- 散列函数的选取不能减弱签名方案的安全性，它必须能够防止他人伪造。因此一个好的散列函数应该具有以下特性：
- 弱无碰撞：给定一个消息x，找到一个满足$h(x') = h(x)$ 的消息$x' \neq x$ 是计算上不可行的；
- 强无碰撞：找到满足$h(x') = h(x)$ 并且$x' \neq x$ 的消息x'和 x是计算上不可行的；
- 单向：给定一个消息摘要z，找到一个满足$h(x) = z$ 的消息x 是计算上不可行的。

- Hash函数如何构造，以达到前面的要求？
- 一个最基本必须满足的要求应该是任何输入串中单个比特发生变化，将会导致输出比特串中大约一半的比特发生变化。
- 构造方法主要有以下几种
- 利用某些对称密码体制，设计Hash函数
- 利用某些数学难题假设，设计Hash函数，可以在某些难问题假设下证明是强务碰撞的。
- 直接设计Hash函数。

- **4.2.2 可证安全的散列函数**
- 一个散列函数是这样设计的：p是一个大素数，且q=(p-1)/2也是素数， 设α和β是域Zp的本原元，值$\log_\alpha\beta$不公开，且假设求$\log_\alpha\beta$在计算上是不可行的。定义散列函数h: Zq×Zq→Zp-{0}为
- $h(x_1,x_2)=\alpha^{x_1}\beta^{x_2} \bmod p$
- 定理：若给定该散列函数的一个碰撞，则$\log_\alpha\beta$一定能有效计算。

- **4.2.3 MD5算法**
- **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开发出来，经MD2、MD3和MD4发展而来。**
- **它的作用是让大容量信息在用数字签名软件签署私人密匙前被“压缩”成一种保密的格式（就是把一个任意长度的字节串变换成一定长的大整数）。不管是MD2、MD4还是MD5，它们都需要获得一个随机长度的信息并产生一个128位的信息摘要。**

- 虽然这些算法的结构或多或少有些相似，但MD2的设计与MD4和MD5完全不同，那是因为MD2是为8位机器做过设计优化的，而MD4和MD5却是面向32位的电脑。
- Rivest在1989年开发出MD2算法。在这个算法中，首先对信息进行数据补位，使信息的字节长度是16的倍数。然后，以一个16位的检验和追加到信息末尾。并且根据这个新产生的信息计算出散列值。后来，rogier和chauvaud发现如果忽略了检验将产生碰撞。MD2算法的加密后结果是唯一的--既没有重复。

- 为了加强算法的安全性，rivest在1990年又开发出MD4算法。MD4算法同样需要填补信息以确保信息的字节长度加上448后能被512整除（信息字节长度mod 512 = 448）。然后，一个以64位二进制表示的信息的最初长度被添加进来。信息被处理成512位迭代结构的区块，而且每个区块要通过三个不同步骤的处理。Den boer和Bosselaers以及其他人很快的发现了攻击MD4版本中第一步和第三步的漏洞。Dobbertin向大家演示了如何利用一部普通的个人电脑在几分钟内找到MD4完整版本中的碰撞。毫无疑问，MD4就此被淘汰掉了。

- 1991年，Rivest开发出技术上更为趋近成熟的MD5算法。它在MD4的基础上增加了“安全-带子”（safety-belts）的概念。虽然MD5比MD4稍微慢一些，但却更为安全。这个算法由四个和MD4设计有少许不同的步骤组成。在MD5算法中，信息-摘要的大小和填充的必要条件与MD4完全相同。Den boer和Bosselaers曾发现MD5算法中的假碰撞（pseudo-collisions）。

- *MD5*算法生成了四个32比特长的字，在实际应用中可以将这四个字连接起来形成一个128比特长的字，作为明文压缩的结果。
- 如果位数仍不能达到要求，则只要对上述算法及程序稍加修改即可。
- 2004年8月17日的美国加州圣巴巴拉，正在召开的国际密码学会议（Crypto’2004）安排了三场关于杂凑函数的特别报告。来自山东大学的王小云教授做了破译MD5、HAVAL-128、 MD4和RIPEMD算法的报告

- 王小云发现，可以很快的找到MD5的“碰撞”，由于版本问题，作者在提交会议论文时使用的一组常数和先行标准不同；
- 在会议发现这一问题之后，王小云教授立即改变了那个常数，在很短的时间内就完成了新的数据分析，这更加证明了论文的信服力，攻击方法的有效性，凸显了研究工作的成功

- MD5的设计者，同时也是国际著名的公钥加密算法标准RSA的第一设计者R．Rivest在邮件中写道：“这些结果无疑给人非常深刻的印象，她应当得到我最热烈的祝贺，当然，我并不希望看到MD5就这样倒下，但人必须尊崇真理。”
- MD5破解工程权威网站http://www.md5crk.com/是为了公开征集专门针对MD5的攻击而设立的，网站于2004年8月17日宣布：“中国研究人员发现了完整MD5算法的碰撞；Wang, Feng, Lai与Yu公布了MD5、MD4、HAVAL-128、RIPEMD-128几个 Hash函数的碰撞。这是近年来密码学领域最具实质性的研究进展。使用他们的技术，在数个小时内就可以找到MD5碰撞。……由于这个里程碑式的发现，MD5CRK项目将在随后48小时内结束”

- 攻击可以发现很多对具有相同原始初始值 IV_0 的两个 1024 位 MD5 消息：
- IV_0: $A_0 = 0x67452301$, $B_0 = 0xEFCDAB89$, $C_0 = 098BADEFD$, $D_0 = 0x10325476$
- $M' = M + \Delta C_1$, $\Delta C_1 = (0, 0, 0, 0, 2^{31}, ..., 2^{15}, ..., 2^{31}, 0)$
- $N_i' = N_i + \Delta C_2$, $\Delta C_2 = (0, 0, 0, 0\ 2^{31}, ..., -2^{15}, ..., 2^{31}, 0)$
- （位置4、11和14非零）
- 此时，$MD5(M, N_i) = MD5(M', N_i')$
- 在 IBM P690 上，用将近一个小时的时间来寻找这样的 M 和 M'，之后，只需要 15 秒到 5 分钟的时间就可以找到 N_i 和 N_i'，此时的 (M, N_i) 和 (M', N_i') 将产生相同的散列值。此外，攻击可以工作于任意给定的初始值。
- 防止这种攻击的方法是在所有明文中间加0

