第11章 消息认证和散列函数

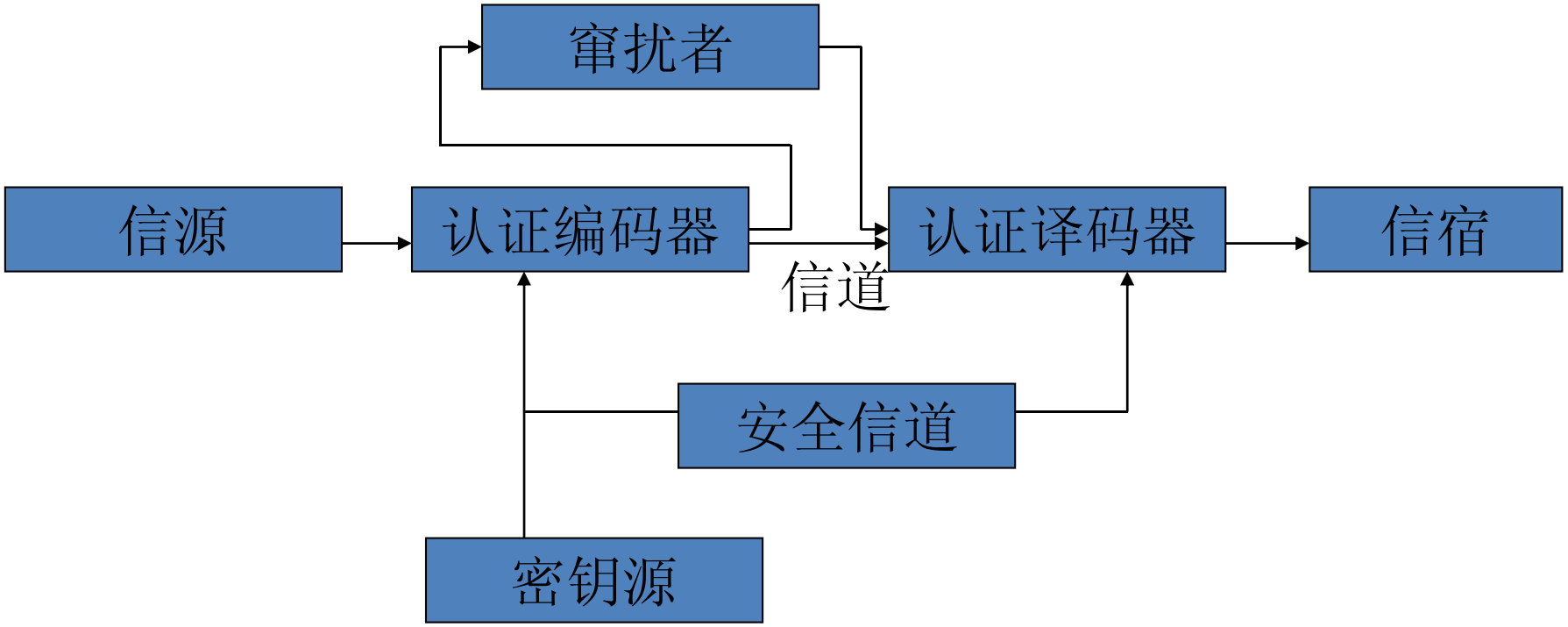
（Message Authentication and Hash Functions)

1 消息认证

网络系统安全要考虑两个方面。一方面，用密码保护传送的消息使其不被破译；另一方面，就是防止对手对系统进行主动攻击，如伪造，篡改消息等。认证（**authentication**）则是防止主动攻击的重要技术，它对于开放的网络中的各种信息系统的安全性有重要作用。认证的主要目的有二：

第一，验证消息的发送者是真正的，而不是冒充的，此为信源识别； 第二，验证消息的完整性，在传送或存储过程中未被篡改，重放或延迟等。

保密和认证同时是信息系统安全的两个方面，但它们是两个不同属性的问题，认证不能自动提供保密性，而保密性也不能自然提供认证功能。一个纯认证系统的模型如下图所示：



在这个系统中的发送者通过一个公开的无扰信道将消息送给接收者，接收者不仅想收到消息本身，而且还要验证消息是否来自合法的发送者及消息是否经过篡改。系统中的密码分析者不仅要截收和破译信道中传送的密报，而且可伪造密文送给接收者进行欺诈，将其称为系统的窜扰者**(tamper)**更加合适。实际认证系统可能还要防止收方、发方之间的相互欺诈。

上述标出的认证编码器和认证译码器可抽象为认证函数。一个安全的认证系统，首先要选好恰当的认证函数，然后在此基础上，给出合理的认证协议**(Authentication Protocol)**。

2 认证函数

（Authentication Functions)

可用来做认证的函数分为三类：

(1) 消息加密函数(Message encryption)

用完整信息的密文作为对信息的认证。

(2) 消息认证码MAC(Message Authentication Code)

