现代密码学

中国海洋大学　信息安全实验室

第4章

Hash函数与消息认证码

本章内容

4.1 Hash函数

4.2 消息认证码 (MAC)

4.3 消息认证

4.1 Hash函数

Hash函数是很多密码系统中的重要模块

在密码学中有着举足轻重的地位

定义

•又称 哈希函数、杂凑函数、单向散列函数

•是一个将任意长度的消息映射成固定长度输出的函数

$$H:\{0,1\}^{*}\rightarrow\{0,1\}^{n}$$

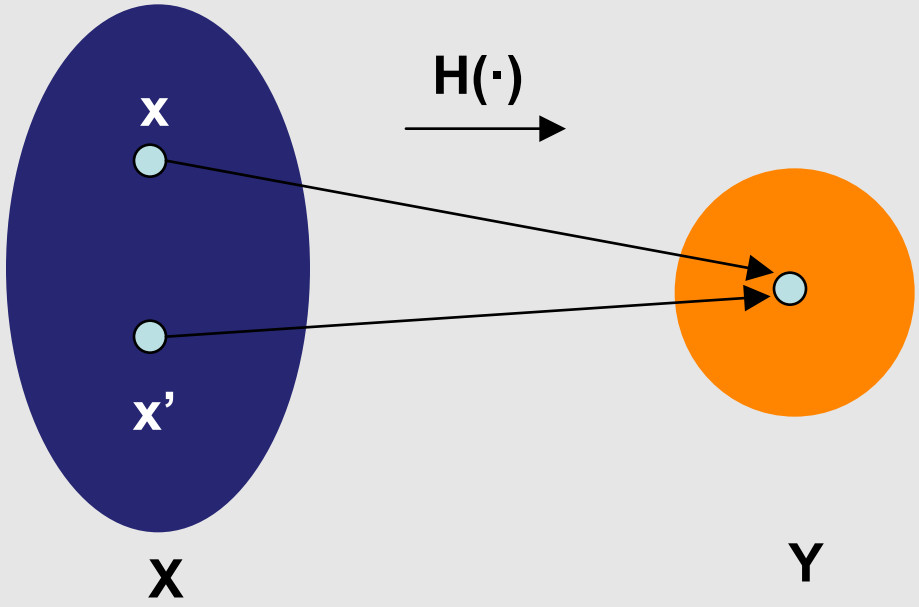
- 输入称为“消息”
- 输出称为“散列值” (Hash值、消息摘要)
- n是散列值的长度

通常，Hash函数是一个具有压缩功能的函数

- 设X是消息的集合，Y是散列值的集合
- 我们总是假设|X|≥|Y|，并且经常假设更强的条件|X|≥2|Y|

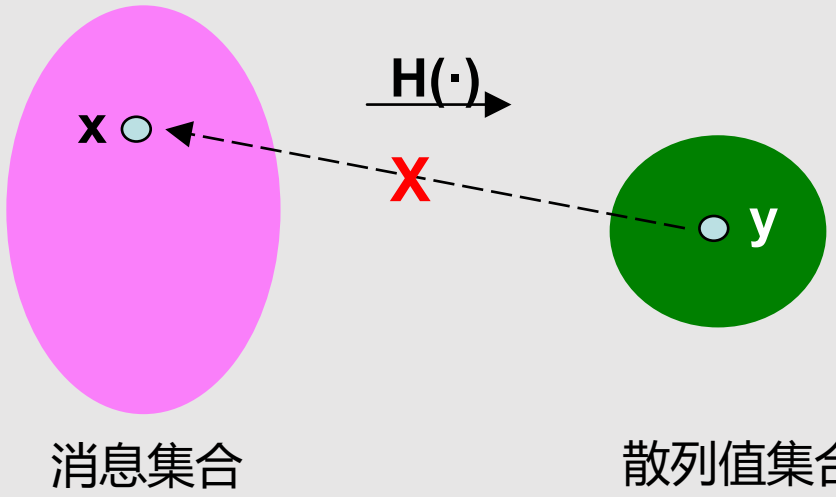
根据鸽巢原理：两个甚至多个消息会映射为同一个散列值

(称 x和x’ 是一对碰撞)



Hash函数 安全性

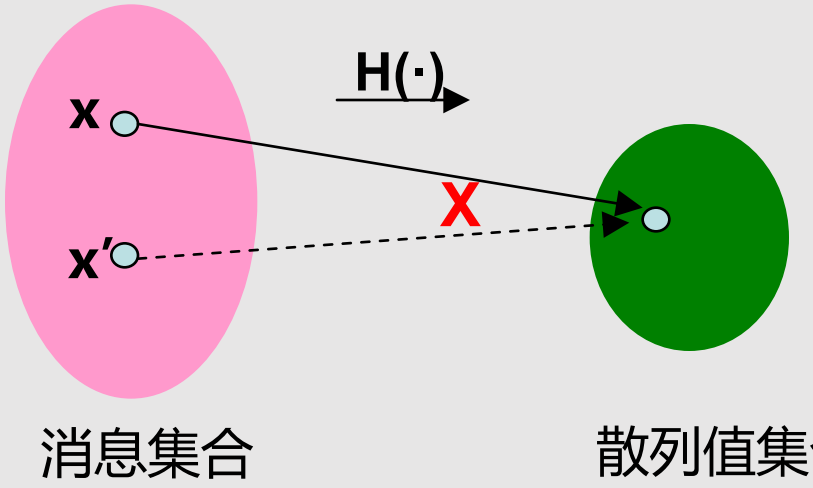
- **原像稳固**
  - 给定散列值y，要找到一个x，使得H(x)=y是计算上不可行的



满足该性质的Hash函数称为“**单向的**”或“**原像稳固的**”

Hash函数 安全性

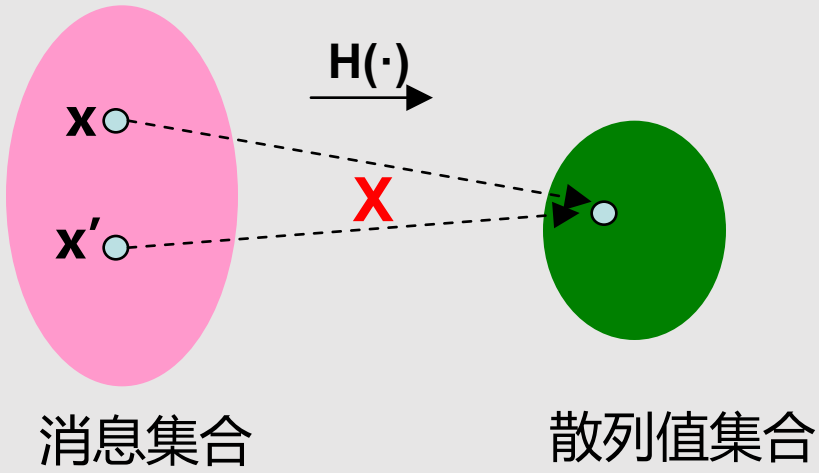
- **第二原像稳固**
  - 给定消息x，找到另一个x’，使得H(x’)=H(x)是计算上不可行的