- **4.2.4 SHA算法**
- **SHA(Secure Hash Algorithm)算法是美国NIST和NSA设计的一种标准算法，它具有较高的安全性。**
- **1992年公布并使用，1994年作了改进，1995年公布了新版即SHA-1。**
- **SHA算法最终输出结果是160bit的信息摘要。该算法首先填充信息，使之为512bit的倍数。**
- **填充方法与MD5完全一样。**
- **5个32bit变量初始化为：A=67 45 23 01,B=EF CD AB 89,C=98 BA DC FE,D=10 32 54 76, E=C3 D2 E1 F0。**
- **然后进行算法的主循环。它一次处理512bit信息，循环次数是信息中的512bit块的数目。**

- 在MD5被攻破之后，世界密码学界仍然认为SHA-1是安全的算法。2005年2月7日，美国国家标准技术研究院（NIST）对外宣称，SHA-1还没有被攻破，并且也没有足够的理由怀疑它会很快被攻破。
- SHA-1的应用范围或许比MD5更加广泛，其安全性较MD5要高出很多。SHA-1是美国国家标准技术研究院（NIST）与美国国家安全局（NSA）共同设计的，一些重要的场合都选择SHA-1来做数字签名。美国政府更是早在1994年就开始采用了SHA-1算法。
- 仅在一周之后，SHA-1也被王小云“碰撞”了。
- 原来代价为2^{80}，现为2^{60}，但并不意味着实际的攻破

- 这一进展，在国际社会的反响甚至超出半年前MD5被破时的情景。NIST表示，美国政府5年内将不再使用SHA-1，并计划在2010年前改用先进的SHA-224、SHA-256、SHA-384及SHA-512的数字签名加密算法。
- 在我国，MD5和SHA-1也是在实际应用中最广泛的两种数字签名算法，包括网上银行等金融业务在内的很多数字签名都采用SHA-1或MD5算法。
- 王小云教授发明的可以迅速而有效地验证一系列Hash函数算法健壮性的工具，令Hash函数的一些隐含弱点更快地暴露在人们眼前，在学术研究上具有更大的理论价值。
- 构造实用安全的散列函数

- 由于王小云教授提出的比特追踪法和明文修改技术在分析Hash函数上取得的成功,使得现行的标准Hash函数中的MDx系列和SHA系列的安全性都不能够再满足实际的要求。
- NIST于2007年发起了Hash函数新标准的征集活动,即SHA-3竞赛。
- NIST 对提交算法的要求和建议主要有:
  (1) 新算法应该能在广泛的软硬件平台上安全有效地实现,包括受限的环境如智能卡

- (2) 新算法应该能支持224bit、256bit、384bit 和512bit 摘要, 支持的最大消息长度至少为(264- 1) bit;
- (3) 新算法应该支持HMAC、PRF、随机化hashing(randomized hashing) ;
- (4) 新算法应该能够抵抗碰撞攻击、原象攻击、第二原象攻击和长度扩展攻击(length□ extension attacks) 等,
- (5) 新算法应该有简单灵活的设计, 可以通过并行实现获得更高的性能效率;
- (6) 新算法应该能够简单地代换SHA-2, 需要保持SHA-2 的一些性质如输入参数、输出大小、安全性要求等;
- (7) 新算法可以有可调的安全参数,如轮数, 从而可以进行安全性和性能的折中;
- (8) 新算法可以使用不同于传统Merkle-Damgard 结构的新结构, 避免对MD 结构的通用攻击.

- 2008年10月31日收到了来自世界各个密码组织或个人提交的64份候选提案。
- 2009年经过最低标准的鉴定,其中有51个算法进入了第一轮候选。
- 2010 年从首次SHA-3候选算法会议后挑选了14个算法进入SHA-3第二轮候选。目前第二轮的评估工作正在进行中。
- 最后将在2012 年公布

- 算法总体的特点
- (1) 设计结构的多样性，传统的Hash 函数MD4、MD5、SHA-1 等采用的都是Merkle-Damgard 迭代结构,
- MD 结构的优点是有抗碰撞保持的性质. 然而 目前已有针对MD 结构的通用攻击，新提交的SHA-3 候选算法使用的结构有多种, 主要有修改的MD 结构、HAIFA 框架、Sponge 结构、流模式等..
- (2) 安全性设计，新的候选算法很多都使用了分组密码和流密码的设计思想和设计方法
- AES 的设计方法和设计原理得到大量应用.
- 由于AES 在安全性和实现性能上有着优秀的表现,并且已经有多年的研究结果, 很多SHA-3 候选算法都借鉴了AES 的成果,
- 有的直接使用AES 的部件, 有的根据AES 的原理设计新的SPN 结构.
- 通过吸收借鉴AES抵抗差分攻击、线性攻击和有效实现方面的设计, 提高Hash 算法抵抗差分攻击等的能力, 提高安全性.

- (3) 性能设计，随着多核芯片和并行计算机的逐渐流行和普及, 并行程序和并行算法的设计也愈显重要.
- 并行计算的一个优势是能够加速计算, 很多SHA-3 候选算法都不同程度地考虑到了并行计算的应用. 有操作模式的并行性如树结构、压缩函数的并行性、指令级并行性如单指令流多数据流(SIMD) 和直接利用并行算法的并行性如使用快速傅里叶变换(FFT) 等.
- (4)安全性能评估。从目前分析评估的结果来看, 采用经过安全修改的传统结构和能够进行安全性证明的结构安全性比较高,
- HAIFA 框架和宽管道MD 结构. 能够可证明安全的压缩函数的设计安全性比较高, 可证明不存在高概率的差分路径, 能够抵抗差分攻击.
- 一些新设计的结构由于缺少时间检验和理论的依据, 不能提供足够的安全性, 如采用流模式的算法都存在共同类似的弱点, 容易被敌手利用同种攻击手段攻破