定义域为信源消息和秘密钥，值域为定长比特向量的一个编码函数。所得向量可以作为认证码（符）。

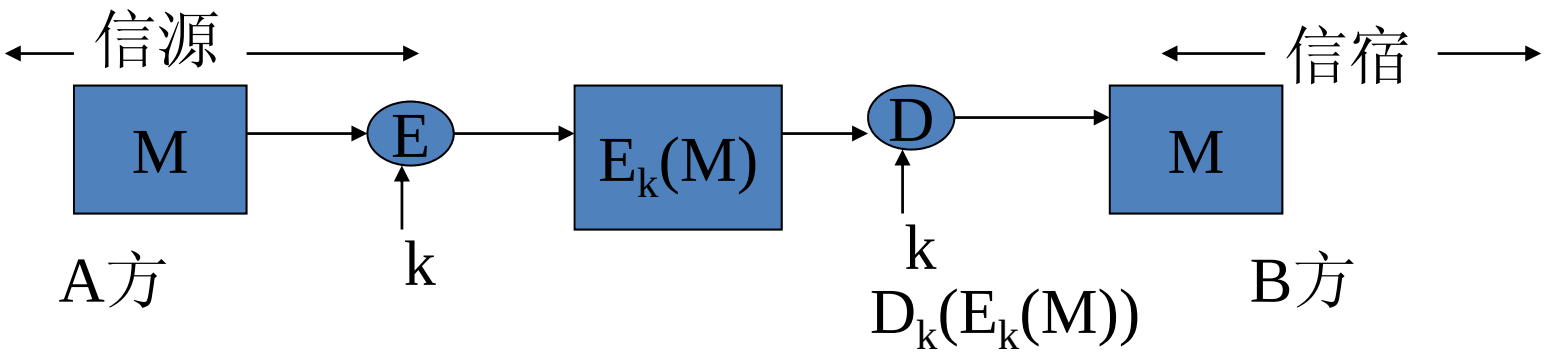
(3) 散列函数(Hash Function)

是一个公开的函数，它将任意长的信息映射成一个固定长度的信息。

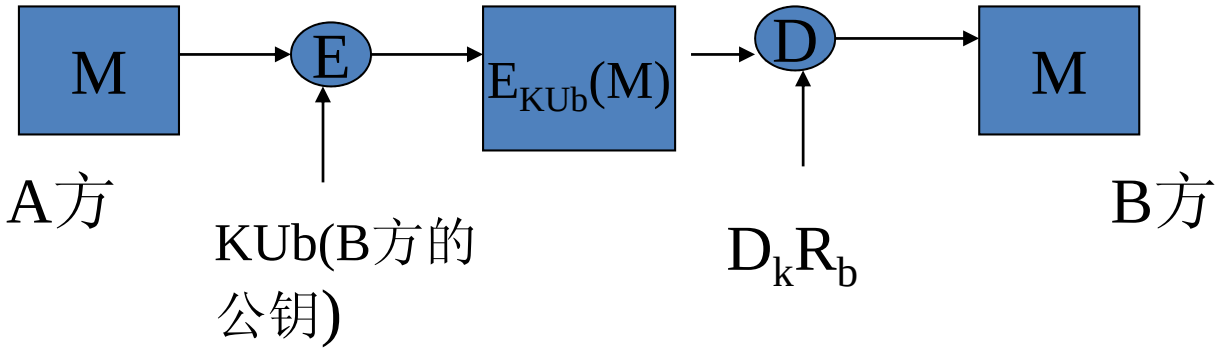
对于（1）用消息加密函数作认证的方式

消息加密函数分二种，一种是常规的对称密钥加密函数，另一种是公开密钥的双密钥加密函数。

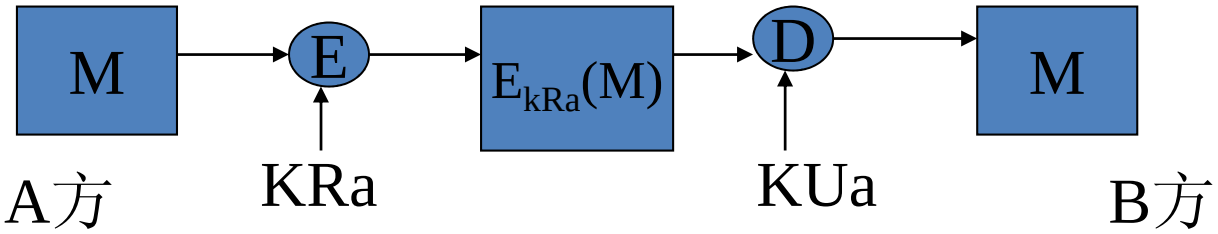
下图的通信双方是，用户A为发信方，用户B为接收方。用户B接收到信息后，通过解密来判决信息是否来自A，信息是否是完整的，有无窜扰。