满足该性质的Hash函数称为“**第二原像稳固的**”

Hash函数 安全性

- **碰撞稳固**
  - 找到两个不同的消息 x和x’，使得H(x)=H(x’)是计算上不可行的。



满足该性质的Hash函数称为“**碰撞稳固的**”

Hash函数 安全性

对Hash函数的攻击实际就是 寻找一对碰撞 的过程

如果Hash函数 H 设计得“好”，对给定x，要想求得相应的散列值，必须通过计算H(x)才行。

如果想绕过 H，采用其他方法求出散列值是计算上不可行的。

即使在已知若干散列值$H(x_1),H(x_2),\ldots$的情况下，仍是如此。

因此，我们有

$H(x_1)+H(x_2)\neq H(x_1+x_2)$

$H(x_1)H(x_2)\neq H(x_1x_2)$

…

(这些特点在后面介绍数字签名时非常有用)

生日悖论 (birthday paradox)

生日问题：

假设每个人的生日是等概率的，每年有365天，问k最小应是多少，才能使k人中至少有2人生日相同的概率大于½ ？

Q: k人生日都不同的概率是多少？

$$\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{k-1}{365}\right)$$

k人中至少有2人生日相同的概率为：

$$p=1-\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{k-1}{365}\right)\approx 1-\mathrm{e}^{-\frac{k(k-1)}{2\times 365}}$$

$p>\frac{1}{2}$时，$k\approx 1.1774\sqrt{365}\approx 23$

只需23人，至少就有两个人生日相同的概率大于½

这个数比人们直觉小得多，因而称为生日悖论

对Hash函数的生日攻击

利用生日悖论原理攻击Hash函数：

目的：设散列值是n比特，构造两个不同的消息m和M，使得m和M具有相同的散列值。

m: 明天8点电影院见，约吗？

M: 请转账100万到我的银行账户

攻击步骤（1）：消息变形

分别改变m和M的内容，但保证语义不变。

- 例如增加空格、使用缩写、使用意义相同的单词、去掉不必要的单词等。
- 分别变成r和R个新消息。

m: 明天8点电影院见，约吗？

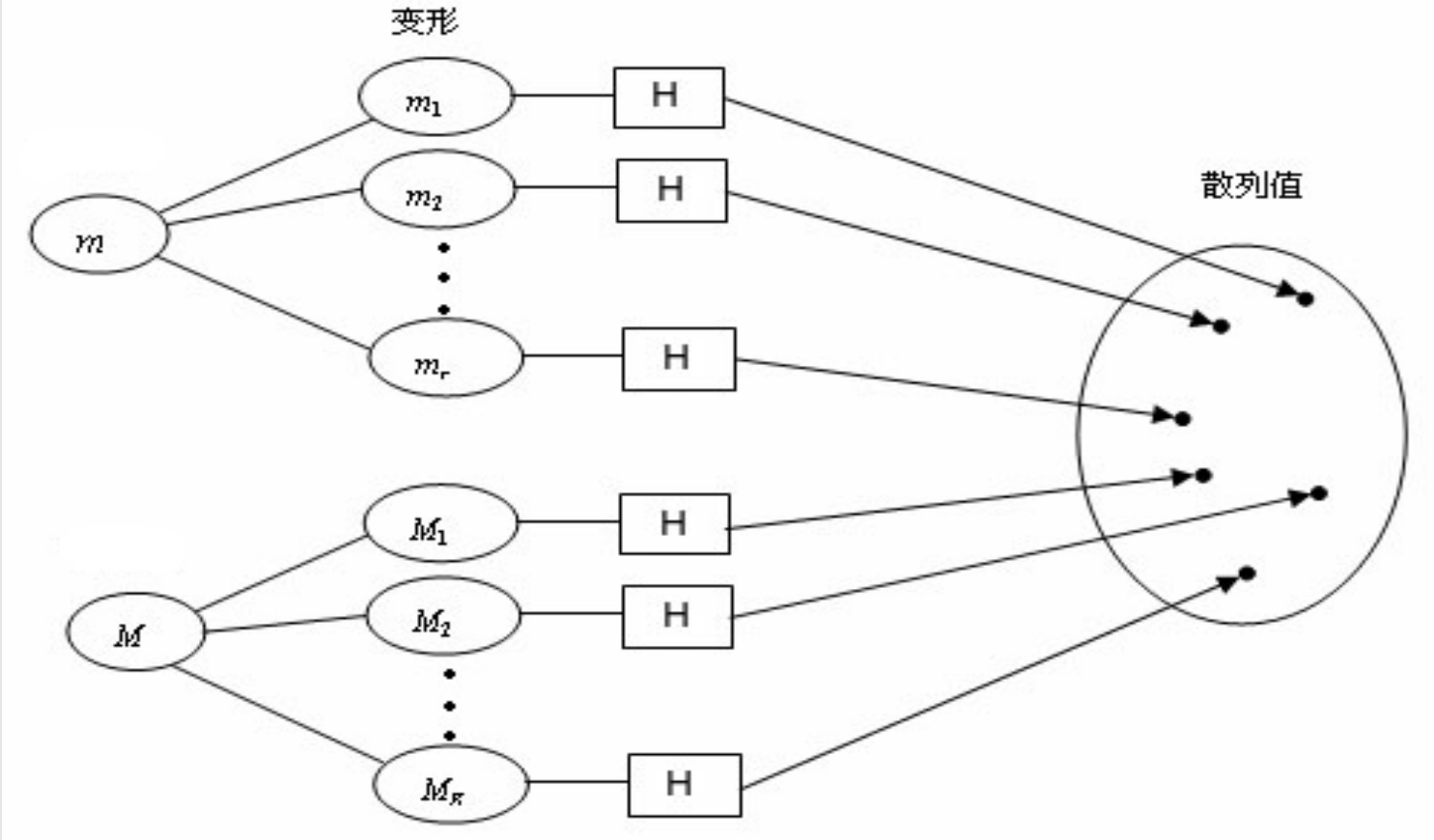
m_1：明天8点电影院见，好吗？
m_2：明天8点电影院见，OK？
m_3：明天八点电影院见吧？
…

M: 请转账100万到我的银行账户

M_1：请转账100万到我的账户
M_2：请转账10,000,000到我的账户
M_3：请转账10,000,000到我的银行账户
….

