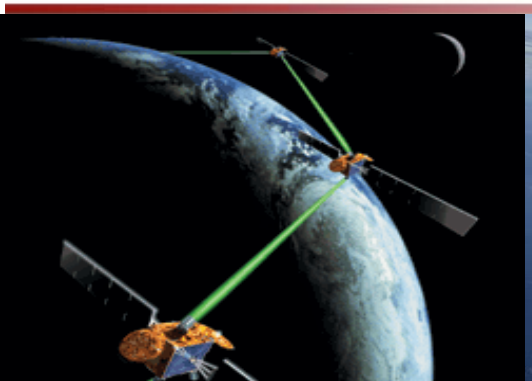
普通高等教育“十一五”国家级规划教材
教育部2011年精品教材

网络安全—技术与实践（第2版）

刘建伟 王育民 编著

清华大学出版社




课件制作人声明

- 本课件总共有17个文件，版权属于刘建伟所有，仅供选用此教材的教师和学生参考。
- 本课件严禁其他人员自行出版销售，或未经作者允许用作其他社会上的培训课程。
- 对于课件中出现的缺点和错误，欢迎读者提出宝贵意见，以便及时修订。

课件制作人：刘建伟

2016年10月20日

消息认证与杂凑函数

一 什么是认证

二 信息加密能保证真实性吗

三 杂凑（Hash）函数

四 消息认证码

五 应用杂凑函数的基本方式

攻击者类型

- 信息产生、处理、传输、存储的各个环节都有安全问题隐患。
- 被动攻击（passive attacks）
  - 信道窃听
  - 盗取存储数据
- 主动攻击（active attacks）
  - 假冒消息
  - 窜改消息
  - 截留消息
  - 修改数据
  - ……

一、什么是认证 (authentication)

保密性 (confidentiality)：采用加密机制

- 不希望他人知道消息的内容
  - 数据加密

真实性（authenticity)：采用认证机制

- 消息内容的完整性
- 消息内容的真实性
- 消息来源的真实性
- 实体身份的真实性
  - 认证机制

认证方式的分类

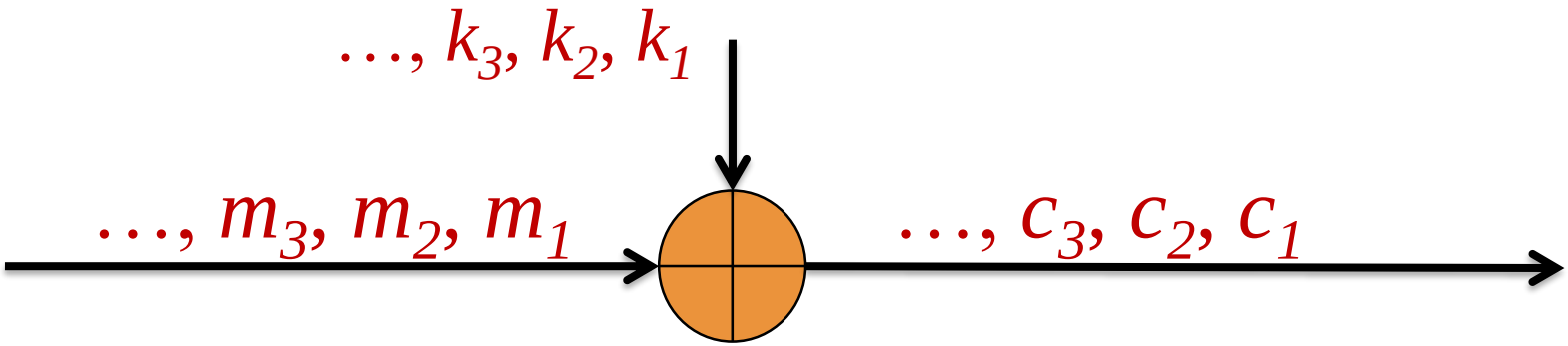
- 通信双方相互信任的认证（如企业内部人员之间）
  - 对称认证（symmetric authentication）
  - 针对第三方的攻击，例如查验文件是否被人修改过
- 通信双方相互不信任的认证（如商业伙伴之间）
  - 非对称认证（asymmetric authentication）
  - 针对来自对方的攻击，例如查验收到的文件是否真实

