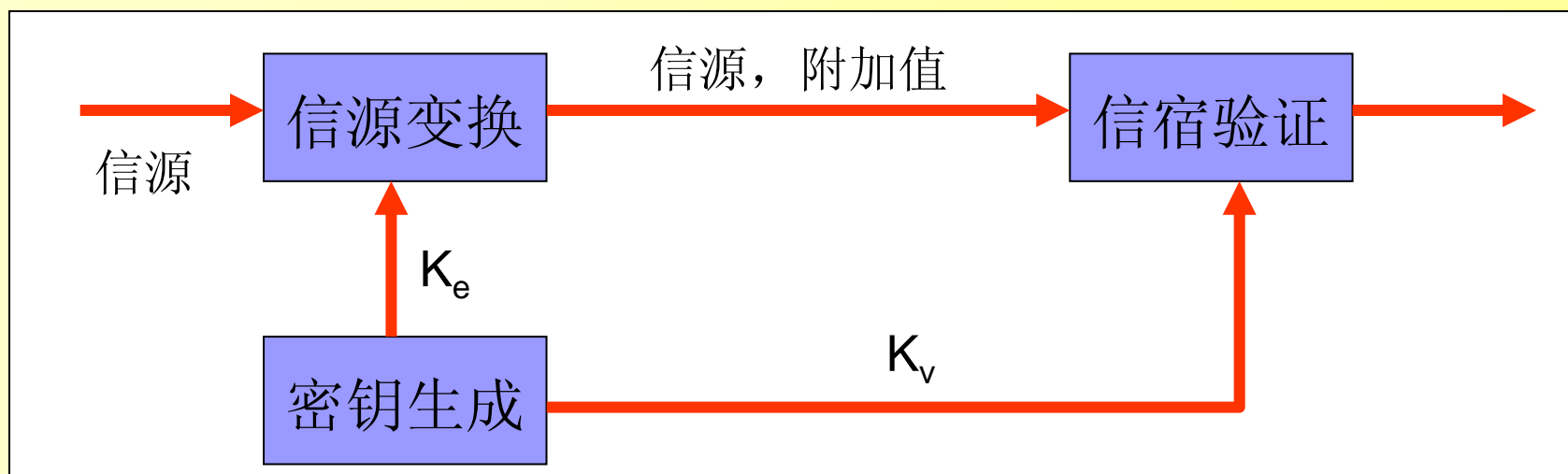
- 消息的发送者通过编码为消息增加一些冗余来生成一个“校验值”，并将该校验值附在消息之后；
- 消息的接收者根据与发送者协商好的一系列规则，利用附加的校验值来检验所接收到的消息的正确性。
- 使得加入的校验值在整个校验值空间中尽可能地均匀分布，这就使得攻击者伪造一个有效校验值的概率达到最小。



篡改检测码

- 设 M 是任意的信息， K_e 为编码密钥， K_v 是验证密钥，
- 篡改检测码： $MDC = f(K_e, M)$ (Manipulation Detection Code)
- 篡改检测码的验证：

$$G(K_v, M, MDC) = \begin{cases} \text{true, 概率为 1, 如果 } MDC = f(K_e, M) \\ \text{false, 压倒性概率, 如果 } MDC \neq f(K_e, M) \end{cases}$$





非对称 — 数字签名体制

- 一个明文消息空间 M
- 一个签名空间 S
- 一个签名密钥空间 K 和一个认证密钥空间 K' 。
- 一个有效的密钥生成算法 $\text{Gen}: N \rightarrow K \times K'$
- 一个有效的签名算法: $\text{Sign}: M \times K \rightarrow S$
- 一个有效的验证算法: $\text{Verify}: M \times S \times K \rightarrow \{\text{true}, \text{false}\}$
- $\text{sk} \in K, m \in M, s = \text{Sign}_{\text{sk}}(m)$

$$\text{Verify}_{\text{pk}}(m, s) = \begin{cases} \text{True, 概率为 } 1, & \text{如果 } s = \text{Sign}_{\text{sk}}(m) \\ \text{false, 压倒性概率,} & \text{如果 } s \neq \text{Sign}_{\text{sk}}(m) \end{cases}$$



RSA签名体制

■ 密钥建立

- 用户Alice的公钥为 (N, e) , (N, p, q, d) 为私钥。
- 其中 $N=pg$, p 和 g 是两个长度差不多的大素数, e 是满足 $\gcd(e, \phi(N))=1$ 的整数。整数 d , 满足 $ed = 1 \bmod (\phi(N))$ 。

■ 签名生成

- 消息的 m 的签名:
- $S = \text{Sign}_d(m) \leftarrow m^d \bmod (N)$

■ 签名验证

- 设Bob是验证者, 他知道公钥 (N, e) 属于Alice, 给定一个消息-签名对 (m, s) 。
- Bob的验证过程为

$$\text{Verify}_{(N, e)}(m, s) = \text{True}, \text{ 如果 } m = s^e \bmod N$$



ElGamal签名体制

■ 生成密钥对

- 生成一个大的随机素数 p 和整数 $\text{mod } p$ 的乘法群 Z_p^* 的生成元 α ;
- 选取一个随机整数 s ($1 \leq s \leq p-2$), 计算 $\beta = \alpha^s \text{ mod } p$;
- 公钥 (p, α, β) , 私钥 s 。

■ 对信息 m 签名

- 选取一个随机整数 k ($1 \leq k \leq p-2$), 计算 $X = \alpha^k \text{ mod } p$
- 从方程 $m = (s \cdot X + k \cdot Y) \text{ mod } (p-1)$ 中求解 Y ;
- 签名为 (X, Y) ;