4.3 信息鉴别
Message Authentication

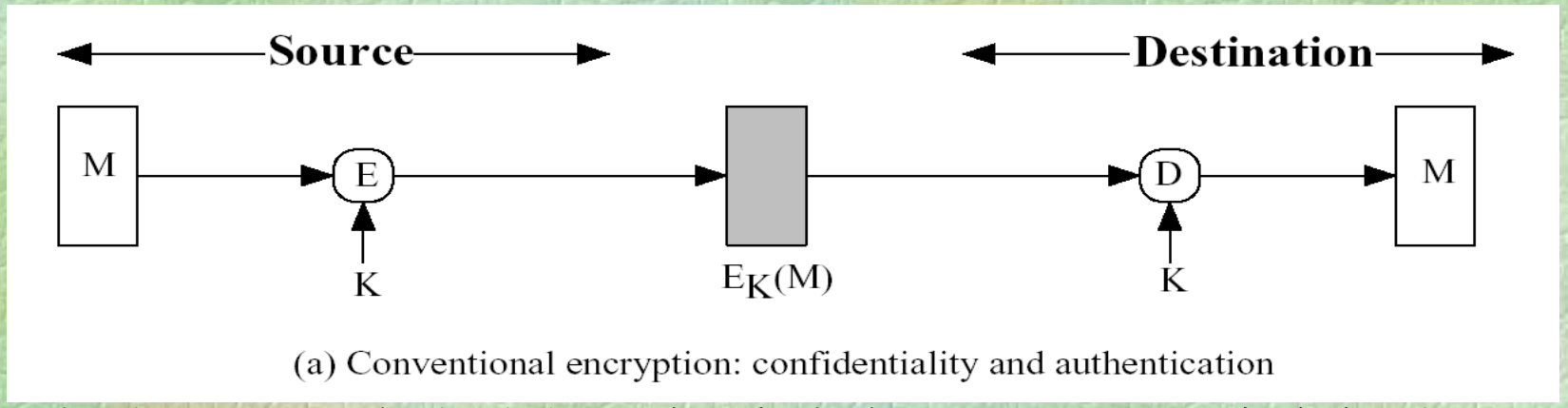
信息鉴别的概念

- 信息鉴别的目的是为信息提供一致性保证。也就是保护通信双方免受第三方的攻击。
- 信息鉴别需要能够鉴别：
  - 信息的来源
  - 信息在传输过程中没有被更改过

信息鉴别的方法

- 使用某种函数来产生鉴别符，用于鉴别一个信息的值
  - 信息加密：以整个信息的密文作为它的鉴别符
  - 信息鉴别码（MAC）：以一个信息的公共函数和用于产生一个定长值的密钥作为鉴别符
  - 散列函数：以散列值作为鉴别符

信息加密—常规加密



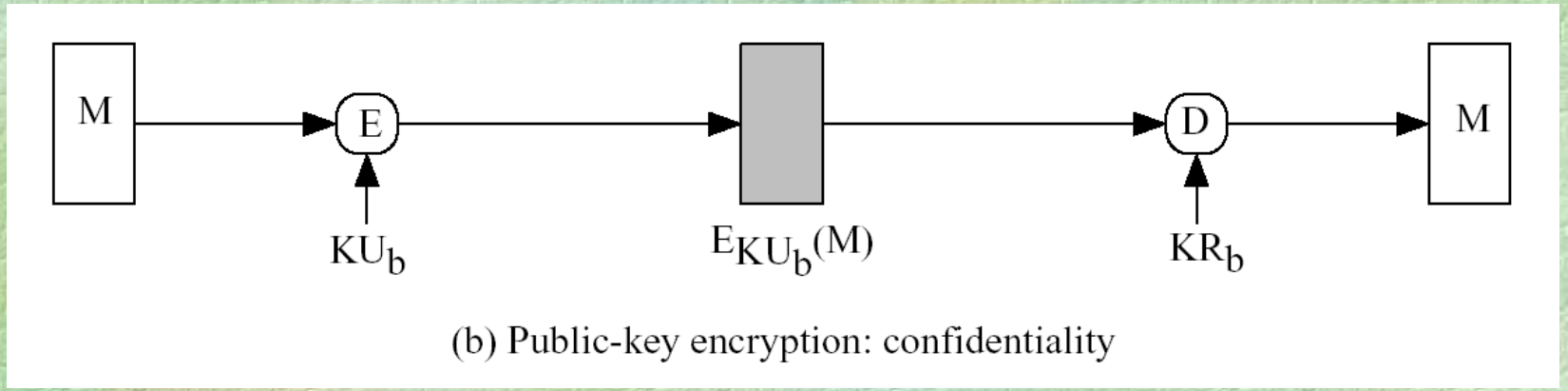
(a) Conventional encryption: confidentiality and authentication

- 仅有通讯双方有密钥，但接收方需要手段来确定解密得到的明文是否合法，是否来自发送方。
- 解决方案：明文具有易于识别但是不能复制且无需求助加密的某种结构。
  - 为信息附加检错码
  - 信息本身具有某种特定的结构（如TCP/IP协议的报文）

常规加密鉴别的特点

- A -> B： E_K [M]
  - 提供保密
    - 仅A和B共享K
  - 提供一定程度的鉴别
    - 仅来自A
    - 传输中不会被更改
    - 需要某种结构或冗余
  - 不提供签名
    - 接收人可以伪造信息
    - 发送人可以否认信息