攻击步骤（2）：计算散列值

分别计算各变形消息的散列值。



攻击步骤（3）：计算碰撞概率

计算m的变形与M的变形发生碰撞的概率。

- n比特长的散列值共有2^n个，对于给定m的一个变形m_i和M的一个变形M_j，m_i与M_j不碰撞的概率是$1-1/2^n$。

M有R个变形，所以M的全部变形都不与m_i碰撞的概率：$\left(1-1/2^n\right)^R$

m有r个变形，因此m的变形与M的变形都不碰撞的概率：$\left(1-1/2^n\right)^{rR}$

m的变形与M的变形发生碰撞的概率：$P(n)=1-\left(1-\frac{1}{2^n}\right)^{rR}\approx 1-e^{-\frac{rR}{2^n}}$

安全性分析

当$r=R=2^{n/2}$时，$P(n)=1-e^{-1}=0.63$。

所以，当散列值长度n=64时，只需分别构造$r=2^{32}$个m的变形和$R=2^{32}$个M的变形，便可以0.63的概率找到一对碰撞，完全可行。

对于散列值长度为64 bit 的Hash函数，
生日攻击的时间复杂度约为$O(2^{32})$，所以是不安全的。

生日攻击告诉我们：为了能达到n-bit的安全性，你所选择的Hash函数的散列值长度应该是2n。

- 例如：如果你想让攻击者成功找到碰撞的可能性低于$1/2^{80}$，那么应该使用散列值长度是160-bit的Hash函数。

Hash函数 迭代构造法

- 大多数实用的Hash函数都是采用迭代技术构造的
- 这种Hash函数称为“迭代Hash函数”

- 主要模块和技术

① 压缩函数

$$f:\{0,1\}^{m+t}\rightarrow\{0,1\}^{m}$$

② 迭代技术

- 设消息x是一个长度为t的比特串。重复应用压缩函数f，对x进行多次压缩，最后得到x的散列值。

算法的核心技术

设计“好”的压缩函数f

攻击者对算法的攻击重点是f的内部结构

和分组密码一样，迭代Hash函数由若干轮组成，所以对f的攻击需分析各轮之间的关系

突破口往往是先找到f的碰撞，再找整个Hash函数的碰撞

由于f是压缩函数，存在碰撞是不可避免的（鸽巢原理）

因此在设计f时就应保证 **找到碰撞在计算上是不可行的**

迭代法构造Hash函数的步骤

1. 预处理
2. 迭代处理
3. 输出变换（可选）

“输出变换”阶段是可选的，若无此阶段，迭代处理的输出便是最终的散列值

① 预处理

用一个填充函数pad(·)在消息x右方追加若干比特，得到比特串y，使得y的长度为t的倍数。即有

$$y = x \| pad(x) = y_1 \| y_2 \| \dots \| y_r, \text{其中} |y_i| = t$$

典型的填充函数是先追加与x等长的值，再追加若干比特（如0）。

该阶段必须保证变换是单射

- 因为如果预处理变换不是单射，则存在$x \neq x'$使得$y=y'$，从而$H(x)=H(x')$，能够很容易找到碰撞。

② **迭代处理**

设H_0=IV是一个长度为m的初始比特串，重复使用压缩函数f，依次计算

$$H_i = f(H_{i-1} \| y_i)$$

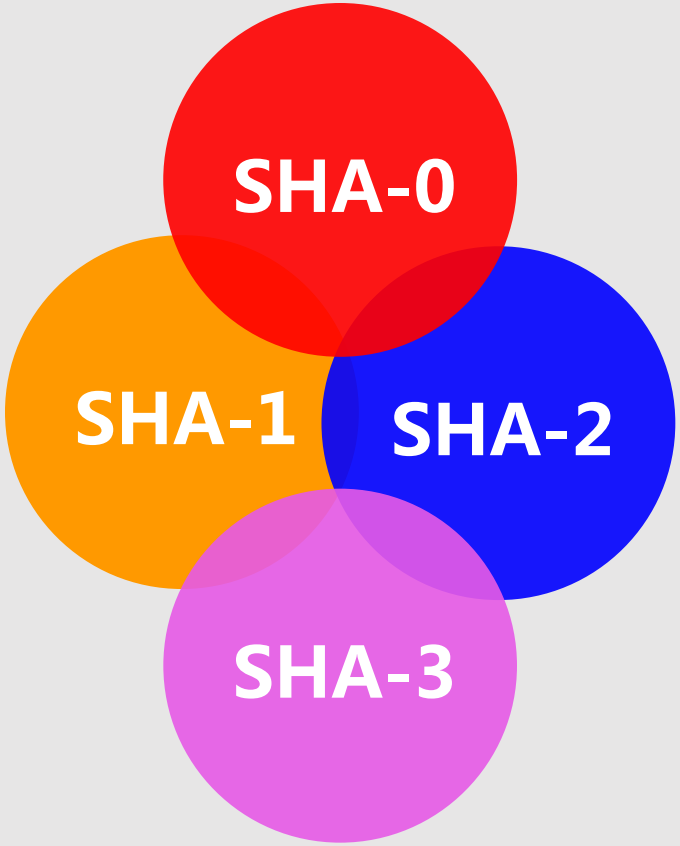
直到计算出H_r为止。

③ **输出变换（可选）**

设函数 $g:\{0,1\}^m \to \{0,1\}^n$, 令

$$H(x) = g(H_r)$$

几个著名的Hash函数



SHA 系列

MD5

时间	1992
发明者	R.Rivest
相关文档	RFC 1321

Ronald L. Rivest

输入	将消息分成512bit一组处理
输出	散列值长度 128 bit
应用领域	商业

MD5

直接构造	不依赖任何密码模块和假设条件
速度	算法简洁，计算速度快
平台	特别适合32位计算机软件实现

由于性能优势，MD5一度成为金融业、数据库系统等首选的Hash函数

MD5加密 的说法是不对的

MD5

百度为您找到相关结果约3,850,000个

MD5在线加密 - MD5加密 - MD5加密工具 - MD5在线转换

关于MD5加密MD5的典型应用是对一段信息(Message)产生信息摘要(Message-Digest),以防止被篡改。比如,在UNIX下有很多软件在下载的时候都有一个文件名相同,文件扩展名...