■ 验证签名

- 验证等式: $\beta^X \cdot X^Y = \alpha^m \text{ mod } p$
- $(\alpha^s)^X \cdot (\alpha^k)^Y = \alpha^{s \cdot X + k \cdot Y} \text{ mod } p$
 $= \alpha^{m + t \cdot (p-1)} \text{ mod } p$
 $= \alpha^m \cdot \alpha^{t \cdot (p-1)} \text{ mod } p$
 $= \alpha^m \text{ mod } p$

2. 散列函数



散列函数的目标

- 散列函数的目的是为文件、报文或其他的分组数据产生“指纹”。要用于报文鉴别，散列函数 H 必须具有如下性质：
 1. H 能用于任何大小的数据分组。
 2. H 产生定长输出。
 3. 对任何给定的 x ， $H(x)$ 要相对易于计算，使得硬件和软件实现成为实际可行。
 4. 对任何给定的码 $H(x)$ ，从 $H(x)$ 计算 x ，在计算上是不可行的。这就是所谓的单向性质。
 5. 对任何给定的分组 x ，寻找不等于 x 的 y ，使得 $H(y)=H(x)$ 在计算上是不可行的。
 6. 寻找对任何的 (x,y) 对使得 $H(x)=H(y)$ 在计算上是不可行的。



生日悖论

- 在k个人中，至少有两个人的生日相同的概率 $p(365, k)$ 大于0.5，问：k的最小值是多少？
- 假设一年365天， $n=365$ ，k个人中没有相同的概率

$$p = \frac{n(n-1)\cdots(n-k+1)}{n^k}$$

$$p(n, k) = 1 - \frac{n(n-1)\cdots(n-k+1)}{n^k} = 1 - \left[\left(1 - \frac{1}{n}\right) \times \left(1 - \frac{2}{n}\right) \times \cdots \times \left(1 - \frac{k-1}{n}\right) \right]$$

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

- $(1-x) \leq e^{-x}$
- $p > 1/2$ 的最小k是多少？
- $1 - e^{-k(k-1)/2} = 1/2$; $k \approx (2(\ln 2)n)^{1/2} \approx 1.18 n^{1/2} \approx 23$ (如果 $n=365$)



生日攻击

- 假定函数 H 有 2^m 个可能的输出（每个输出的长度为 m 比特）。
- H 接收一个随机输入产生集合 X ，在使用另外一个随机输入产生集合 Y 。
- 两个集合要取多少个值，才能使两个集合至少有一个匹配：

存在 $x \in X$, $y \in Y$, 使得 $H(x) = H(y)$

$$k = 2^{m/2}$$



构造意义相同的冲突报文

Dear Anthony,

This letter is

you to

Mr.

P.

to introduce

Alfred

I am writing

to you

--

--

new

chief

Barton, the

jewellery

buyer for

newly appointed

senior



散列函数的应用

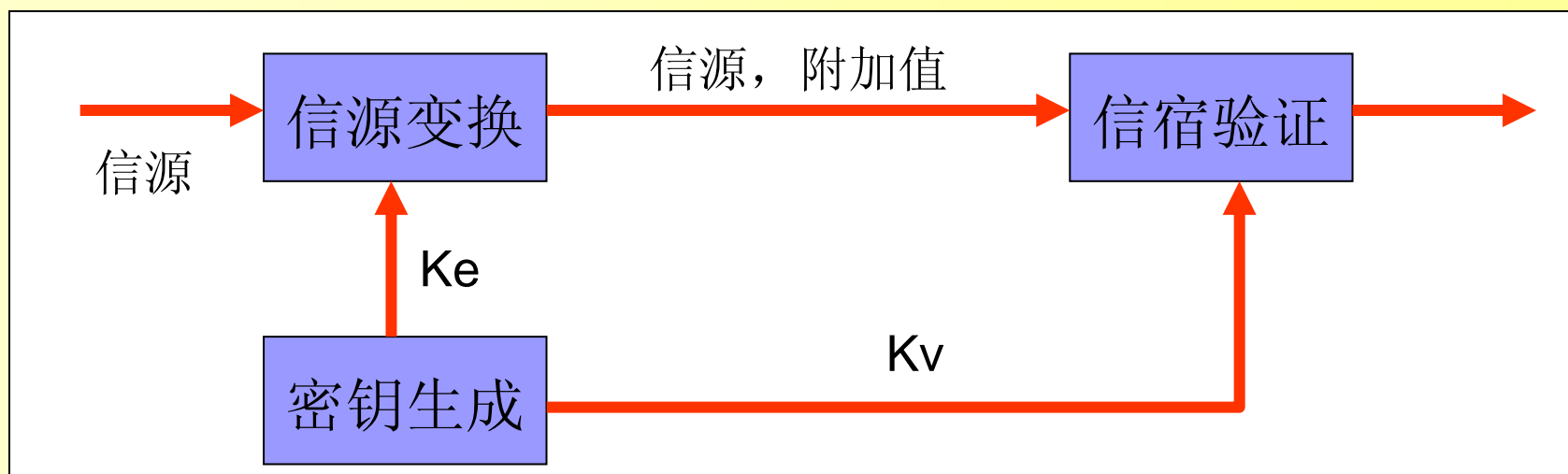
- $H(x)$ 是一个散列函数。
- $H(k \parallel M)$ 可以作为消息 M 的消息鉴别码。
- 攻击者要将 M 篡改为 M' ，则要计算(猜测) $H(k \parallel M')$ ，这在计算上是不可能的：以压倒性概率不可能。
- 如果攻击者找到 M' 使得 $H(k \parallel M') = H(k \parallel M)$
- 则他将消息 $\{M \parallel H(k \parallel M)\}$ 篡改成 $\{M' \parallel H(k \parallel M)\}$ 接收者却无法通过验证发现。