二、信息加密即保证真实吗？

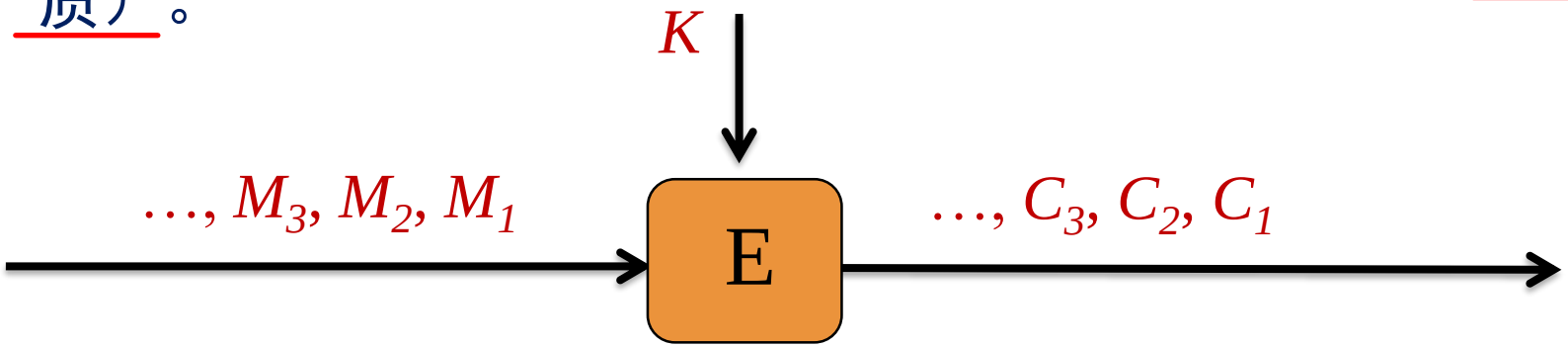
- 对称认证：确保消息的真实性
  - 保证信息来源可信（data origin authentication）
  - 保证信息的完整性（the integrity of the information）
- 直到1970年代末，人们仍然错误地相信：
  - 如果解密后得到符合语义的消息，即可断定消息来源的真实。
  - 因为这意味密文经过真实密钥加密而来，而掌握密钥的人是可靠的。
- 事实上，保证真实性不仅要求加密算法安全，还取决使用正确的加密模式 ！！

反例一：对称加密不能保证真实性

- 例1. 流密码：主动攻击者可以通过置乱相应密文比特，来达到改变任意明文比特的目的。



- 例2. ECB模式分组密码：主动攻击者可以记录一些密文分组信息，用来替代其它分组。如果分组之间不相关，完全不可能检测到攻击；除非明文之间有关联（利用冗余性质）。



反例二：公钥加密不能保证真实性

- 使用公钥加密可以提供保密性，但不能提供认证。
- 发方Alice采用收方Bob的公钥K_p^B对消息m进行加密，因只有Bob知道自己的私钥K_s^B，故只有Bob才能对收到的消息准确解密。
- 但是，任何人可以假冒Alice，用Bob的公钥K_p^B对消息m加密。因此，这种方法不能确保所收消息的真实性。



保证完整性的方式-1

- 类似于对称密码，数据的真实性依赖于一个短密钥的保密性和真实性。

  首先计算：MAC=hash(m||k)

  - 信息的真实性依赖密钥的保密性与真实性
  - $\{m \| MAC\}$

- 消息认证码MAC（Message Authentication Code）

  - 在计算Hash值过程中，有密钥的参与。

保证完整性的方式-2

- 篡改检测码MDC（Manipulation Detection Code）

  首先计算：MDC=hash(m)

  - 在计算hash的过程中，没有密钥的参与。
- 信息的真实性基于 MDC的真实性
  - 例如，针对所有重要文件计算MDC，文件将发往异地的朋友，其中MDC通过电话传输。电话信道的真实通过语音识别保障。
- 增加冗余并不充分保证提高抗攻击的等级
  - 可能发起重放攻击
  - 双方共享密钥，完全对等
  - 仅提供不可否认(non-repudiation)，但实际操作也很困难

  注意：上述方法的前提是发送者(sender)和接收者(receiver)要相互信任，一旦出现纠纷将无法判决。

三、杂凑（Hash）函数

- 信息的真实性通过验证秘密的保护，以及一个短烙印（*imprint*）或Hash值的真实性来确认。
- Hash函数来源于计算机技术：将任意串压缩成定长的比特串。
- 密码学意义Hash函数：cryptographic hash functions
- Hash函数的一些常见称谓：
  - 中文：杂凑函数、散列函数、哈希函数
  - 英文：hash code, hash total, hash result, imprint, checksum, compression, compressed encoding, authenticator, fingerprint, Message Integrity Code, message digest.