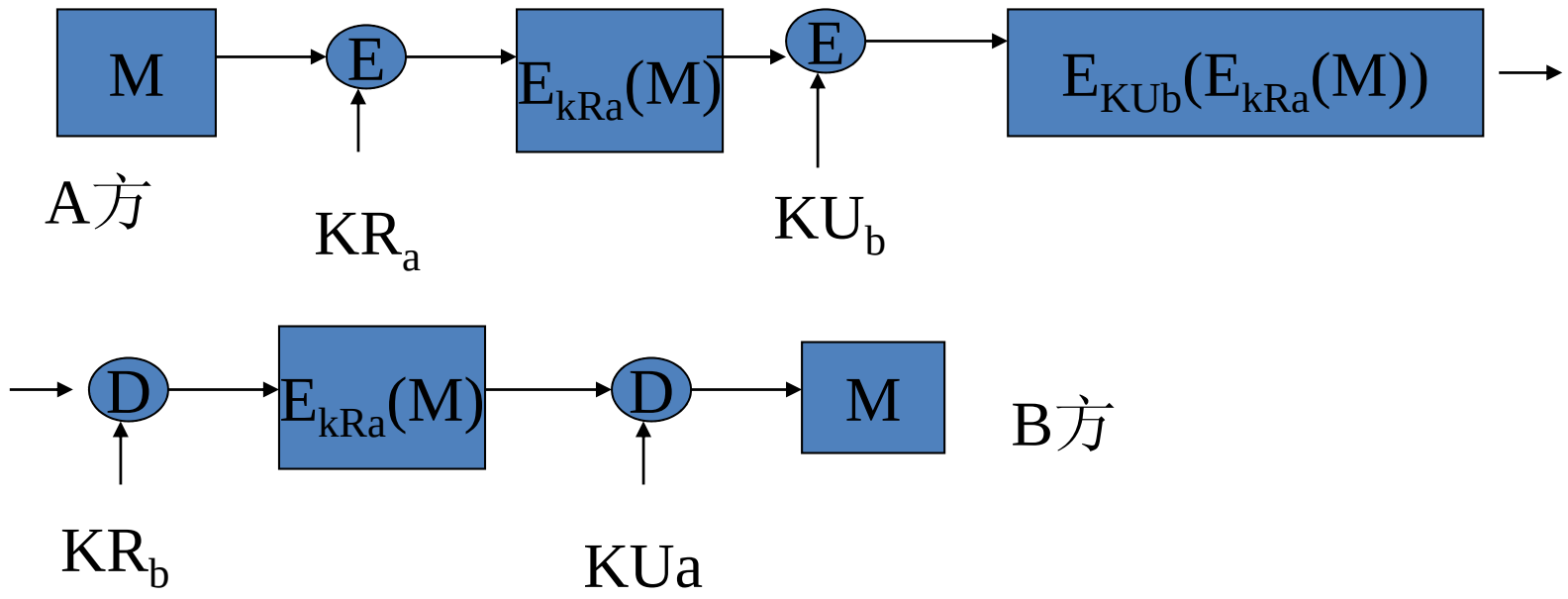
(a) 常规加密：具有机密性，可认证



(b) 公钥加密：具有机密性



(c) 公钥加密：认证和签名



(d) 公钥加密：机密性，可认证和签名

真实明文的自动判决使用桢效验序列**FCS(frame check sequence)=F(M)**或**F(E_k(M))**(分内外差错控制情形决定),用内差错控制,可使攻击者没有办法使假密文保持**FCS** 正确.