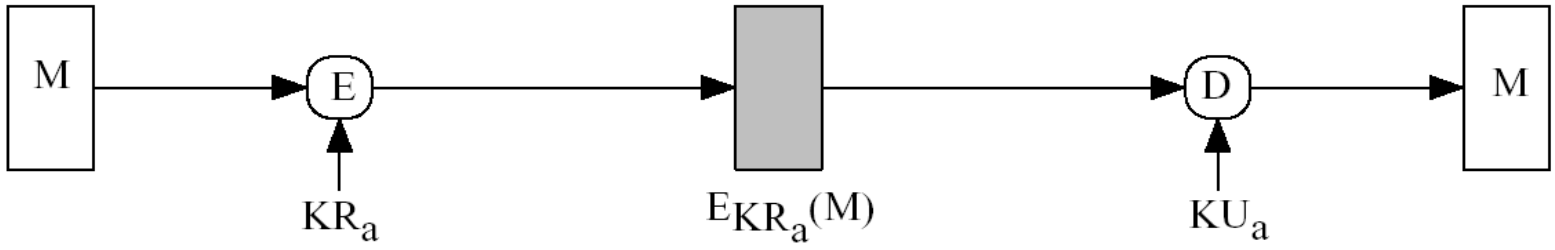
信息加密—公钥加密之一



(b) Public-key encryption: confidentiality

- **A -> B：E_{KU_b} [M]**
  - 提供保密
  - 不提供鉴别

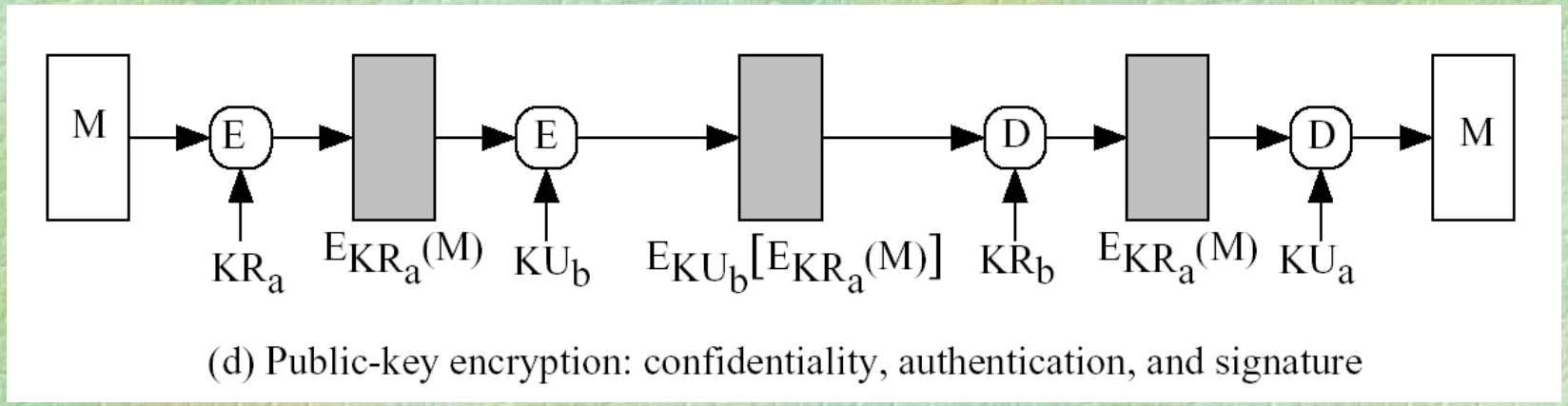
信息加密—公钥加密之二



(c) Public-key encryption: authentication and signature

- **A -> B：E_{KR_a} [M]**
  - **提供鉴别和签名**
    - **仅有A有KR_a可进行加密，任一方使用KU_a验证签名**
    - **需要某种结构或冗余**

信息加密—公钥加密之三



(d) Public-key encryption: confidentiality, authentication, and signature

- A -> B：$E_{KU_b}[E_{KR_a}(M)]$
  - KU_b提供保密，KR_a提供鉴别和签名

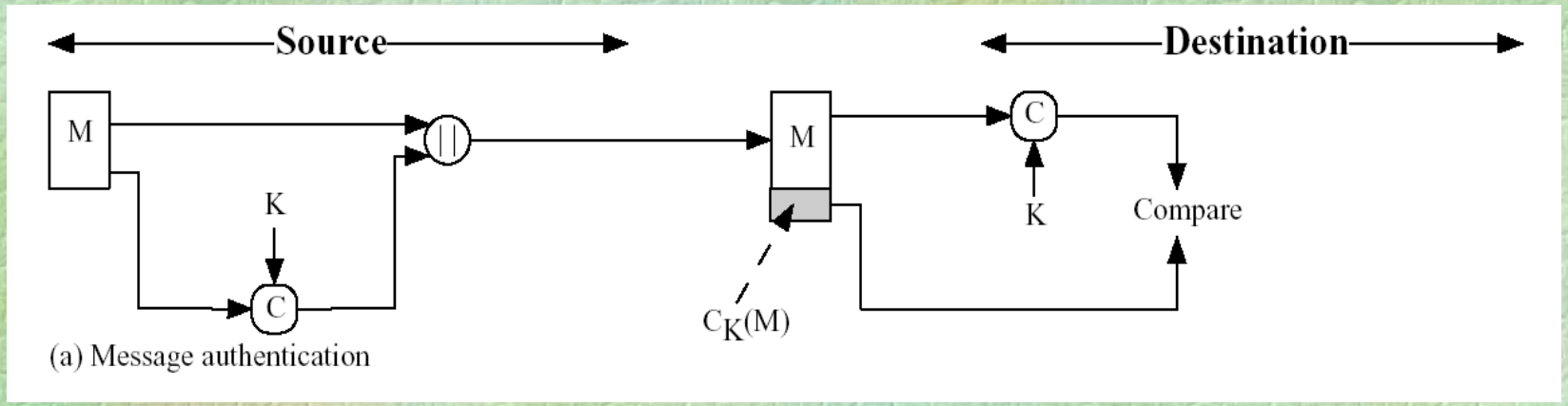
信息鉴别码（MAC）

- 也称为密码校验和，由如下形式的函数C生成

$$MAC = C_K(M)$$

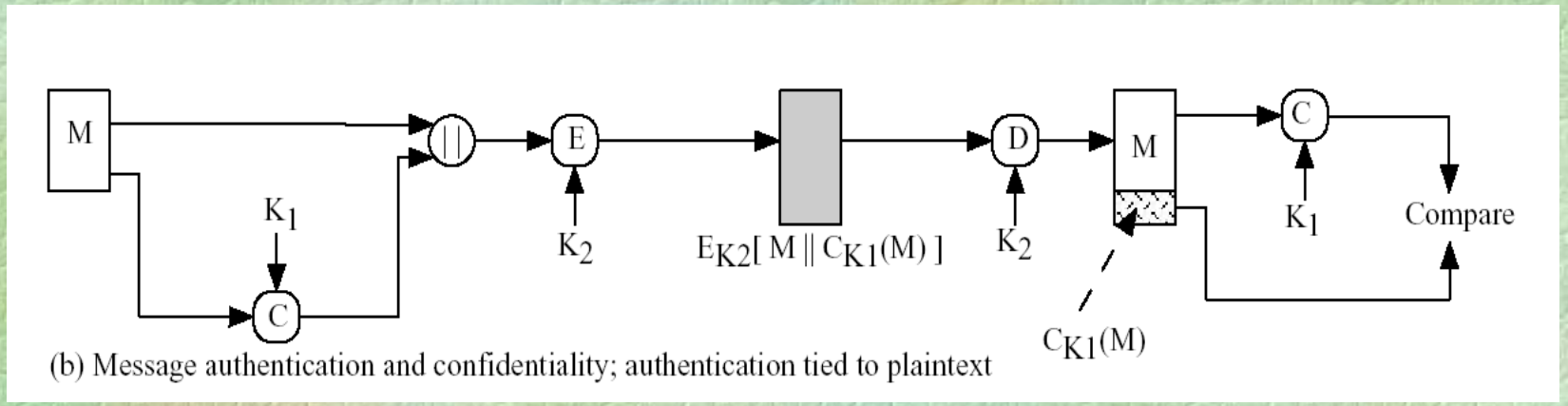
- 其中M是变长的报文，K是仅由收发双方共享的密钥，$C_K(M)$是定长的辨别符