1. 单向杂凑(Hash)函数

- 杂凑(Hash)函数是将任意长的数字串*m*映射成一个较短的定长输出数字串的函数，我们关心的通常是单向杂凑函数；
- 分类：强单向杂凑与弱单向杂凑 (无碰撞性collision-free)；
- 单向杂凑函数的设计理论；
- 杂凑函数除了可用于数字签名方案之外，还可用于其它方面，诸如消息的完整性检测、消息的源点认证检测等。

2. 什么是杂凑函数

- 函数$y=H(x)$，要求将任意长度的x变换成固定长度的y，并满足：

  1. 单向性：任给y，由$y=H(x)$计算x困难

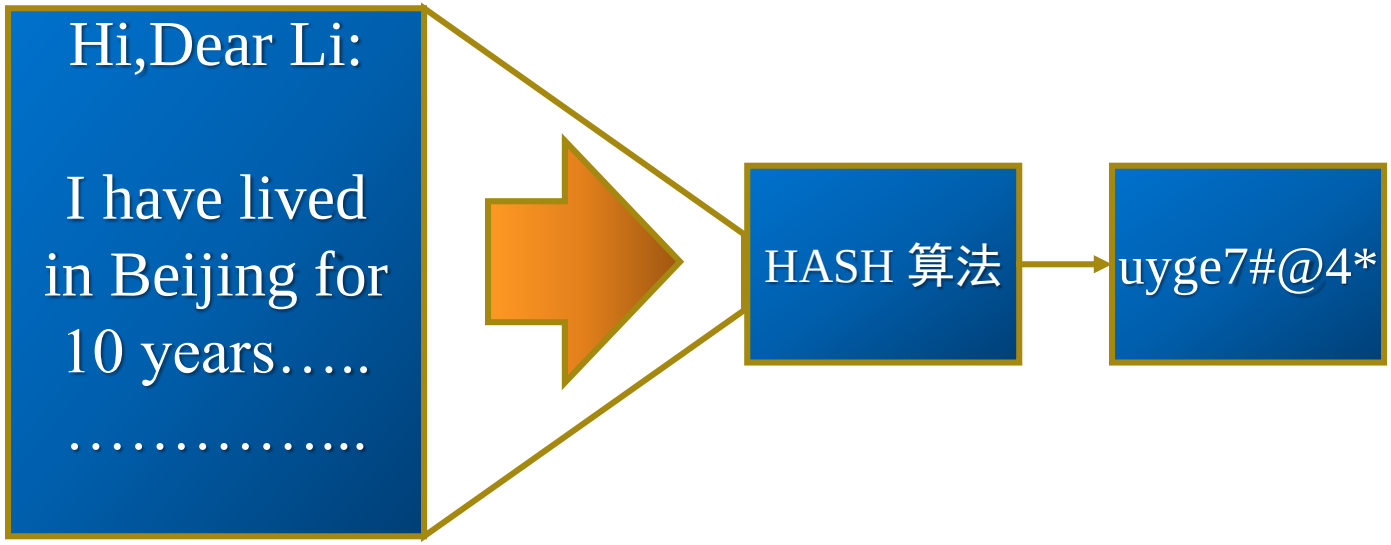
  2. 快速性：任给x，计算$y=H(x)$容易

  3. 无碰撞：寻找$x_1 \neq x_2$，满足$H(x_1)=H(x_2)$是困难的。

- 常用的hash 函数有 MD5, SHA，以及采用分组密码算法构造的hash函数（也叫做杂凑函数）等。

md50;

单向杂凑函数： One-way Hash Function



- 算法特点：不定长度输入，固定长度输出（MD5输出为16字节、SHA-1输出长度为20字节）
- 雪崩效应：若输入发生很小的变动，则可引起输出较大变动。
- 完全单向：已知输出无法推算出输入，已知两个输出的差别无法推算出输入的差别。