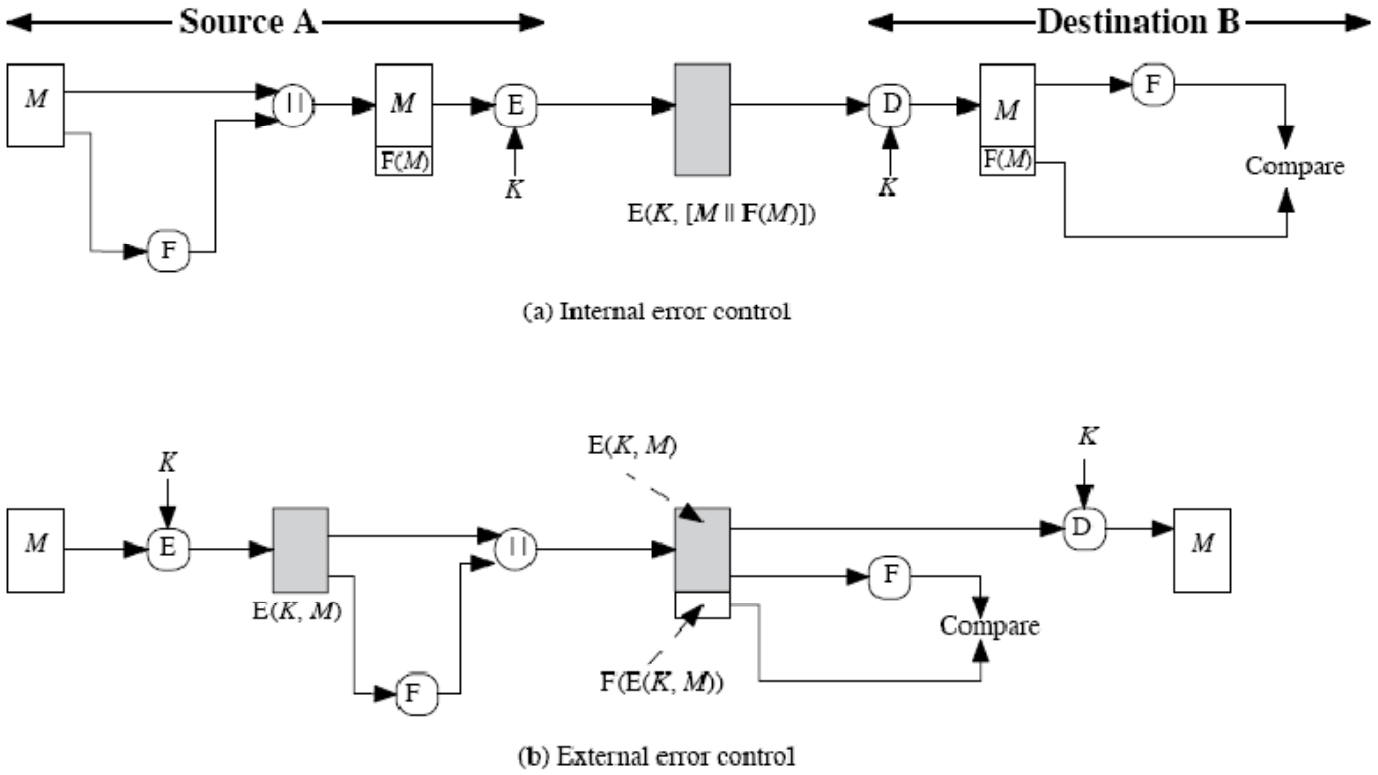


Figure 11.2 Internal and External Error Control

（２）消息认证码

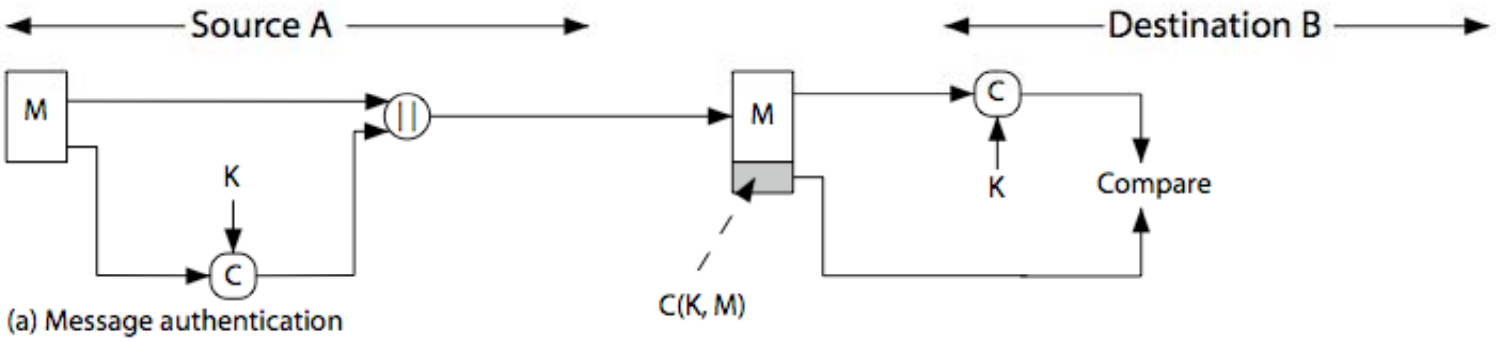
可用消息认证码**MAC**对消息做认证。**MAC**为定义域为信源消息和秘密钥集合，值域为定长比特向量的一个编码函数算得的向量值（认证码或认证符）；