信息鉴别码的基本用法一



- **A -> B： M || C_K (M)**
  - **提供鉴别　　K**

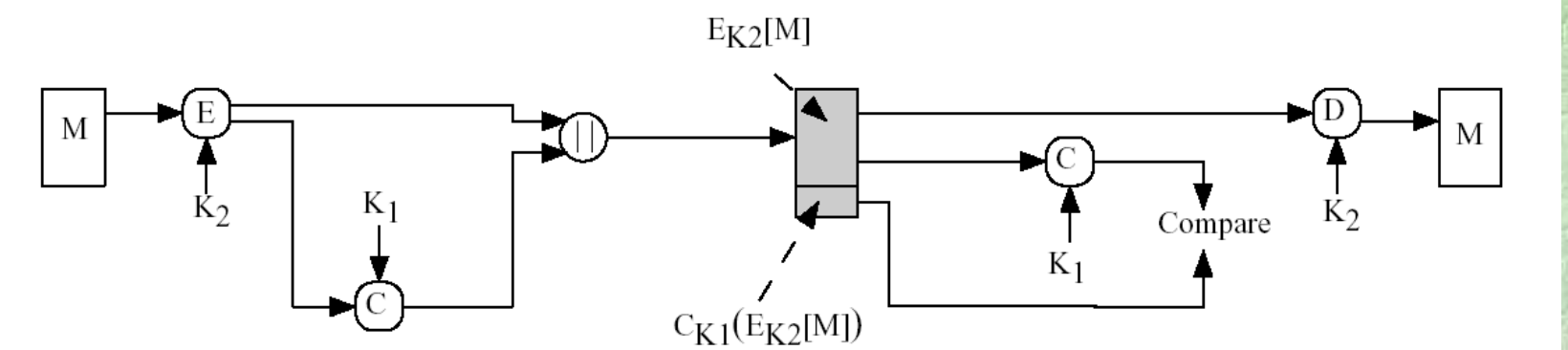
信息鉴别码的基本用法二



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

- **A -> B：E_{K2} [M || C_{K1} (M)]**
  - **提供鉴别　　K_1**
  - **提供保密　　K_2**

信息鉴别码的基本用法三



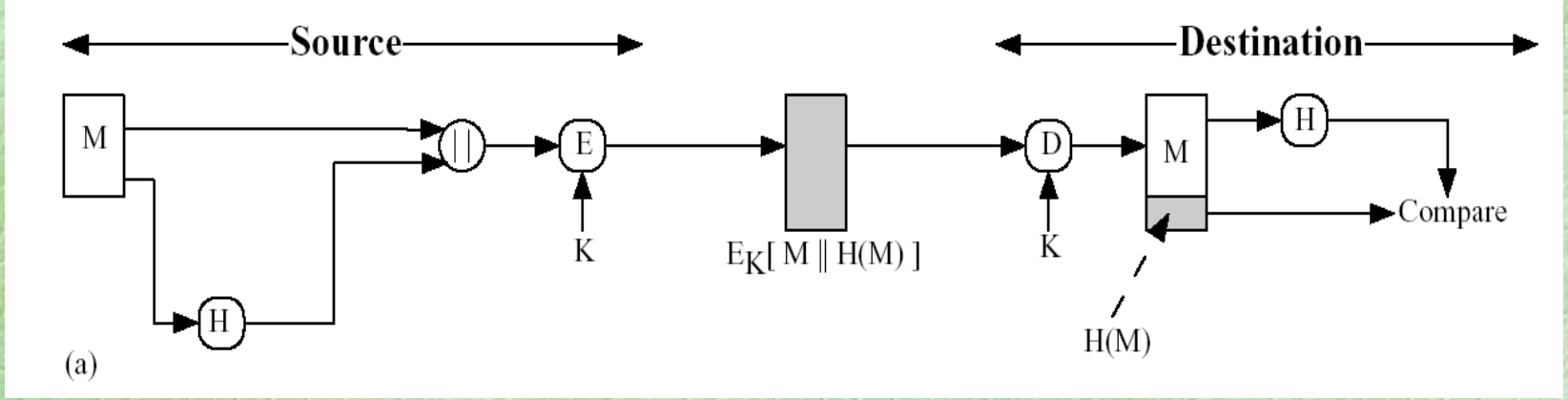
(c) Message authentication and confidentiality; authentication tied to ciphertext

- **A -> B：E_{K2} [M] || C_{K1} (E_{K2} [M])**
  - **提供鉴别　　K_1**
  - **提供保密　　K_2**

MAC函数应该具有的性质

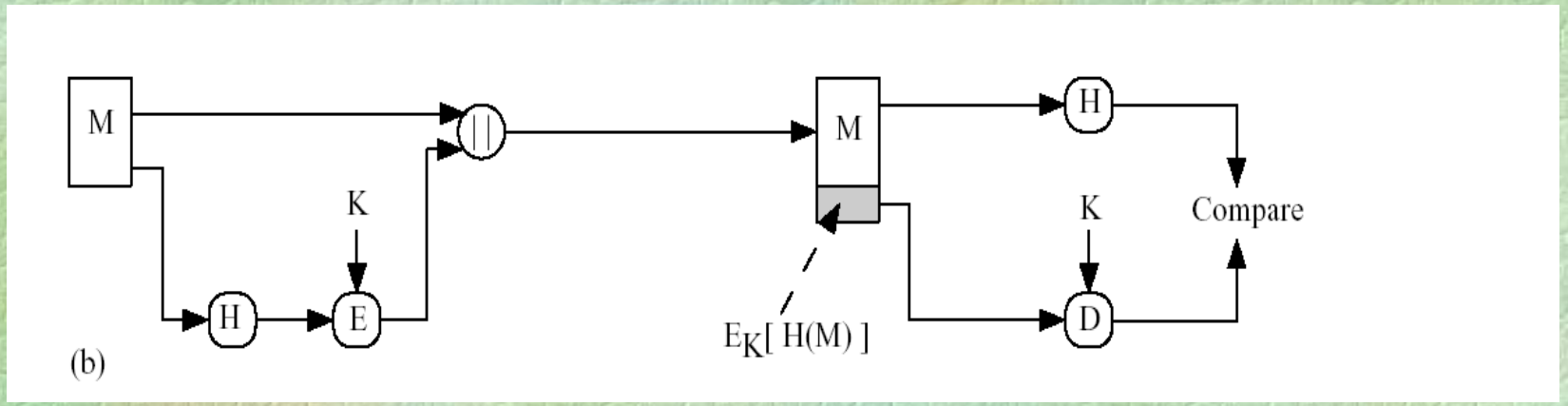
- 如果对手窃取到M和$C_K(M)$，试图生成一个信息M’，使$C_K(M') = C_K(M)$，这在计算上不可行。
- 应该能够均匀分布。对于随机选择的信息M和M’，$C_K(M') = C_K(M)$的概率为2^{-n}，其中n为MAC的长度。