md5jiami.51240.com/ - 百度快照 - 65%好评

MD5加密 - 站长工具

本工具可以提供32位,16位等MD5加密。... 文字加密解密 MD5加密 URL加密 JS加/解密 JS混淆加密压缩 ESCAPE加/解密 BASE64 散列/哈希 迅雷,快车,旋风URL加解密 ...

tool.chinaz.com/Tools/... - 百度快照 - 89%好评

MD5加密_百度百科

MD5的全称是Message-Digest Algorithm 5（信息-摘要算法），在90年代初由MIT Laboratory for Computer Science和RSA Data Security Inc的Ronald L. Rivest开发出...

作用 MD2 MD4 MD5 函数设计 四轮操作

baike.baidu.com/

MD5加密

MD5,加密,32位,16位,编码,算法,校验,解密,验证,破解... MD5加密 请将您要加密的内容复制到以下区域 以下为转换后的MD5密文 计算器 信用卡进度 旅游景点 知...

md5.supfree.net/ - 百度快照 - 评价

MD5加密工具,32位MD5在线加密

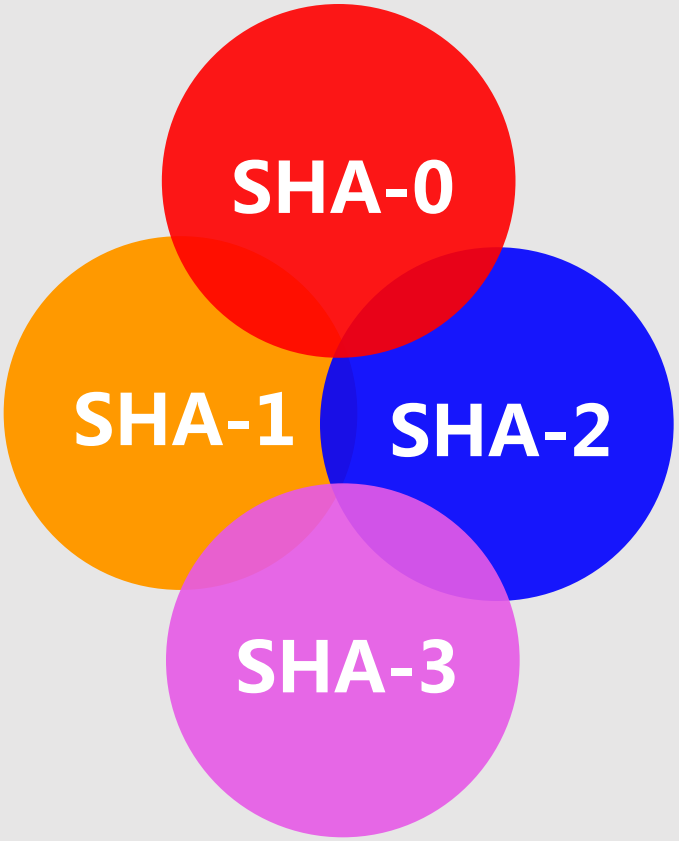
MD5加密工具可以对任意字符串实施32位MD5加密,以防止字符串被识别和更改。MD5是一种不可逆的加密算法,除了加密字符串,更广泛的用途是生成文件的"数字指纹",用于...

www.qqxiuzi.cn/bianma/... - 百度快照 - 评价

md5在线解密破解,md5解密加密

本站针对md5等全球通用加密算法进行反向解密,拥有全球最大的数据库,成功率全球第一,很多复杂密文只有本站才可解密,支持多种算法,实时查询记录超过24万亿条,共占用...

www.cmd5.com/ - 百度快照 - 81%好评



SHA: Secure Hash Algorithm	
发布者	NIST、NSA
应用领域	美国政府 (联邦标准, FIPS)

对MD5、SHA-0、SHA-1的破译

不再建议使用这些算法

SHA-2 系列

时间	2001-2014
发布者	NSA
相关文档	FIPS PUB 180-4

- **SHA-224**
- **SHA-256**
- **SHA-384**
- **SHA-512**

易遭到“长度扩展攻击”，建议使用：

SHA-512/224、SHA-512/256

比特币中使用 SHA-2 产生区块

SHA-3 系列

时间	2015
发布者	NIST
相关文档	FIPS PUB 202

- SHA3-224
- SHA3-256
- SHA3-384
- SHA3-512

4.2 消息认证码MAC

Q: 消息在传输过程中是否发生改变或被恶意篡改，我们怎么能知道？

使用“数据完整性技术”

当消息发生改变时，我们可以检查出来

Q: Hash函数能否保证数据的完整性？

举例:

某公司经网络发送一张订单m，为防范遭恶意篡改，先用Hash函数计算订单散列值H(m)，再将之与订单一起发送 [m,H(m)]。

接收方计算收到订单m'的散列值H(m')，若与收到的散列值相同，则确信订单未被篡改。

问此方法是否行得通？为什么？

不行！

原因：攻击者可先篡改订单,再计算假订单的散列值 (接收方无法识别)

关键问题：Hash函数没有密钥 (给定消息，任何人都可以计算)