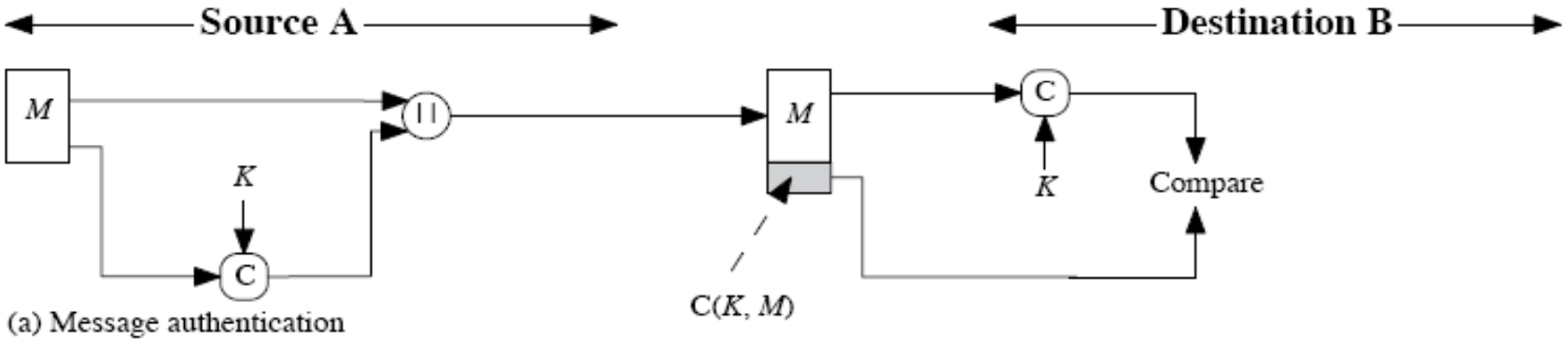
表示为　**MAC=C(K,M) .**

消息M和MAC一起被发送给接受方．接受方也计算**C(K,M)**来检验与收到的**MAC**是否一致？以此来判决：

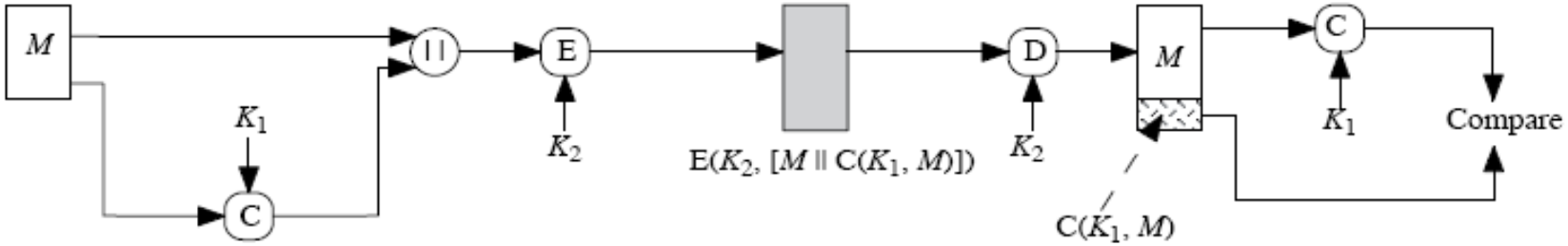
消息的源，内容的真伪，时间性和意定的信宿.

Message Authentication Code

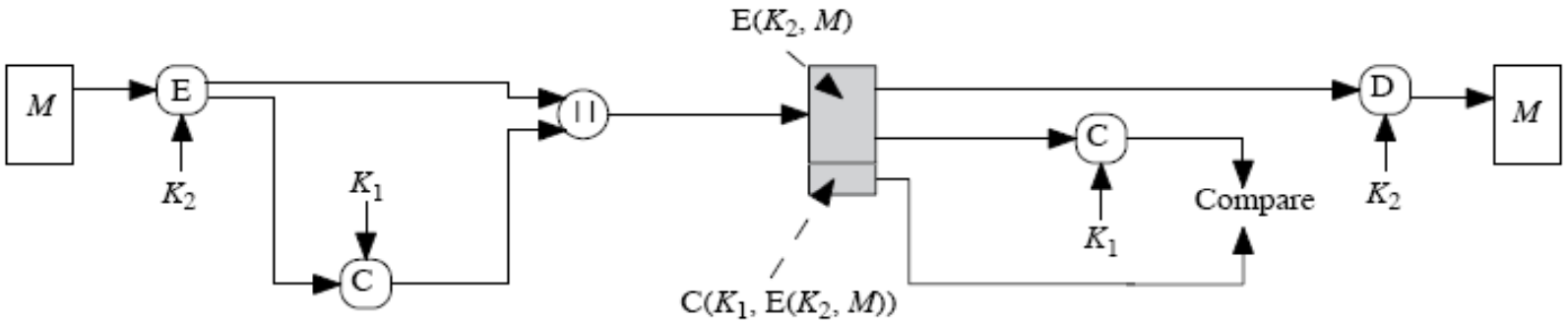




(a) Message authentication



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

Figure 11.4 Basic Uses of Message Authentication Code (MAC)

换言之，利用函数f和密钥k，对要发送的明文x或密文y变换成r bit的消息认证码f(k,x)(或f(k,y))，将其称为认证符附加在x(或y)之后发出，以x//As(或y//As)表示，其中“//”符号表示数字的链接。接收者收到发送的消息序列后，按发方同样的方法对接收的数据(或解密后)进行计算，应得到相应的r bit数据。

两种实用的MAC算法

(一)十进制移位加MAC算法

Sievi于1980年向ISO提出一项消息认证法的建议[Davies等1984]，这种认证法称为十进制移位加算法(Decimal Shift and Add Algorithm)，简记为DSA。它特别适用于金融支付中的数值消息交换业务。