使用散列函数的信息鉴别方法一



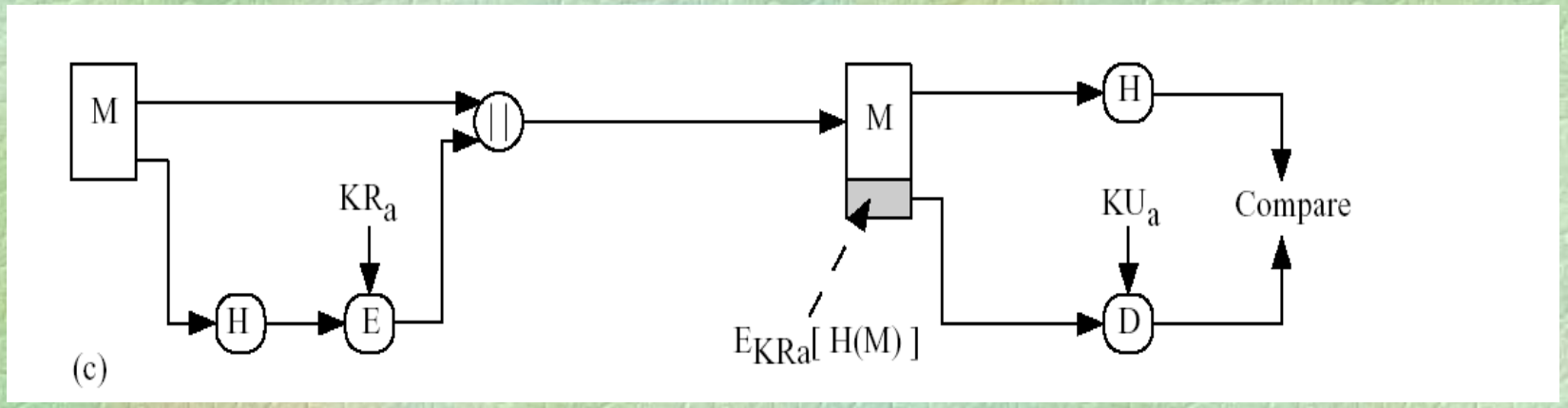
- 使用常规加密方法对附加散列码的信息进行加密
  - 提供保密　　K
  - 提供鉴别　　H(M)

使用散列函数的信息鉴别方法二



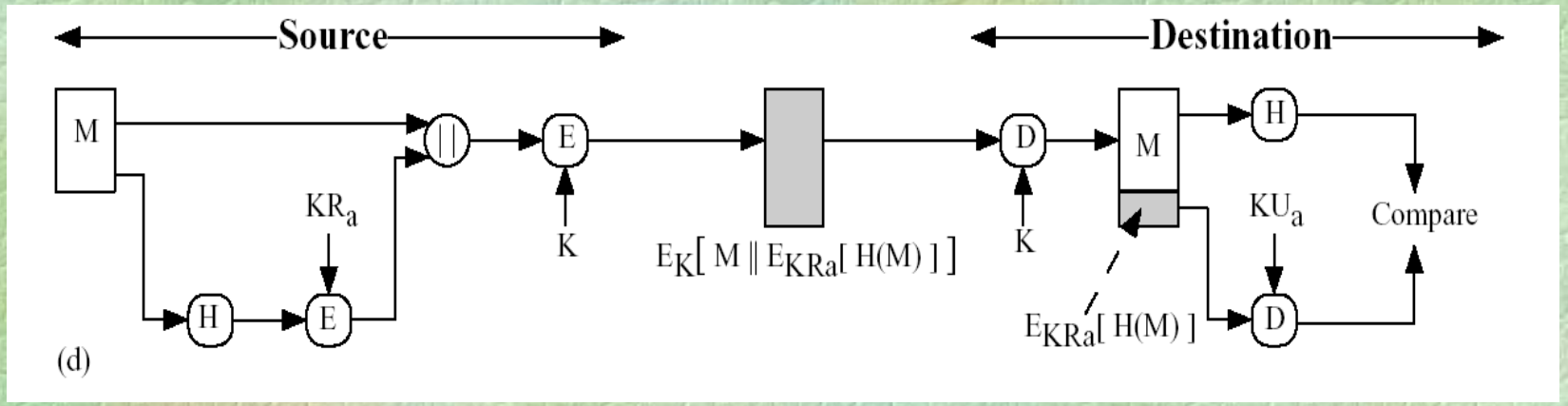
- 使用常规加密方法仅对散列码进行加密
  - 提供鉴别　　H(M)