3. 杂凑函数的安全性需求

① 输入长度可变：函数的输入可以是任意长度；

② 输出长度固定：函数的输出是固定长度；

③ 效率高：已知 x，求$H(x)$较为容易，可用硬件或软件实现；

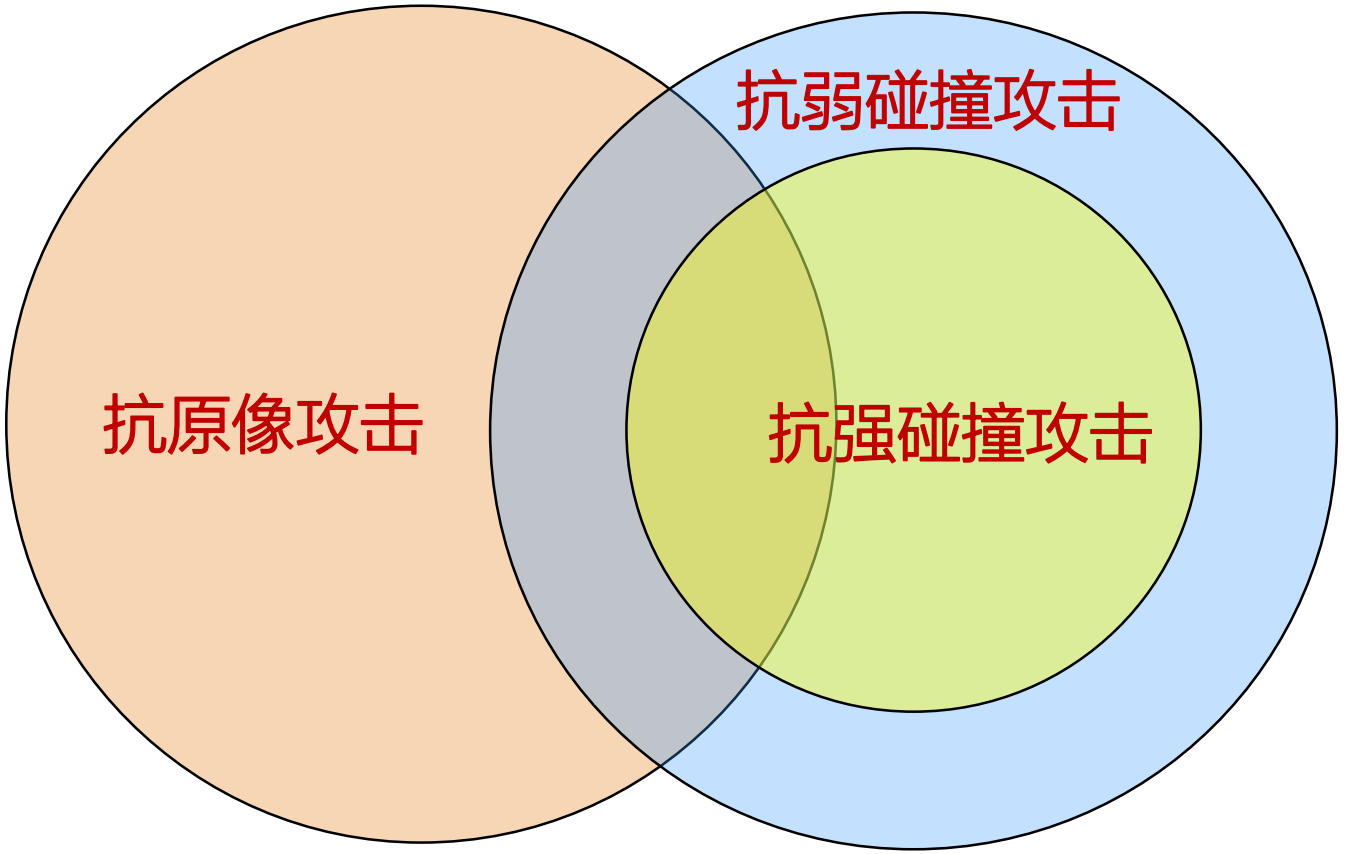
④ 抗原像攻击（单向性）：已知 h，要找到满足$H(x)=h$的 x 在计算上是不可行的；