可以看得出，如果设计一个带有密钥的算法，便可以解决这一问题，也就可以防范攻击者进行伪造

消息认证码(MAC)算法便是这种带密钥的算法	它是实现数据完整性的重要工具
	其产生的输出也相应地被称作MAC

消息认证码MAC　安全性

MAC的安全性要求——抗伪造：

1. 在不知道密钥的情况下，给定任何消息，产生相应的MAC是计算上不可行的

2. 即使已知很多消息及对应的MAC，对新消息产生MAC仍是计算上不可行的

消息认证码MAC　MAC的构造

Internet 标准

CBC-MAC

利用已有分组密码间接构造MAC

利用分组密码的CBC模式构造MAC的方法：

设$E_k(\cdot)$是一个分组长度为n的分组密码的加密算法。

对任意消息x，首先对x进行分组，每组的长度为n。设消息x为 $x=x_1\ x_2...x_r$

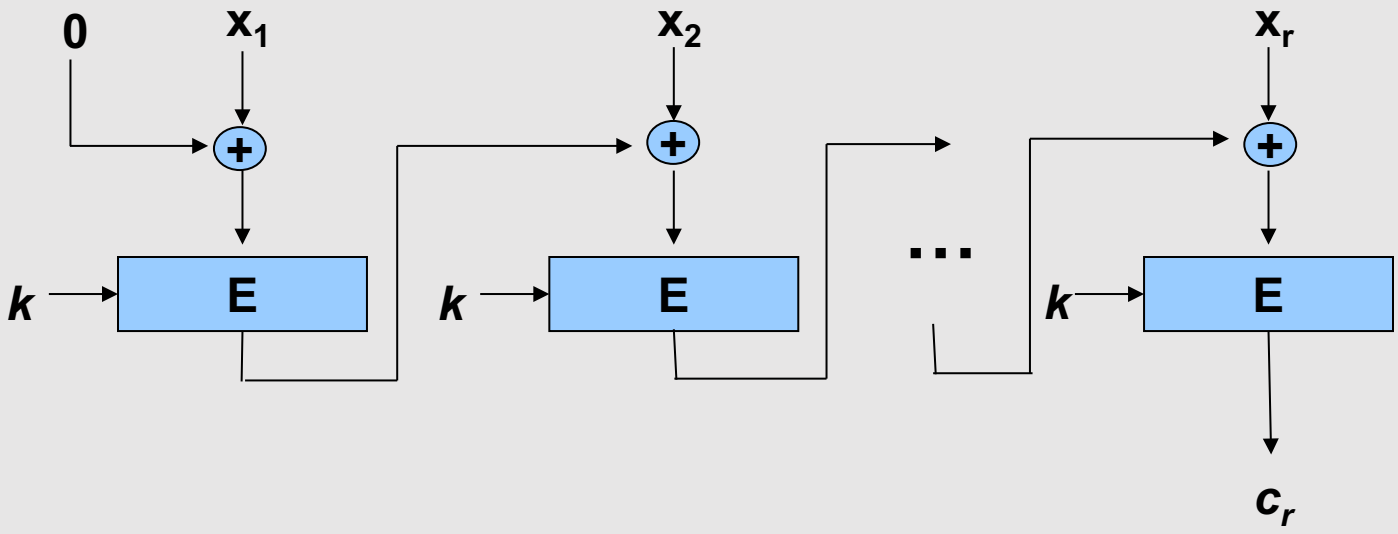
- 基于CBC模式构造MAC

① 选取初始向量: $c_0=IV=0$

② 依次计算:

$$c_i = E_k(x_i \oplus c_{i-1}) \quad (1 \le i \le r)$$

③ 输出散列值: c_r (只保留最后一个输出分组)



- **注意**
  - 该方法对于定长消息是安全的。
  - 如果消息长度可变，即使密钥K没泄露，同样不安全
    - 只要攻击者获得任意消息m=$x_1x_2...x_{r-1}$的MAC

$$CBC\text{-}MAC_K(x_1x_2\cdots x_{r-1})=c_{r-1}$$

    - 他任意找一个消息x_r(长度为一个分组)，便可产生一对碰撞
    - 因为 $m_1=x_1x_2\cdots x_r, m_2=c_{r\text{-}1}\oplus x_r$