篡改检测码

- 设M是任意的信息，Ke为编码密钥，Kv是验证密钥，
- 篡改检测码：MDC = f(Ke, M) (Manipulation Detection Code)
- 篡改检测码的验证：

$$G(K_v, M, \text{MDC}) = \begin{cases} \text{true, 概率为 1, 如果 } \text{MDC} = f(\text{Ke}, M) \\ \text{false, 压倒性概率, 如果 } \text{MDC} \neq f(\text{Ke}, M) \end{cases}$$





RSA的数字签名方案

- 需要签名的消息 M ，散列函数 H ， $\langle K_u, K_R \rangle$ 公钥对。
- 签名：
 - $H(M)$
 - $s = \text{RSA}(K_R, H(M))$
- 验证签名：
 - $H(M) \stackrel{?}{=} \text{RSA}(K_u, s)$



非对称 — 数字签名体制

- 一个明文消息空间 M
- 一个签名空间 S
- 一个签名密钥空间 K 和一个认证密钥空间 K' 。
- 一个有效的密钥生成算法 $\text{Gen}: N \rightarrow K \times K'$
- 一个有效的签名算法: $\text{Sign}: M \times K \rightarrow S$
- 一个有效的验证算法: $\text{Verify}: M \times S \times K \rightarrow \{\text{true}, \text{false}\}$
- $\text{sk} \in K, m \in M, s = \text{Sign}_{\text{sk}}(m)$

$$\text{Verify}_{\text{pk}}(m, s) = \begin{cases} \text{True}, & \text{概率为 } 1, & \text{如果 } s = \text{Sign}_{\text{sk}}(m) \\ \text{false}, & \text{压倒性概率}, & \text{如果 } s \neq \text{Sign}_{\text{sk}}(m) \end{cases}$$



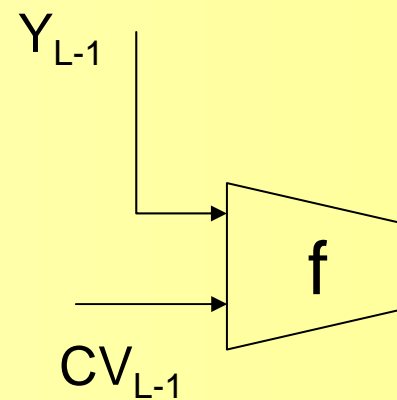
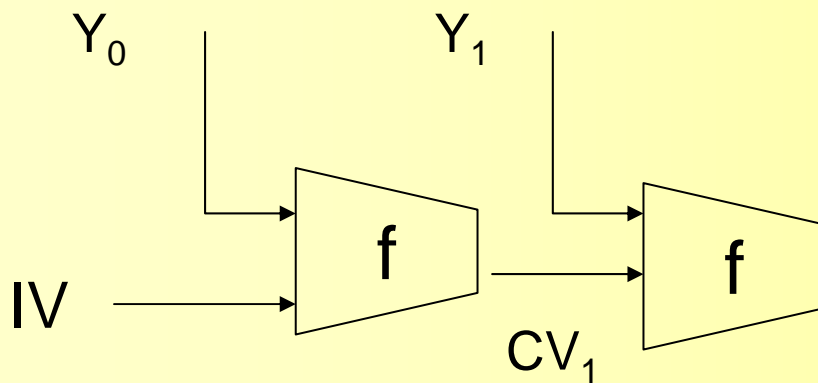
攻击的例子

- Alice准备一份合同的两个版本，一份对Bob有利，一份将使他破产；
- Alice对这两种版本的每一份都作一个细微的改变（例如，在回车之前加一个或二个空格，通过在n行中作修改，则可以得到 2^n 种不同的文件）。
- Alice比较这两种版本的散列值，找出相匹配的一对(N,M)，其中对M对Bob有利,N将使他破产，并且 $H(M)=H(N)$ 。
- Alice请求Bob对合同M签名： $M \parallel E(K_{R_{Bob}}, H(M))$
- Alice 在适当的时候向法官证明Bob签署过合同：
 $N \parallel E(K_{R_{Bob}}, H(N)) (= N \parallel E(K_{R_{Bob}}, H(M)))$