使用散列函数的信息鉴别方法三



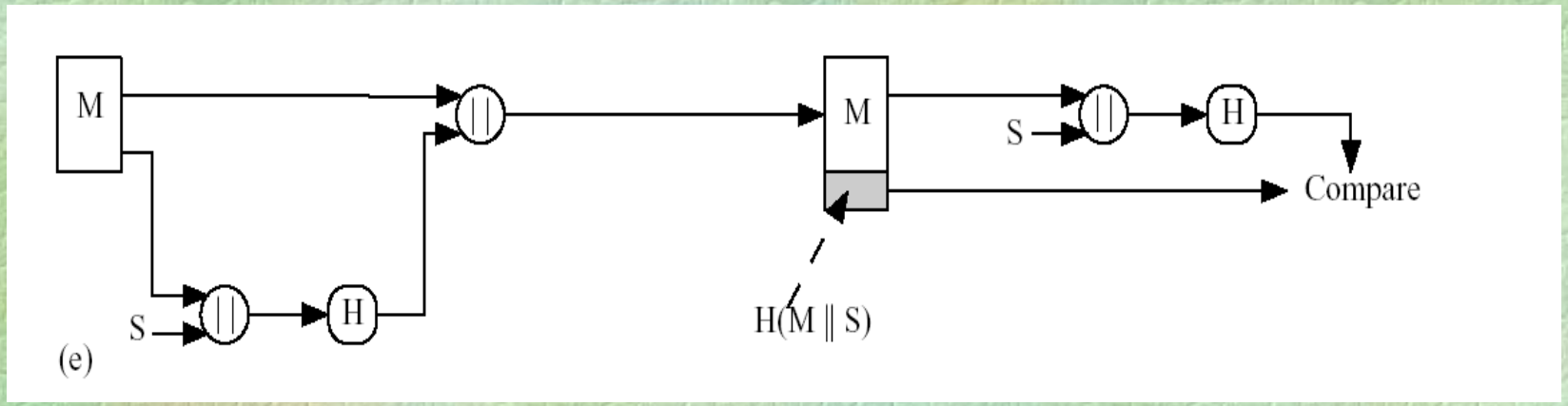
- 使用公开密钥加密方法及发方的私钥仅对散列码加密
  - 提供鉴别和数字签名
    - 加密保护的H(M)

使用散列函数的信息鉴别方法四



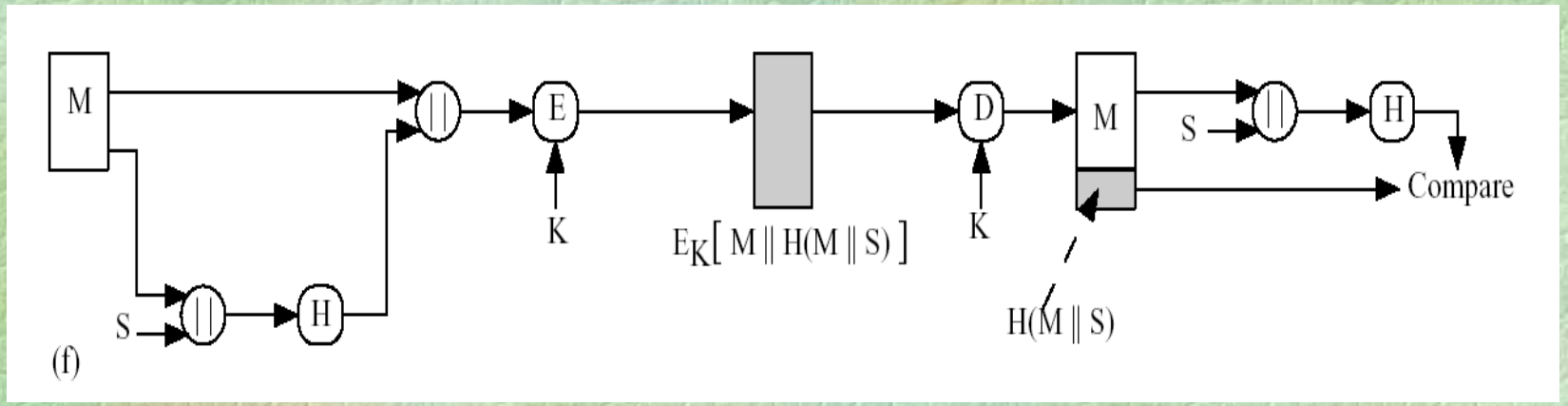
- 使用常规密钥对信息和已使用公开密钥加密的散列码一起加密
  - 提供鉴别　　和数字签名
  - 提供保密　　K

使用散列函数的信息鉴别方法五



- 使用通讯双方共享的秘密值S参与计算散列值，不加密报文
  - 提供鉴别　　S

使用散列函数的信息鉴别方法六



- 使用通讯双方共享的秘密值S参与计算散列值，加密报文
  - 提供鉴别　　S
  - 提供保密　　K

4.4 身份识别协议

- 随着Internet技术的不断推广与发展，当代人类的生活与网络已密不可分。
- 将遇到以下类似情况：使用磁卡在自动取款机中取钱；使用电话卡为长途电话付费；通过网络向计算机作远程登录。
- 采用电子的方式来证明自己的身份。
- 但如果仅采用口令这类验证的方法是不安全的。

- 在ATM上取款时，若仅用卡上的磁性条和自己的口令，则任何监视通讯线路的人都能轻而易举地截获磁性条上的信息，从而非法访问银行帐户。
- 这样的解决方案对于发送消息的人而言是脆弱的，它容易遭到第三者的攻击。
- 为了解决以上的问题，自然想到能否在接收方收到信息时，对发送信息的人的身份加以验证，即通过身份识别来防止第三者的冒名顶替。

- 好的身份识别方案应能够实现以下的功能(用Sender代表信息发出者，Receiver代表信息接收人):
- 当Sender向Receiver说明自己的身份时，任何截获到Sender身份信息的人，以后将永远无法冒充Sender;
- 即使是在Sender证实身份后，接收者Receiver本人也无法冒充Sender.

- 身份验证方案的基本模型是非常简单的：首先Receiver选取一个随机串，将它发给Sender，
- Sender在收到这个随机串后，使用和Receiver共享的秘密密钥K的加密函数E_k，计算出$Y=E_k(x)$，并将它发还Receiver，
- Receiver同时也计算一个$Y'=E_k(x)$。
- 如果结果与Sender传来的Y相同，那么Sender的身份得到了验证，否则Sender就是不合法的。
- 上述模型是由询问和应答两部分构成的。
- 所有的身份识别方案都是“询问—应答”的。但是上述方案并不是完美的，
- 使用的密钥及加密函数是被Sender和Receiver共同享有的，这样对于Sender本人而言还是不够安全，
- 如果密钥一旦被Receiver泄露,那么其它人就可以轻而易举地冒充Sender了。
- 因此，一个有用(或完美)的身份识别方案应该不需要共享密钥。

- **4.4.1 Schnorr身份识别方案**
- **Schnorr方案的实行需要一个可信的中介机构，这个机构将规定方案的各项参数，以及向Sender签发证书。为叙述方便，以下用TA来代表这样一个机构。**
- **TA要完成的工作如下：**
- **(1)选择一个大的素数p($p \geq 2^{512}$)，使得对于Zp^*的离散对数问题是难解的。**
- **(2)寻找p-1的一个大的素因子q($q \geq 2^{140}$)。**
- **(3)在Zp^*中寻找阶为q的元素 α。**
- **(4)选择安全参数t，使$q \geq 2^t$。**
- **(5)选择一个散列函数，所有的信息在签名前都要进行散列，当然散列函数本身要求是安全的。**