$$CBC\text{-}MAC_K(x_1x_2\cdots x_r)=CBC\text{-}MAC_K(c_{r-1}\oplus x_r)=c_r$$

CBC-MAC只有用于定长消息才能抗伪造

Q: 如何让CBC-MAC对于变长消息也能抗伪造？

下面有几种改进方案

失败的改进方案

•方法：追加消息长度为最后一个分组

消息m的长度记为|m|，则

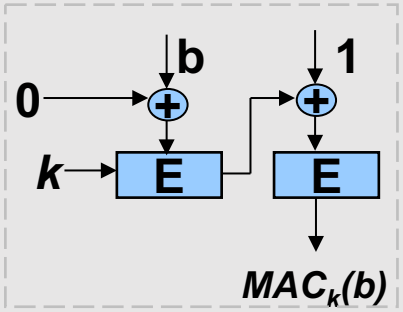
$MAC_K(m)=CBC\text{-}MAC_K(m|||m|)$

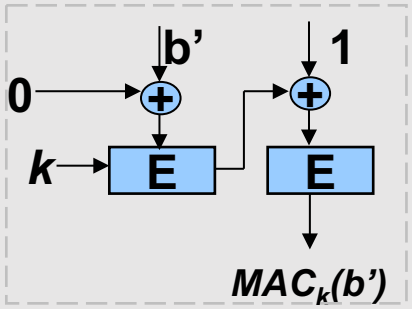
•但已被Bellare、Rogaway破译，所以不安全

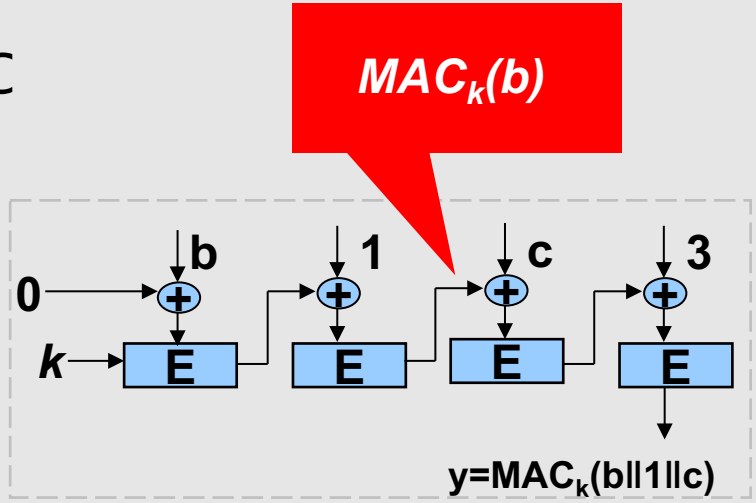
Mihir Bellare

Phillip Rogaway

- **伪造方法**
  - 攻击者任选3个一个分组长度的消息 b, b', c
  - 攻击者分别获得以下三个MAC



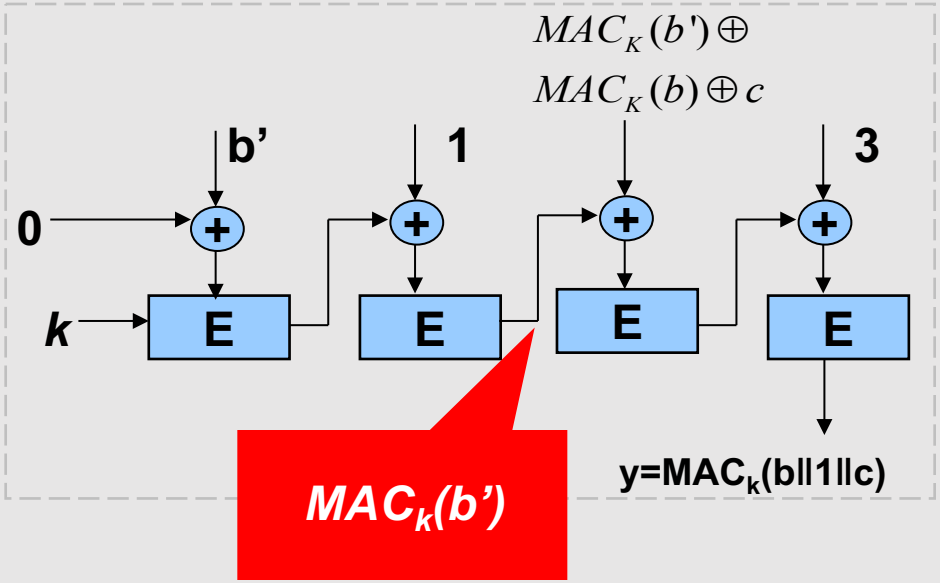




- **伪造：**

$$m = b' \| 1 \| MAC_K(b') \oplus MAC_K(b) \oplus c$$

$$MAC_K(m) = y$$



几个成功的改进方案

•改进方案1—— 前缀长度

– 将消息长度作为第一个分组

- $MAC_K(m)=CBC\text{-}MAC_K(|m| \,||\, m)$

缺点：必须首先知道消息的长度

•改进方案2—— 长度与密钥分离

– 将消息长度加密后作为计算MAC的密钥

$$MAC_K(m) = CBC - MAC_{E_K(|m|)}(m)$$

缺点：必须首先知道消息的长度

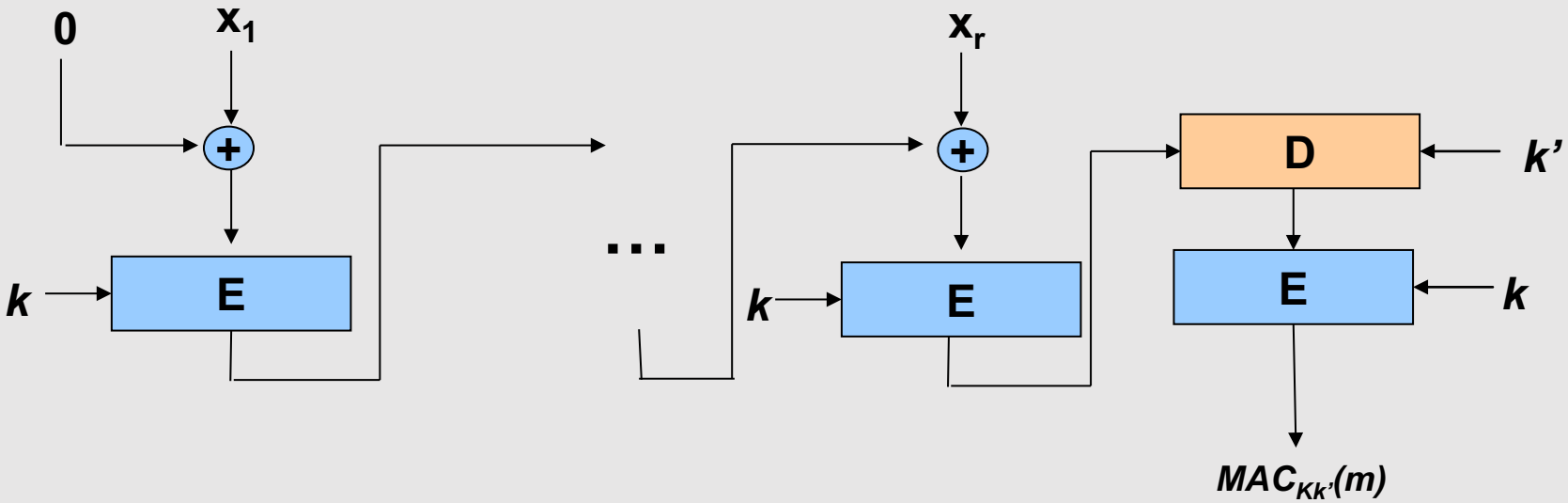
- **改进方案3 —— 再加密**

  $MAC_{KK'}(m) = E_{K'}(CBC\text{-}MAC_K(m))$

- 缺点：需要管理两个密钥
- 优点：不需要预先知道消息长度
- 相比之下，改造方案3最具吸引力

- **ISO标准的CBC-MAC** （改进方案3的改版——双密钥三重加密）
  - 设消息$m=x_1...x_r$

$MAC_{KK'}(m)=E_K(D_{K'}(CBC\text{-}MAC_K(m)))$



HMAC

基于分组密码构造法(比如CBC-MAC)是传统上构造MAC最为普遍的方法

近年来，构造MAC的研究兴趣已转移到基于Hash函数的构造方法，这是因为：

1. Hash函数的软件实现快于分组密码
2. Hash函数没有出口限制

现在已有很多基于Hash函数构造的MAC，HMAC就是其中最著名的一个

时间	1996
发明者	Bellare 等
相关文档	RFC2104 (1997年)，事实上的Internet标准
应用	IPSec协议等安全协议
构造方法	利用现有的Hash函数为构造模块 • HMAC-MD5：利用MD5构造HMAC • HMAC-SHA1：利用SHA-1构造HMAC