散列函数的一般结构

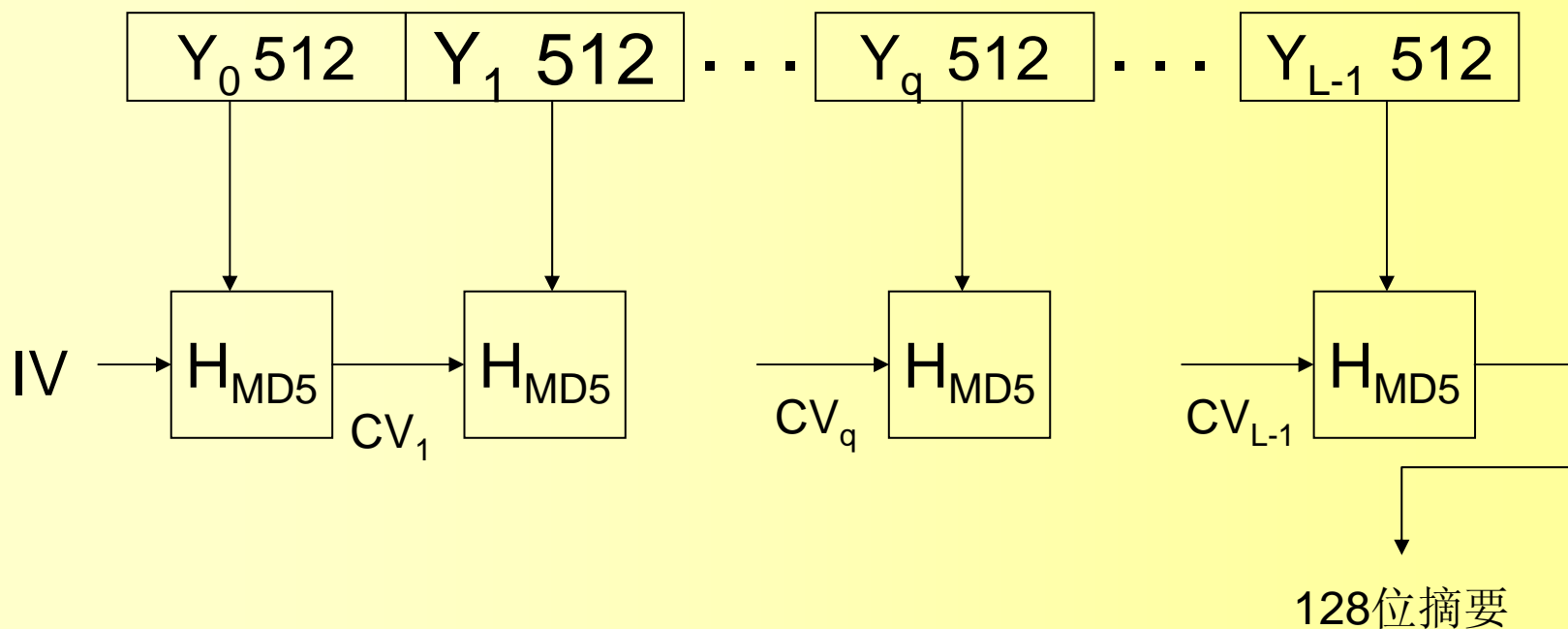
- $CV_0 = IV;$
- $CV_i = f(CV_{i-1}, Y_{i-1}); \quad 1 \leq i \leq L$
- $H(M) = CV_L$





2.2 MD5报文摘要算法 (Rivest, RFC1321)

报文	$L \cdot 512 - 64$ -填充长度	填充 1-512	长度 64
----	--------------------------	-------------	----------