- 建立一个安全的数字签名方案，该方案由一个秘密的签名算法SigTA和公开的验证算法VerTA组成。
- TA所建立的方案参数除了签名算法SigTA以外，其他都是公开的。
- TA的工作完成后，Sender和Receiver就可以开始进行身份验证了。
- 首先Sender将向TA申请一个证书，步骤如下：
- (1)TA用Sender申请证书时提供的身份信息，给Sender指定一个ID(当然对于不同的申请者，ID是不同的)。
- (2)Sender选择一个秘密的随机数a(0≤a≤q-1），计算$v=\alpha^{-a} \bmod p$，并送给TA。
- (3)TA根据签名算法SigTA， 产生签名S=SigTA(ID，v)，并发给Sender一个证书C(Sender)=(ID,v,S)。

- Sender和Receiver通过以下询问—应答完成身份识别：
- (1)Sender选择随机数k(0≤k≤q-1)，计算出$\Gamma=\alpha^k \bmod p$，并将Γ和自己的证书C(Sender)=(ID，v,S)送给Receiver。
- (2)Receiver通过验证算法VerTA来验证Sender的证书C(Sender)中TA的签名S，如果签名是合法的，选择随机数r($1\le r\le 2^t$)，并送Sender。
- (3)Sender 计 算 Y=k+armod q， 并 将 Y 送 给 Receiver。
- (4)Receiver验证Γ是否等于$\alpha^Y v^r \bmod p$。

- 在Schnorr方案中，Receiver对Sender身份的验证分为两部分。
- 首先，VerTA验证算法将检查Sender证书的有效性，从而确定Sender拥有合法的证书。
- 假设第三者(以后称为Cheater)通过伪造证书C’(Sender)=(ID,v’,S’)来冒充Sender。TA的签名算法SigTA是保密的，
- Cheater无法获得关于(ID,v’)的正确签名S’，
- 从而Receiver将确定Cheater的C’(Sender)是伪造的。

- 如果签名算法因为某种原因被泄露，Cheater将可以得到一个合法的证书C(Sender)，这时Receiver就不能通过VerTA来证明Cheater非法，这时Schnorr方案的第二部分将发挥作用。
- 在第二部分里，因为涉及到一个由Sender自己指定的参数a，而且在身份识别过程中，a的值不会暴露，因此Cheater将不能冒充Sender，除非他也知道a的值(从v中计算a将涉及到解离散对数问题，而这是难解的)。

- 说明Schnorr方案的完备性，即Sender可以通过该方案向Receiver证明自己的身份

$\alpha^{Y}v^{r}\equiv\alpha^{k+ar}v^{r}(\mathrm{mod}p)\equiv\alpha^{k+ar}\alpha^{-ar}(\mathrm{mod}p)\equiv\alpha^{k}(\mathrm{mod}\ p)\equiv\Gamma$

- 另一方面，也可以通过以下定理来说明Schnorr方案的可靠性：
- 定理4.3：假设Cheater知道一个值Γ，对这个Γ在通过Schnorr方案验证身份时，成功地模仿Sender的概率$\varepsilon\geq(2^{t}-1)^{-1}$，那么Cheater能够在多项式时间内计算出a

- 这个定理说明如果一个人能够以不可忽略的概率成功执行Schnorr身份识别方案，那么，他一定知道Sender的秘密参数a。
- 反过来说，即一个人如果不知道秘密参数a，那么他能成功地执行Schnorr方案身份识别的概率是可以忽略的(TA设定的参数t保证了这个概率可忽略，一般情况下t取40是合理且有效的)。这也就是Schnorr方案的可靠性。

- 尽管Schnorr方案可靠，但仍不能说它是安全的。
- 所谓安全是指Cheater通过参与执行该方案多项式次，且经过多项式大量的计算，仍无法确定关于参数a的任何信息，也就是说Cheater通过参与执行该方案多项式次仍无法获得a的任何信息，
- 目前还无法证明Schnorr方案满足这个要求
- 但同时也没有证据表明该方案是脆弱的。
- 因此，在大多数场合，Schnorr方案仍不失为一个优良的身份识别方案。

4.4.2 Okamoto身份识别方案

- Okamoto对Schnorr方案进行了改进，保证在假设Z_p中一个特定的离散对数问题是难解的条件下，改进方案是安全的。
- TA在Okamoto方案中除了要确定p,q,t和散列函数、签名方案等参数外，还要选择两个阶为q的元素α_1， $\alpha_2 \in Zp^*$，
- Sender在计算v时将选择两个秘密参数a_1和$a_2(0 \le a_1, a_2 \le q-1)$，且$v=\alpha_1^{-a1}\alpha_2^{-a2}(\text{mod } p)$。
- 身份识别时，随机数k的选择也由一个变为两个：k_1和$k_2(1 \le k_1, k_2 \le q-1)$，且$\Gamma=\alpha_1^{k1}\alpha_2^{k2}(\text{mod } p)$。
- 在Receiver选定随机数r后，Sender将计算两个Y：$Y_1=(k_1+a_1r)\text{mod } q$，$Y_2=(k_2+a_2r)\text{mod } q$送给Receiver，Receiver最后的验证相应地变为：$\Gamma=\alpha_1^{Y1}\alpha_2^{Y2}v^r(\text{mod } p)$。
- Okamoto将一个秘密参数a扩展成两个a_1和a_2，这样的改进看似不大，但却从根本上解决了安全性的证明。

- 定理：假设Cheater知道一个值Γ，对这个Γ成功模仿Sender的概率$\varepsilon \geq (2^t - 1)^{-1}$，那么Cheater和Sender合伙以概率1－1/q能够在多项式时间计算$\log_{a1} a_2$
- Sender选择了两个秘密参数而不是一个，所有可能的秘密参数有序对构成一个集合，这个集合中有q个对，它与Sender的(a_1, a_2)是“等价的”。
- 知道这个集合中两个不同的对就提供了计算离散对数c的有效算法。
- Sender知道其中一个对，如果Cheater能够冒充Sender，那么就能计算出集合中的一个对，并且与Sender的对不同的概率几乎为1。
- 因而Sender与Cheater一起能够找到两个不同的对，从而计算出c，也就产生假设矛盾。
- 尽管Okamoto方案是安全的，但与Schnorr方案相比，执行时间几乎翻倍。

- 作业:P138 5,6