Mihir Bellare

其他MAC算法

基于分组密码的构造

3GPP-MAC：3G通信

XOR-MAC

RMAC

PMAC

基于Hash函数的构造

UMAC：速度快

NMAC

4.3 消息认证

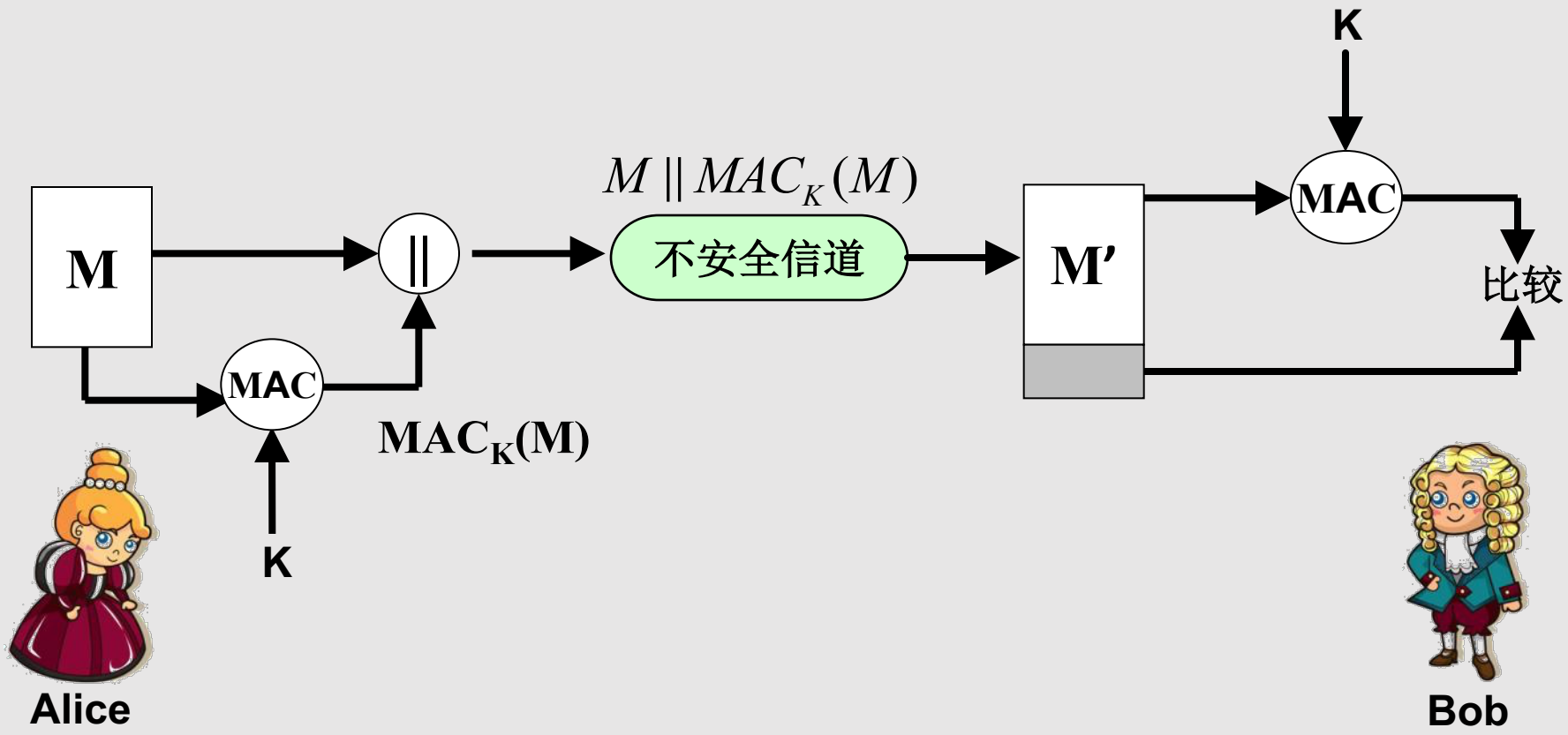
消息认证，又称数据源认证，它的重要目标：

1. 保证传输消息的完整性（检查消息是否被改动过）
2. 保证消息是由指定发送者发来的

主要技术

基于 MAC 的认证

基于MAC的认证



基于MAC的认证 (续)