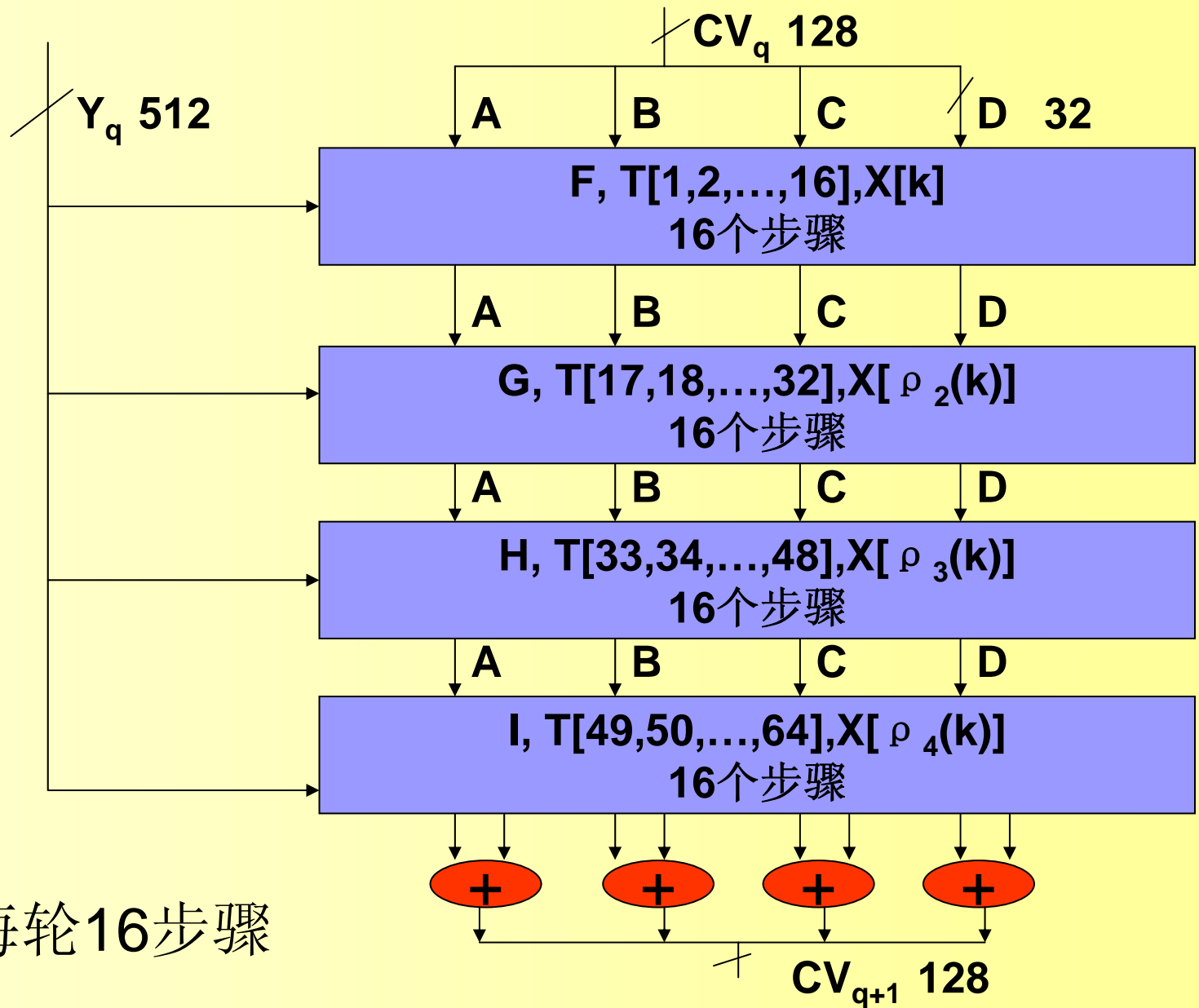




- MD5: RFC1321
- MD4: RFC1320
- MD2: RFC1319



H_{MD5}

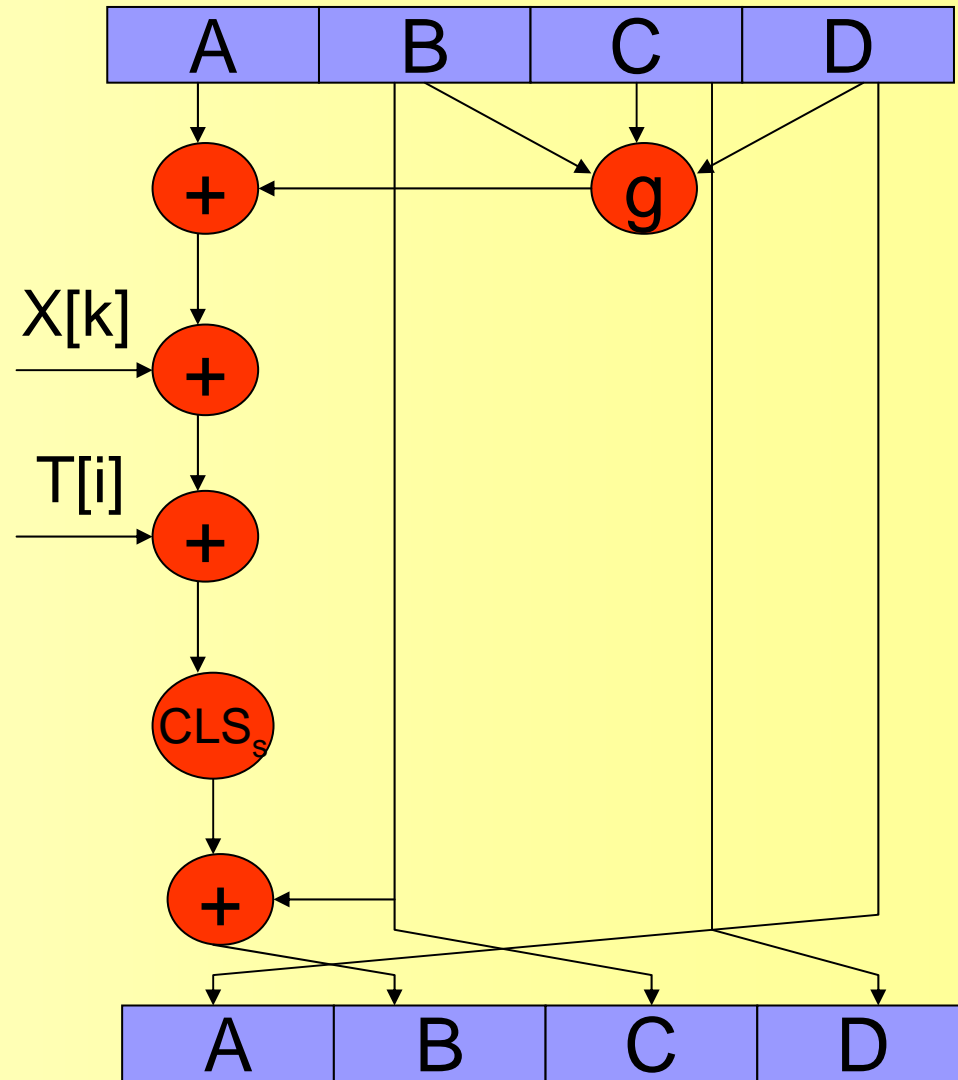


4轮，每轮16步骤



H_{MD5}

1. $X[k]$ 是分组的第 k 个 32 比特,
 $k=1,2,\dots, 16$
2. $T[i]= (2^{32} \text{abs}(\sin(i)))$ 的整数部分
3. $b=b+((a+g(b,c,d)+X[i]+T[i])\lll s)$
循环左移 s 各比特。
4. 函数 g
第1轮 $F(b,c,d)= (b \wedge c) \vee (\neg b \wedge d)$
第2轮 $G(b,c,d)= (b \wedge d) \vee (c \wedge \neg d)$
第3轮 $H(b,c,d)= b \oplus c \oplus d$
第4轮 $I(b,c,d)= b \oplus (b \wedge \neg d)$
5. $\rho_2(k)= (1+5k) \bmod 16$
 $\rho_3(k)= (5+3k) \bmod 16$
 $\rho_4(k)= 7k \bmod 16$





MD5的强度

- Rivest猜想MD5
 - 给定 $H(x)$, 求 y 使得 $H(y)=H(x)$ 需要 2^{128} 数量级的操作
 - 求 x,y , 使得 $H(y)=H(x)$ 的操作, 需要 2^{64} 数量级的操作
- 1996年Dobbertin 提出了针对MD5单轮压缩函数的攻击。
- 2004年8月17日在美国加州圣巴巴拉国际密码学会议(Crypto'2004)上, 山东大学王小云教授等报告了MD5的破解方法。