消息按十位十进制数字分段处理，不足十位时在右边以0补齐，下面举例说明。令x_1=1583492637是要认证的第一组消息，令b_1=5236179902和b_2=4893524771为认证用的密钥。DSA算法是以b_1和b_2并行对x_1进行运算。

先算x_1+b_1, $x_1+b_2(\mod 10^{10})$, 而后根据b_2的第一位数值4对x_1+b_2循环右移4位，记作$R(4)(x_1+b_1)$再与(x_1+b_1)相加得

$$R(4)(x_1+b_1)+(x_1+b_1)\equiv P_1(\mod 10^{10})$$

类似地，右路在b1的第一位数值5控制下运算结果为

$$R(5)(x_1+b_2)+(x_1+b_2)=Q_1(\mod 10^{10})$$

表：

	左路	右路
第一轮	b_1=5236179902	b_2=4893224771
	+ x_1=1583492637	+ x_1=1583492637
	b1+x1=6819672539	b2+x1=6477017408
	+R(4)(b1+x1)=2539681967	+R(5)(b2+x1)=1740864770
	P1=9359354506	Q1=8217882178

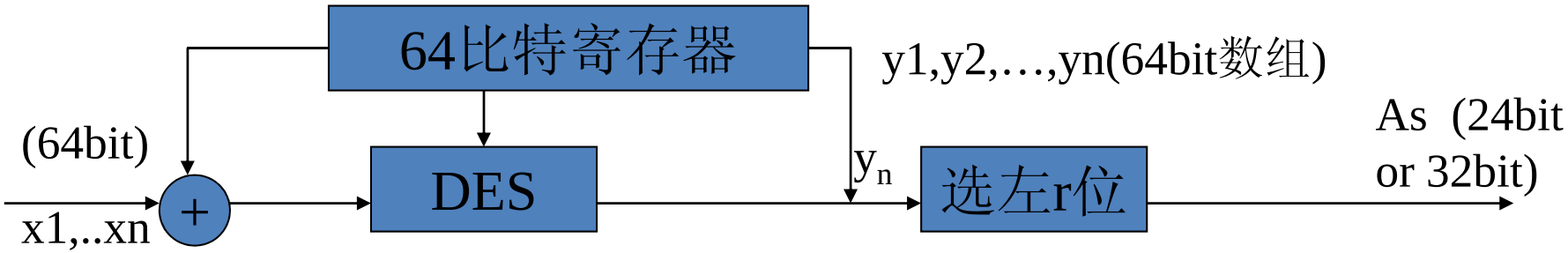
第二轮	+ x2=5283586900	+ x2=5283586900
	P1+x2=4642941406	Q1+x2=3501469078
	+R(8)(P1+x2)=4294140646	+R(9)(Q1+x2)=7835014690
	P2=8937082052	Q2=1336483768
	….	….
第十轮	P10=7869031447	Q10=3408472104
	P10+Q10=1277403551 (mod 10^{10})	
	403551	
	+ 1277	
	认证符 404828 (mod 10^{10})	

将第二组消息x_2=528358586900分别与P1和 Q1相加，其结果又分别以P1和Q1的第一位控制循环移位后进行相加得到P2和Q2，依此类推。直至进行十轮后得P10和Q10。计算P10+Q10 (mod 10^{10}),并将结果的后6位数与前4位数在