⑤ 抗第二原像攻击（抗弱碰撞性）：对任意给定的x，要找出满足$y(y\neq x)$且$H(y)=H(x)$的y在计算上是不可行的。

⑥ 抗碰撞攻击（抗强碰撞性）：找出任意两个不同的输入x和y，使得$H(y)=H(x)$在计算上是不可行的。

⑦ 第⑤⑥给出了杂凑函数无碰撞的概念。

三个安全性之间的联系

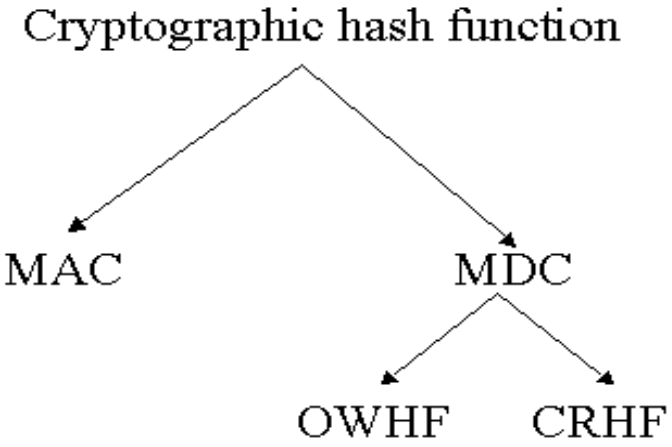


4、Hash码的分类

- 消息认证码：MAC的计算中有密钥的参与
- 消息检测码：MDC的计算中无密钥的参与
  - 单向Hash函数(one-way hash function，OWHF)
  - 抗碰撞Hash函数(collision resistant hash function, CRHF)。

    注：CRHF是更好的强单向Hash函数

- 无碰撞性Hash函数(collision free hash function)
  - 无碰撞的hash函数是存在的
  - 但在实际中很难找到它们。



5、单向Hash函数（OWHF）

定义1：单向杂凑函数满足以下条件：

1. 变量x可以是任意长度，而$h(x)$的结果具有固定的长度n bits (如64b，80b等)
2. 抗原象攻击：已知一个杂凑值h，找一个输入串x，使得$h=h(x)$，这在计算上是不可行的。
3. 抗碰撞攻击：找两个输入x和$y(x\neq y)$，使得$h(x)=h(y)$，这在计算上是不可行的。

6、抗碰撞Hash函数（CRHF）

定义2：抗碰撞的杂凑函数满足以下条件：

1. 变量x可以是任意长度，而$h(x)$值具有固定的n bits (如$128b...160b$)长度。
2. 杂凑函数必须是单向的。
3. 杂凑函数抗碰撞：意味着很“难”找到两个不同的消息x和y，杂凑后得到相同的值$h(x)=h(y)$。

7、杂凑函数在密码学中的应用

- 在数字签名中，杂凑函数一般用来产生消息摘要MDC。
  - 首先，将要签署的消息进行杂凑运算；
  - 此后，对MDC进行数字签名；
  - 验证签名时，只需验证MDC的正确性，就可以判断消息的真伪。
- 在公钥密码系统中，杂凑函数被用于数据的完整性验证。
- 在需要随机数的密码学应用中，杂凑函数被广泛地用作实用的伪随机函数。这些应用包括：密钥协商，认证协议，电子商务协议，知识证明协议。

四、消息认证码MAC

- MAC也称为密码校验和，它由下述函数产生：
  - $MAC=h(m||k)$
  - 其中，m是一个变长的消息，k是收发双方共享的密钥，$h(m||k)$是定长的认证符。
- 在实际中，发送者将消息m和此认证符一起发给接收者：
  - $(m, h(m||k))$
- 接收者在收到消息后，计算：$MAC'=h(m||k)$
  - 比较$MAC'=MAC$？
  - 若相等，则说明消息未被篡改；
  - 若不等，则说明消息被改动了。

五. Hash函数应用之一：构造MAC

- 无保密功能，只能用来认证；
- **MAC：**为了保护消息的真实性，我们可以首先计算$MAC=h(k||m)$，然后将其附着在信息的后面，即$\{m,h(k||m)\}$。
- 消息的真实性依赖于共享密钥的真实性和保密性。
- 任何拥有这一密钥的人均可以验证该消息的真实性。
- 对消息真实性的保护已经转化成安全地在通信双方之间建立共享密钥的问题。