2.3 SHA-1

- 安全散列算法(SHA)由美国国家标准和技术协会(NIST)提出, 作为联邦信息处理标准(FIPS PUB 180) 在1993年公布。
- 1995年发布了一个修订版 FIPS PUB 180-1通常称为SHA-1
- SHA也是基于MD4的。
- 最大报文长度 $2^{64}-1$, 散列码长度160bit。
- 结构与MD5类似, 抗攻击能力比MD5强;



2.4 RIPEMD-160

- 欧共体的RIPE项目研制的;
- MD4的变种, 为抵抗已知的关于MD4、MD5的攻击而设计的;
- 摘要长度为160, 报文长度不受限制;

3. 密码散列报文鉴别码HMAC



HMAC RFC2104

- **keyed-Hashing for Message Authentication Code**
- 在最近几年，研究的热点转向由密码散列码导出MAC。这样的目的在于：
 - 密码散列函数如MD5和SHA-1的软件执行速度比对称分组密码如DES的快。
 - 很容易获得密码散列函数的库代码。
 - 美国或其他国家对密码散列函数没有出口限制，而对称分组密码，即使用作MAC也是限制的。

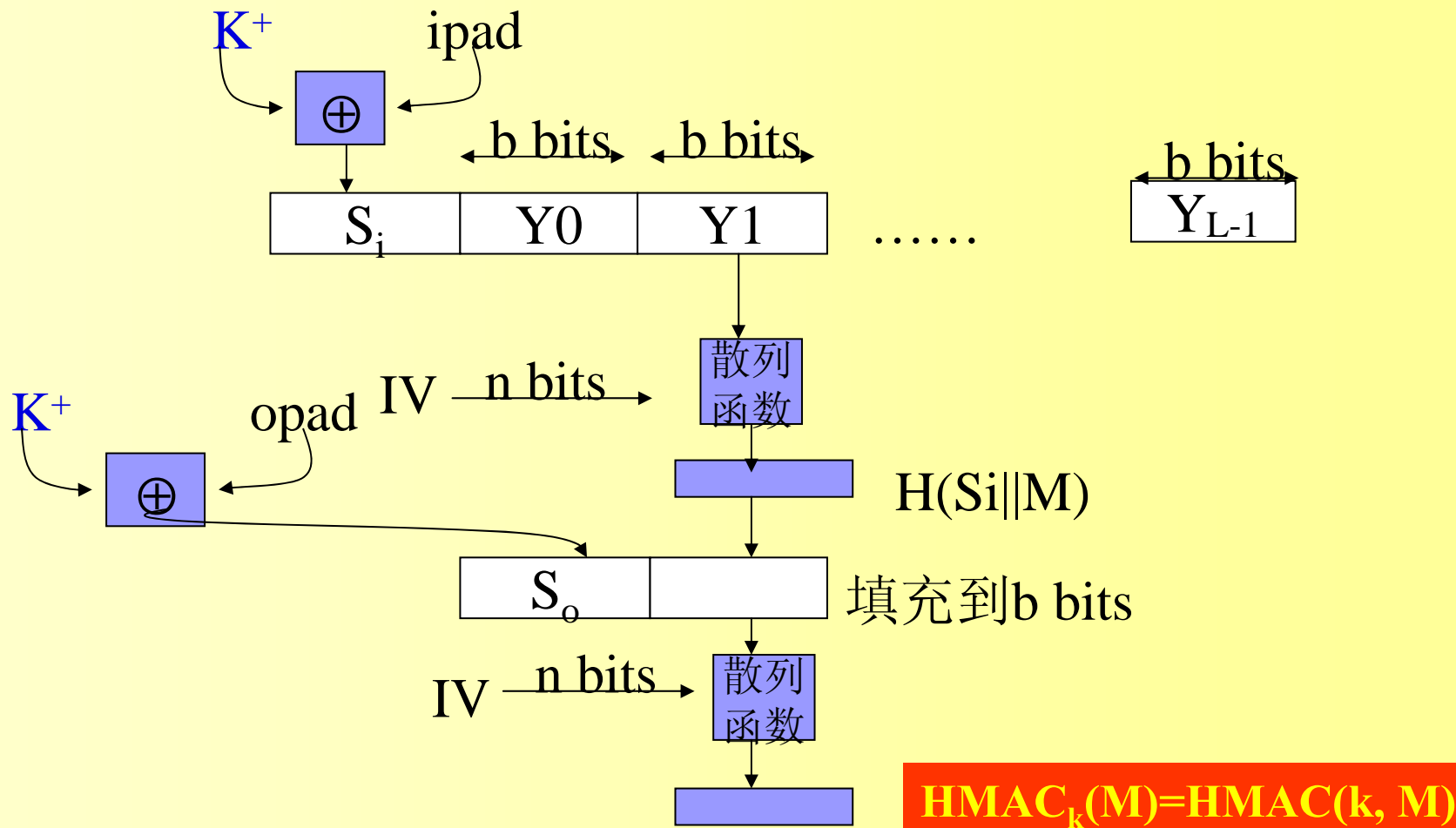


HMAC设计目标

- 无需修改地使用现有的散列函数。特别是，散列函数的软件实现执行很快，且程序代码是公开的和容易获得的。
- 当出现或获得更快的或更安全的散列函数时，对算法中嵌入的散列函数要能轻易地进行替换。
- 保持散列函数的原有性能不会导致算法性能的降低。
- 使用和处理密钥的方式很简单。
- 基于对嵌入散列函数合理的假设，对鉴别机制的强度有一个易懂的密码编码分析。