(mod 10^{10})下相加，得6位认证符！

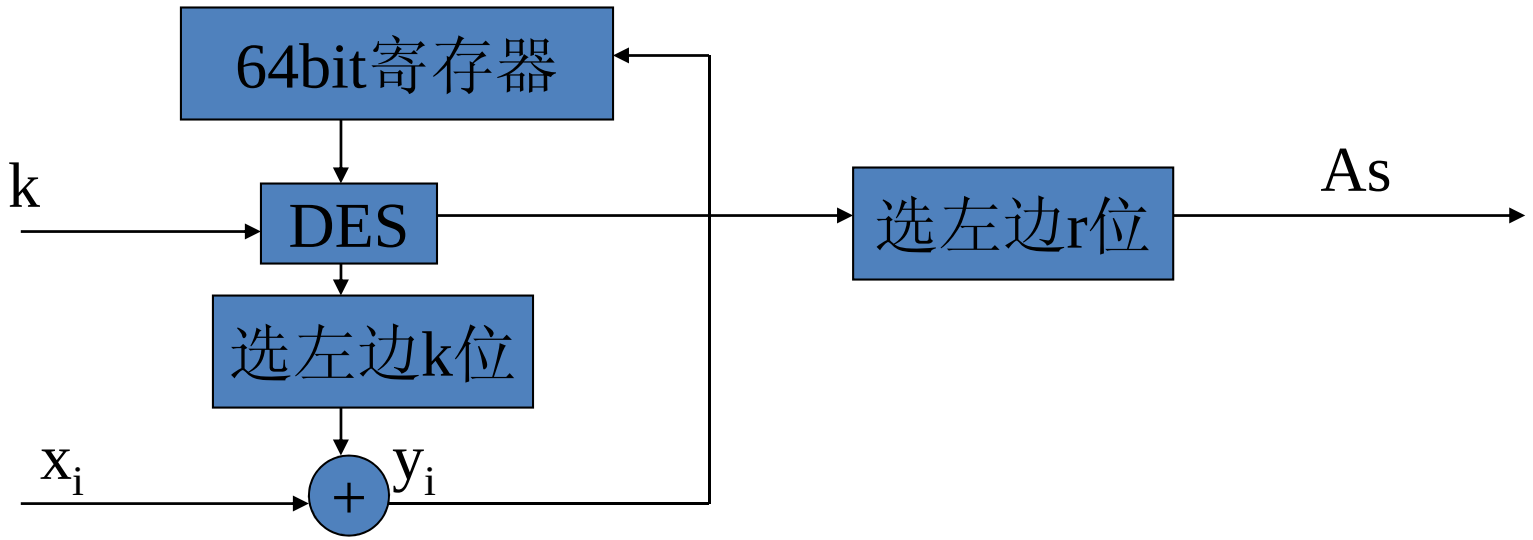
(二)采用DES的认证算法：

有二种基于DES的认证算法，一种按CFB模式，一种按CBC模式运行。在CBC模式下，消息按64bit分组，不足时以0补齐，送入DES系统加密，但不输出密文，只取加密结果最左边的r位作为认证符。



利用CBC模式下DES的认证法

r取大小可由通信双方约定。美国联邦电信建议采用24bit[见FTSC-1026],而美国金融系统采用32bit [ABA,1986]



利用工作于CFB模式下DES

（3） 散列函数(Hash Function)

若对相当长的文件通过签名认证怎么办？如一个合法文件有数兆字节长。自然按160比特分划成一块一块，用相同的密钥独立地签每一个块。然而，这样太慢。

解决的办法：引入可公开的密码散列函数(Hash function)。它将取任意长度的消息做自变量，结果产生规定长度的消息摘要。[如，使用数字签名标准DSS，消息摘要为160比特]，然后签名消息摘要。对数字签名来说，散列函数h是这样使用的：

消息： x 任意长

消息摘要： Z=h(x) 160bits

签名： $y=sig_k(Z)$ 320 bits

(签名一个消息摘要)

验证签名:(x,y),其中$y= sig_k(h(x))$，使用公开的散列函数h，重构作Z'=h(x)。然后检验$V_{erk}(Z,y)$,来看Z?=Z'。

(一)无碰撞的散列函数

用以认证的散列函数，能否减弱认证方案的安全性？这个问题是要分析的。签名的对象由完整消息变成消息摘要，这就有可能出现伪造。

(a)伪造方式一：Oscar以一个有效签名(x,y)开始，此处$y= sig_k(h(x))$。首先他计算Z=h(x)，并企图找到一个$x'\neq x$满足h(x')=h(x)。若他做到这一点，则(x',y)也将为有效签名。为防止这一点，要求函数h具有无碰撞特性。

定义1(弱无碰撞)，散列函数h称为是弱无碰撞的，是指对给定消息$x \in X$，在计算上几乎找不到异与x的$x' \in X$使h(x)=h(x')。

(b)伪造方式二:Oscar首先找到两个不同的消息x和x‘,使满足h(x)=h(x'),然后Oscar把x 给Bob且使他对x的摘要h(x)签名，从而得到y,那么(x',y)是一个有效的伪造。

定义2(强无碰撞)散列函数h被称为是强无碰撞的,是指在计算上几乎不可能找到相异的x，x'使得h(x)=h(x')。

注：强无碰撞自然含弱无碰撞！

(c)伪造方式三：在某种签名方案中可伪造一个随机消息摘要Z的签名。若h的逆函数h^{-1}是易求的，可算出$h^{-1}(Z)=x$, 满足x=h(Z)，则(x,y)(其中$y=sig_k(h(x))$)为合法签名。

定义3(单向的)称散列函数h为单向的，是指计算h的逆函数h^{-1}在计算上不可行。

下面要证明“强无碰撞特性包含有单向性”。为此，先对散列函数h做一规范说明：

首先h:X Z，这里设X,Z为有限集且|X|>=2|Z|。这是一个合理的假设：若X中的元素编码长度为$\log_2|X|$的比特串，Z中的元素编码长度为$\log_2|Z|$的比特串。那么消息摘要Z=h(x)至少比源消息X短一个比特。

定理：假设h:X→ Z为一个散列函数，这里|X| 和|Z|是有限的且|X|>=2|Z|。若h^{-1}为h的逆函数，那么存在一个概率的Las Vegas算法，它能找到h的一个碰撞的概率至少为1/2。

证明：利用h的逆函数h^{-1}来寻找h的碰撞的Las Vegas概率算法：

(1) 随机选择x ∈ X

(2)计算z=h(x)