Hash函数应用之二：构造MDC

- **MDC：**如果采用MDC，消息的真实性就转化成检验一个固定长度的比特串的真实性（优点：不需共享密钥），即 $MDC=h(m)$
- 需要认证信道传输MDC：发送者需要将MDC通过一特定的信道传送给对方。
- 例如，发送者可以打电话告诉对方MDC值。而这个消息的真实性依赖于电话中对于发送者声音的确认。
- $h(x)$应为抗碰撞的杂凑函数（CRHF）
  - 发送者发送：m
  - 接收者验证：$h(m')=h(m)$?

MDC数学表示

迭代Hash函数

- 信息 X 被分为 t 个分组：

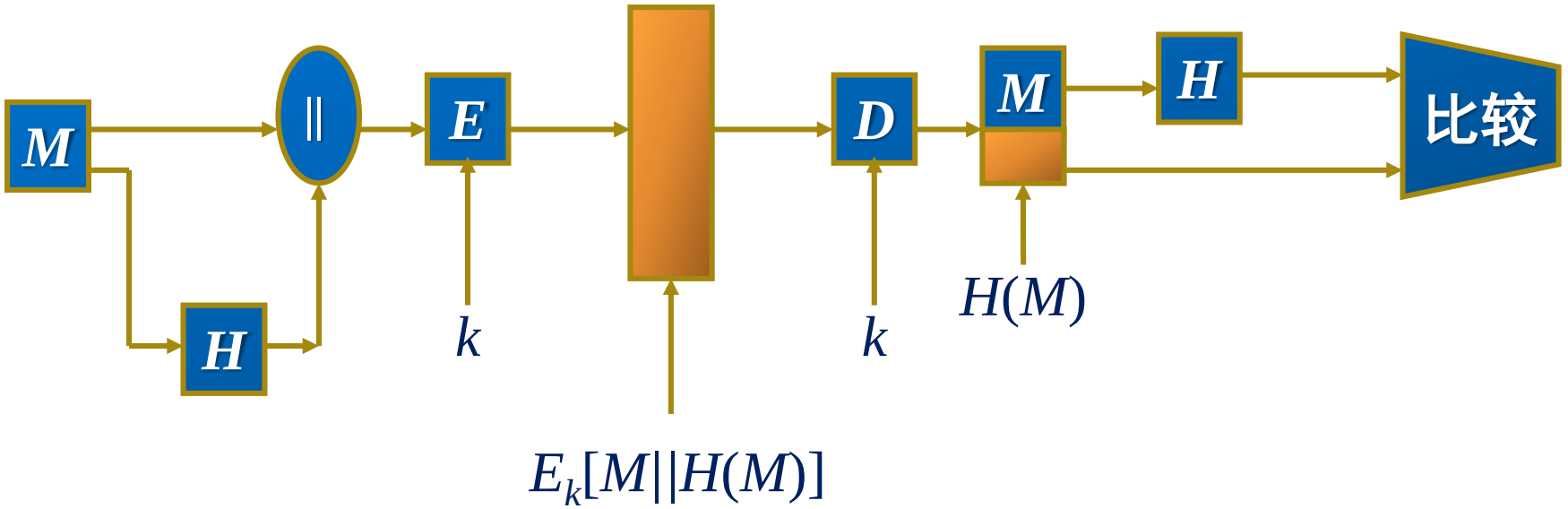
  $X_1, X_2, \ldots, X_t$（如必要，需填充补位padding）