如果Bob计算得到的MAC与接收到的MAC一致，则说明：

1. **防篡改**：Alice发送的消息未被篡改

   因为攻击者不知道密钥，无法在篡改消息后产生相应的MAC

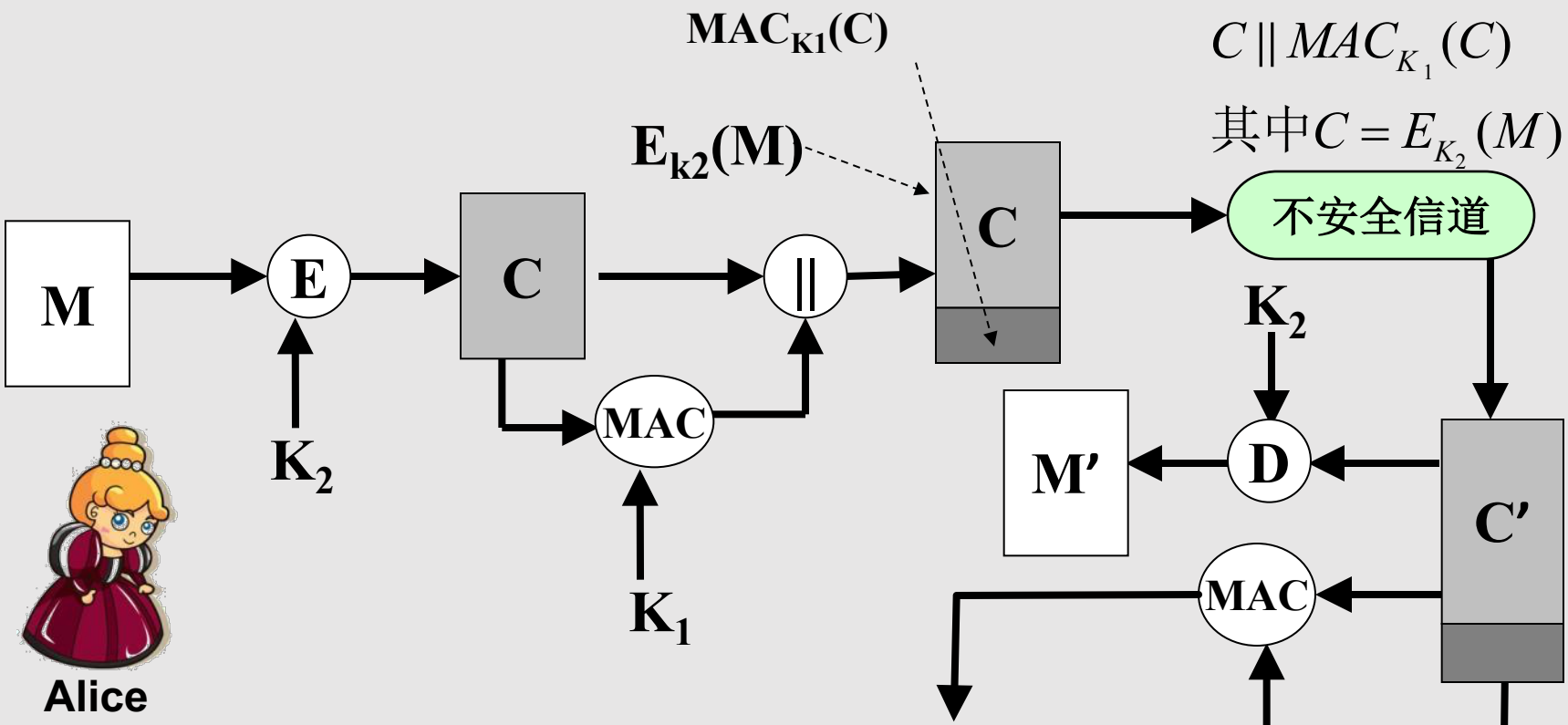
2. **防冒充**：Alice不是冒充的

   因为除收发双方外，没人知道密钥，攻击者无法冒充Alice发送合法的MAC

由于消息本身是明文形式，所以这一过程中未实现机密性。

为提供机密性，可在计算MAC之前或之后进行一次加密。

提供机密性的消息认证（方案1）

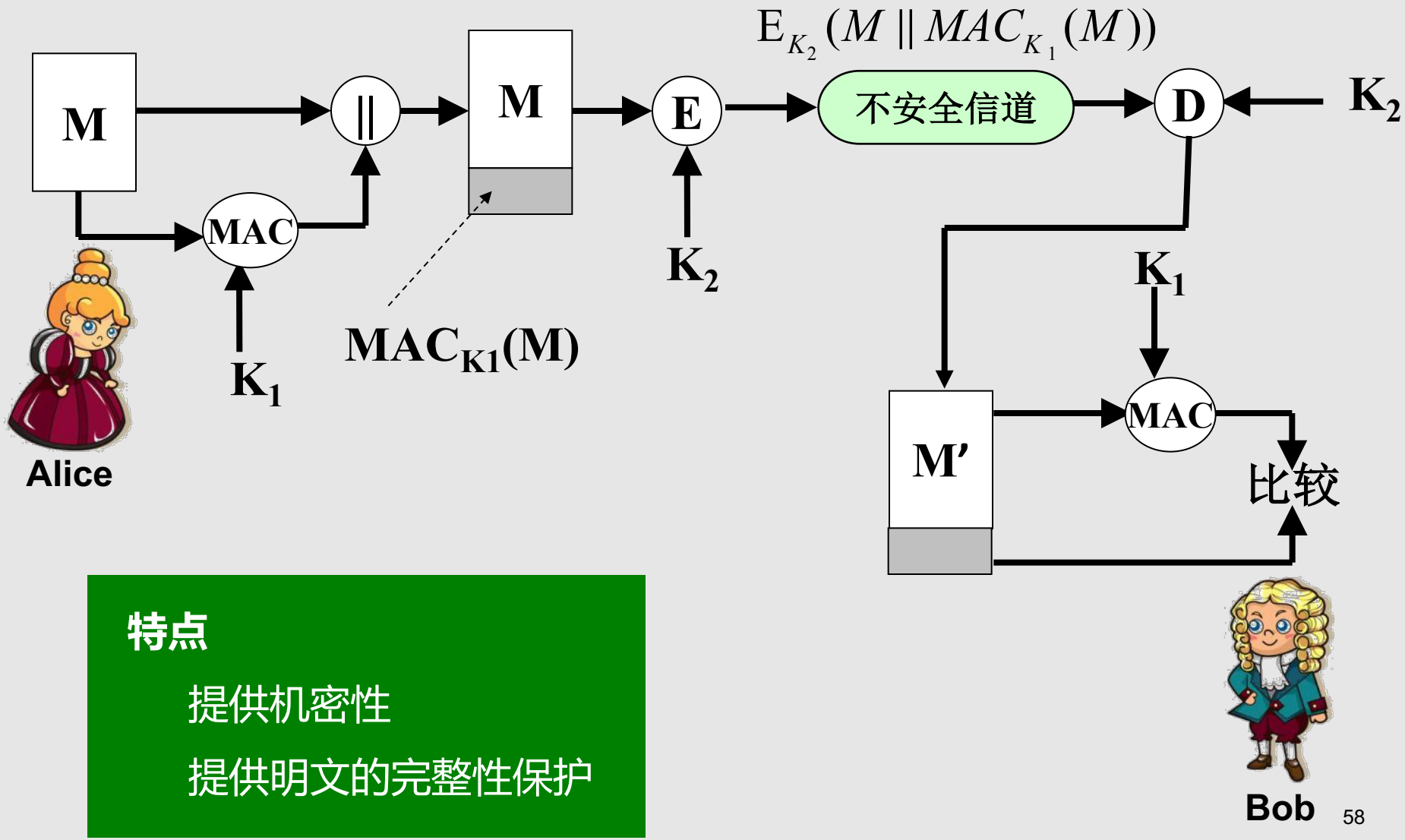


特点

提供机密性

提供密文的完整性保护

提供机密性的消息认证（方案2）



特点

提供机密性

提供明文的完整性保护

两种方案的比较

方案1

明文M先被加密，再与MAC一起发送

方案2

明文M与MAC被一起加密

通常，我们希望直接对明文进行认证，因此 **方案2** 的使用方式更为常用。

本章需要掌握和了解的内容

1. 掌握Hash函数的定义和分类、Hash函数的性质
2. 掌握MAC安全性的含义
3. 掌握两种消息认证技术的使用方法
4. 了解生日攻击的原理，以及对散列值长度的要求
5. 了解CBC-MAC的构造法，以及失败改进方案不安全的原因
6. 了解HMAC

练习题

1. Hash函数的安全性不包括哪个性质（ D ）

A. 单向　　B. 第二原像稳固　　C. 碰撞稳固　　D. 输出稳固

2. 下面哪个说法不正确（ C ）

A. 对Hash函数的攻击就是寻找一对碰撞的过程

B. 迭代构造Hash函数时，预处理过程必须是单射的

C. 对Hash函数的生日攻击说明，输出长度与其安全性无关

D. Hash函数具有压缩功能

3. MAC算法的功能是实现数据的（ B ）

A. 机密性　　B. 完整性　　C. 可用性　　D. 非否认