HMAC 算法 RFC2104





HMAC算法

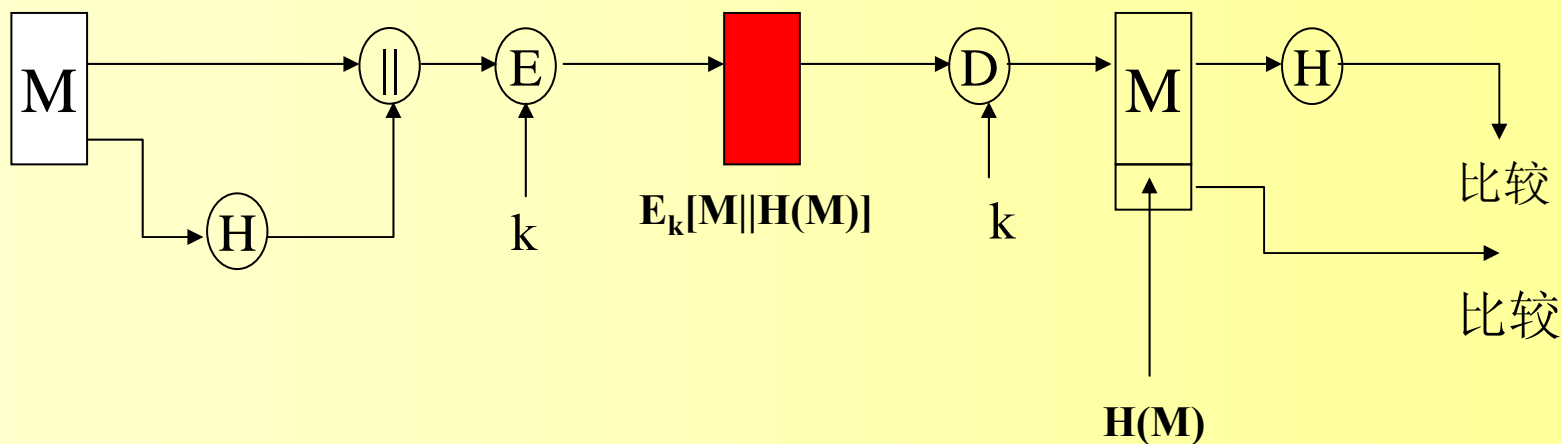
1. 对密钥 K 的左端填充一些0生成一个 b bits的串 K^+ 。
2. ipad: $b/8$ 个00110110, opad: 01011010。
3. 将 K^+ 与中ipad按比特异或(XOR)产生一个 b bits 分组 S_i 。
4. 将报文 M 附加到 S_i 后。
5. 使用 H 计算第3步产生流的散列值。
6. 将 K^+ 与opad按比特异或产生一个 b bits 的分组 S_0 。
7. 将第4步产生的散列值附加到 S_0 后。
8. 使用 H 计算第6步产生流的散列值, 并输出这个结果。

4. 散列函数与数字签名的应用



散列函数的使用 (a)

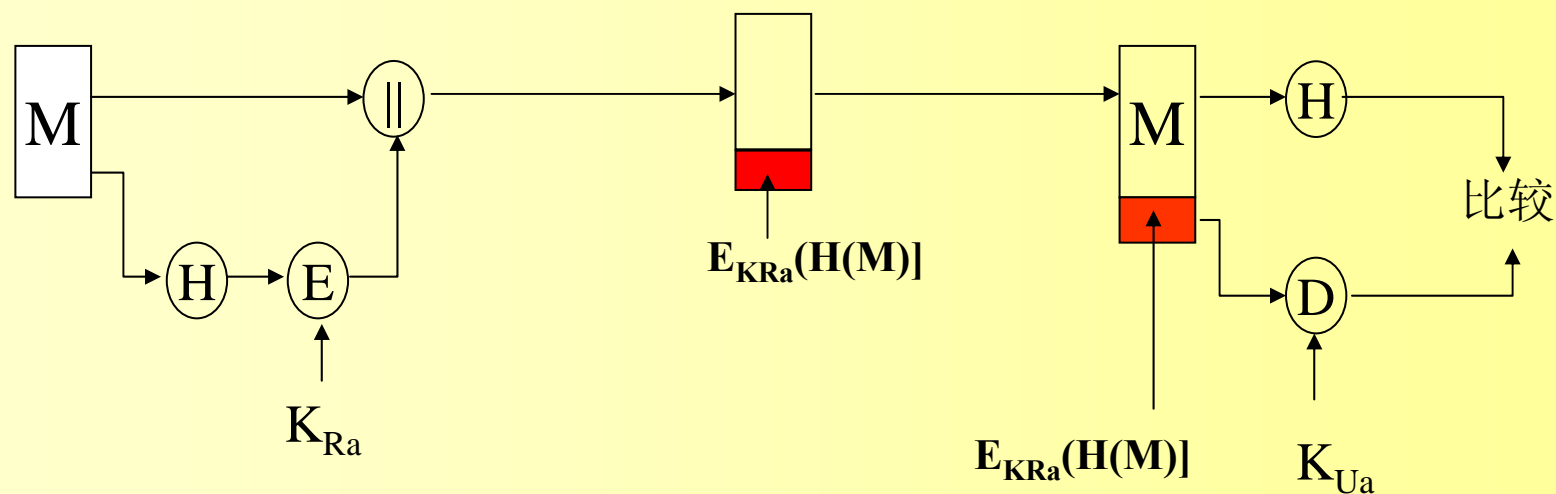
a) 提供了保密和鉴别。





散列函数的使用 (c)