  计算：$H_0=IV$, $H_i=f(X_i, H_{i-1})$, …，$h(X)=g(H_t)$

  - h为Hash函数，f为轮函数，g为输出变换
  - IV（Hash函数的部分）和padding（填充）将对安全有重要影响。

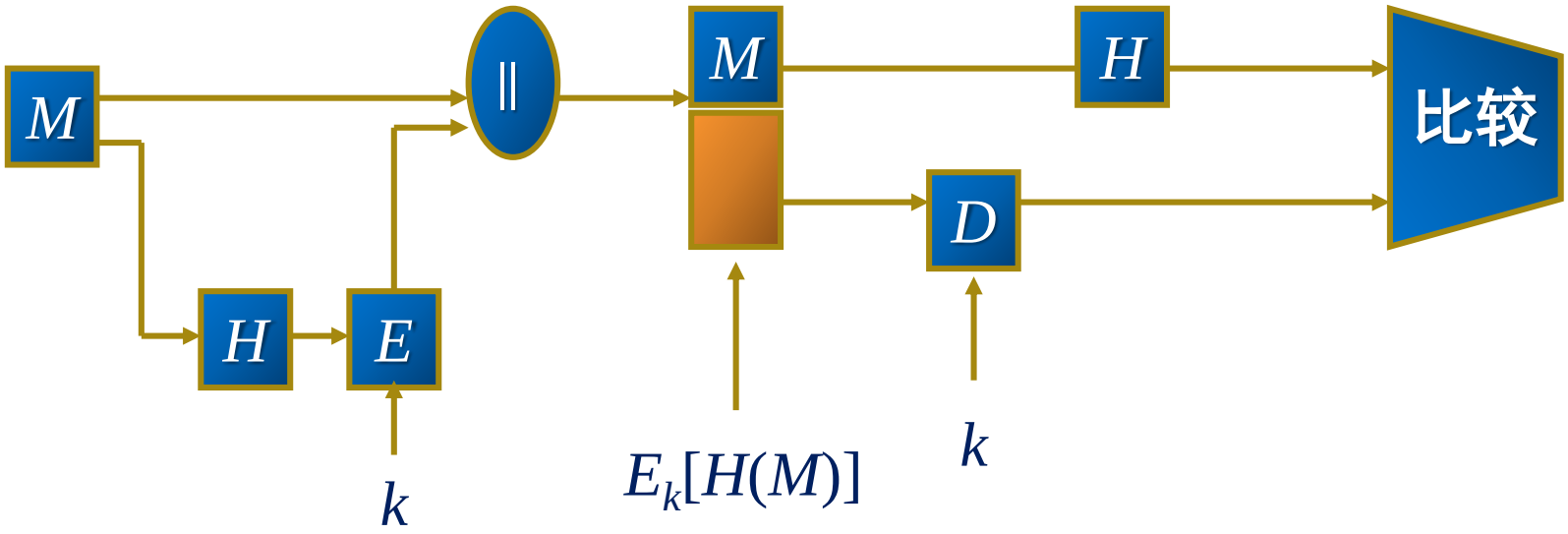
六. 应用杂凑函数的基本方式

1. 既提供保密性，又提供消息认证



六. 应用杂凑函数的基本方式

2. 仅仅提供消息认证



3. MD系列杂凑函数介绍

- Ron Rivest设计的系列杂凑函数系列:
  - MD4[Rivest 1990, 1992, 1995; RFC1320]
  - MD5是MD4的改进型[RFC1321]
  - MD2[RFC1319]，已被Rogier等于1995年攻破
- 较早被标准化组织IETF接纳，并已获得广泛应用
- 安全性介绍

3. MD系列杂凑函数介绍（续）

- Ron Rivest设计的系列杂凑函数系列:
  - MD2[RFC1319]，已被Rogier等于1995年攻破
  - MD4[RFC1320]，已被攻破；
  - MD5是MD4的改进型[RFC1321]，被山东大学王小云攻破。
- Hash值长度为128bits。
- MD5被标准化组织IETF接纳并获得广泛应用；
- 但是，目前MD5已经过时，不能再使用。


15.10.2006

SHA/SHA-1/SHA256/SHA512

- NIST和NSA为配合DSS，设计了安全杂凑标准(SHS)，其算法为SHA[FIPS PUB 180]，修改的版本被称为SHA-1[FIPS PUB 180-1]。
- SHA/SHA-1采用了与MD4相似的设计准则，其结构也类似于MD4，但其输出为160bit。
- 王小云声称，SHA-1已经破了一半，因此世界上正在转向采用SHA-256和SHA-512。

SHA与MD4和MD5的比较

	MD4	SHA	MD5
Hash值	128bit	160bit	128bit
分组处理长	512bit	512bit	512bit
基本字长	32bit	32bit	32bit
步数	48(3*16)	80(4*20)	64(4*16)
消息长	≤2^64bit	2^64bit	不限
基本逻辑函数	3	3(第2,4轮相同)	4
常数个数	3	4	64
速度		约为MD4的3/4	约为MD4的1/7

谢谢！