(3)计算$x_1 = h^{-1}(z)$

(4)若$x_1 \neq x$,那么x_1和x在h下碰撞成功。

否则，往复再来。来看成功的概率：

首先定义X中元素关于h的等价，若$x_1, x \in X$,有$h(x)=h(x_1)$, 则称x_1与x等价；记为x等价于x_1。

等价类$[x]=\{x_1 \in X \mid x$等价于$x_1\}$

因为，每一个等价类[x]由Z的元素的原象组成，所以等价类的数目至多为|Z|。记等价类的集合为C。现假设x是第一步选择的X中的元素。对这个x,在第(3)步中将返回|[x]|个可能的x_1值，而|[x]| -1个x_1值将与x不同。于是在[x]类内成功的概率为(|[x]| -1)/|[x]|。

对于前述算法成功的概率是通过平均所有可能x的选择来计算的：

$$P(\text{成功})=(1/|X|)\sum_{x\in X}((|[x]|-1)/|[x]|)$$

$$=(1/|X|)\sum_{c\in C}\sum_{x\in c}((|c|-1)/|c|)$$

注意，C为子类c的集合；c=[x].

$$=(1/|X|)\sum_{c\in C}(|c|-1)$$

$$=(1/|X|)(\sum_{c\in C}|c|-\sum_{c\in C}1)$$

$$\geq(|X|-|Z|)/|X| \geq(|X|-|X|/2)/|X|=1/2$$

因此，构造了一个成功率至少为1/2的Las Vegas算法。

注：说明强无碰撞与单向性的关系是，单向性含于强无碰撞之中

(二)生日攻击：

任找23人，从中总能选出两人具有相同生日的概率至少为1/2。

假设$h:X\rightarrow Z$是一个散列函数，X和Z是有限的，且$|X|>=2|Z|$，记$|X|=m$和$|Z|=n$，易见至少有n个碰撞;问题是如何找到它们？

自然的方法是，在X中随机选择k个不同元素$x_1,x_2,\ldots,x_k \in X$，计算$Zi=h(x_i)$，$1<=i<=k$。

然后确定一个碰撞是否已发生(例如：通过分类Zi的值)。

这个过程类似于随机抛k个球进入n个箱子，然后检查一下是否有一些箱子包含了至少两个球(这k个球对应于k个随机值x_i，且n个箱子对应于Z中n个元素）

用上述模型来计算一个碰撞的概率下界。这个下界仅依赖于k和n，但不依赖于m。预先做一个假设：

任意z ∈ Z,$|h^{-1}(Z)|\approx m/n$(这个假设是合理的，若对任意z ∈ Z的原象个数分布不均匀，则找1个碰撞的概率将增加)。由这些假设可推断xi的象$h(x_i)=Zi(i=1,2,\ldots,k)$也可视为Z中的随机元素。(这k个随机元，也可有相同的)。一个重要的问题是：

若计算这k个随机元素$z_1,z_2,..,z_k$ ∈ Z两两不同(无碰撞时)，对应于初始问题—无碰撞的概率：

考虑有序$z_1,z_2,\ldots,z_k$中的z_i的选择可能，第一个选择z_1是随机的，$z_2\neq z_1$的概率为1-1/n;z_3不同于z_1和z_2的概率为1-2/n,等等。

因此，无碰撞的概率为(随机抛k个球进入n个箱子，没有箱子被抛进两个以上的球的概率)

$$(1-1/n)(1-2/n)\ldots(1-(k-1)/n)=\prod_{i=1}^{k-1}(1-i/n)\approx 1-e^{(-(k(k-1)/(2n))}$$

注：若设$\varepsilon=1-e^{(-(k(k-1)/(2n))}$,可解出k的关于n, ε的函数，有$e^{(-(k(k-1)/(2n))}=1-\varepsilon$，$(-(k(k-1)/(2n))\approx\ln(1-\varepsilon)$,

$k^2-k\approx 2n\ln(1-\varepsilon)^{-1}$

若略去-k项，有$k\approx (n(\ln(1/(1-\varepsilon))\times 2)^{0.5}$

若取$\varepsilon=0.5$,我们估计$k\approx 1.17\ n^{0.5}\approx n^{0.5}$

说明，在集合X中，$n^{0.5}$个随机元素散列的结果产生一个碰撞的概率为50%!

所谓生日攻击就是：如果X为一些人的集合，Y是一个非闰年的365天集合，h(x)表示x的生日。($x \in X$)。在上述估计式中取n=365, 得k≈22.3，即随机的23人中至少有一个重复生日的概率至少为50%。

生日攻击给出消息尺寸的下界。一个**40bit**的消息摘要将是不安全的。因为$k \approx 1.17 \cdot (2^{40})^{0.5}$, 也就是$\approx 2^{20}$(大约**100**万)，即在**100**万个随机散列值中将找到一个碰撞的概率为**1/2**。通常建议，消息摘要的尺寸为**128bits**。