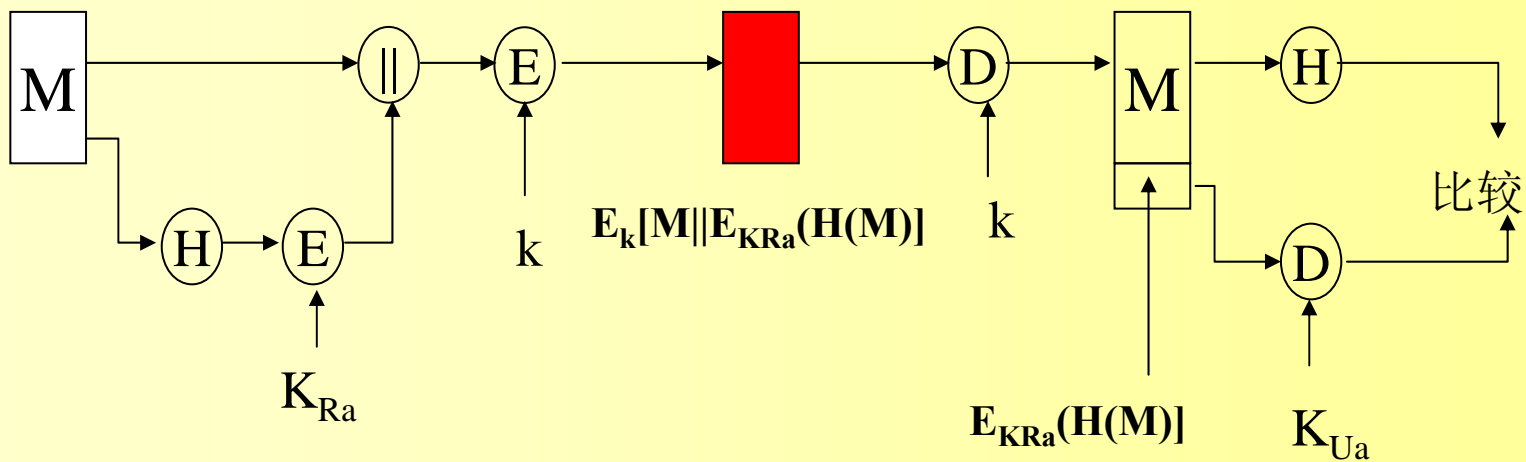
- 提供鉴别、防抵赖。





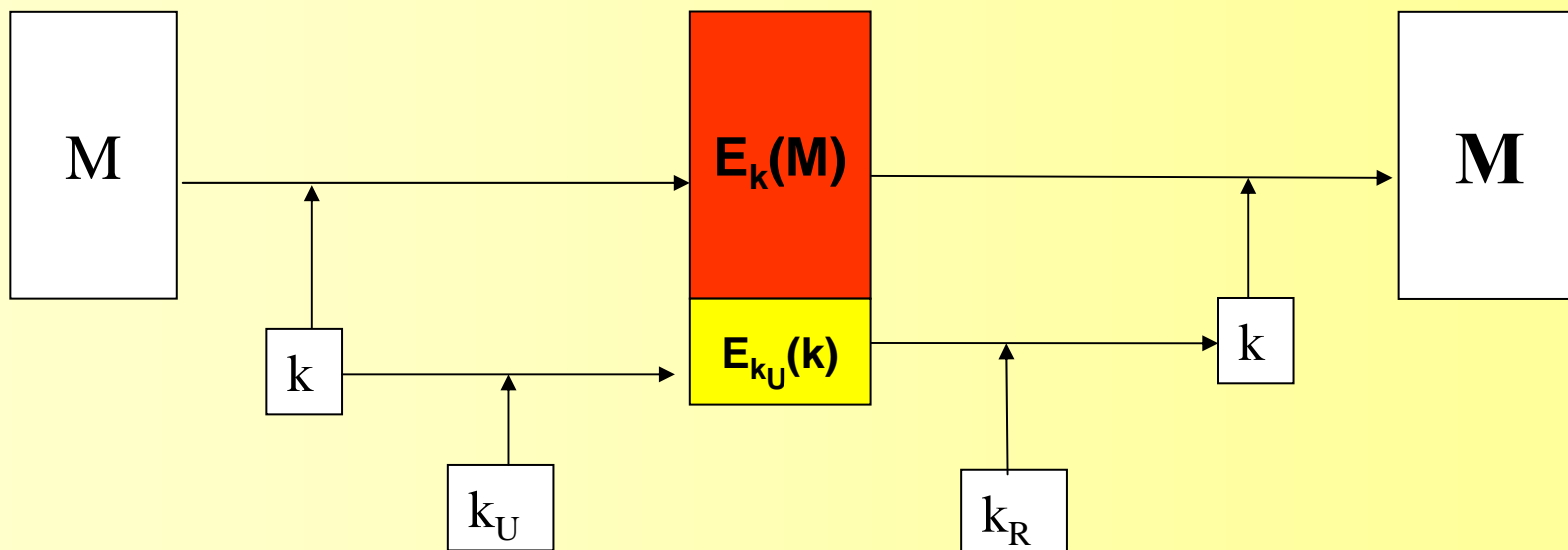
散列函数的使用 (d)

- 同时提供保密性、鉴别和抵赖。





数字信封





散列函数的使用 (b)

- 散列与加密结果合并为一个整体函数实际上就是一个MAC。 $E[H(M)]$ 是变长报文 M 和密钥 K 的函数值，且它生成一个定长的输出，对不知道该密钥的对手来说是安全的。
- 提供鉴别

