网安期末复习笔记

“**三网合一**”的三网指的是：**电信网**、**计算机网络**和**有线电视网。**

电信网（蜂窝移动通信系统）→计算机网络（分布式）：

- 50-60年代：计算机网络雏形，主机-通信线路-终端。
- 1969年：**ARPANET**（~交通网，分布式），**分组交换**技术，形成**资源子网和通信子网**的网络结构，路由器使用**存储转发**方式。
- 1983年：**Internet**，采用标准TCP/IP协议，实现异质网络互联。

诈骗地下黑色产业链的分工：

1. **漏洞挖掘**：发现漏洞提供给公司安全部门
2. **代码编写**：根据漏洞编写病毒程序
3. **信息窃取**：用别人写好的病毒程序去偷信息（蛮力破解的撞库、中间人攻击、拒绝服务攻击、病毒等）
4. **信息贩卖**：把偷来的信息拿去网上卖
5. **窃取诈骗**：社会工程学范畴，用买来的信息去诈骗

一、必会密码学算法

（手动计算）

二、必会密码学基础

1、密码学基本概念

密码学=**密码编码学**（如何加密）+**密码分析学**（如何破译）

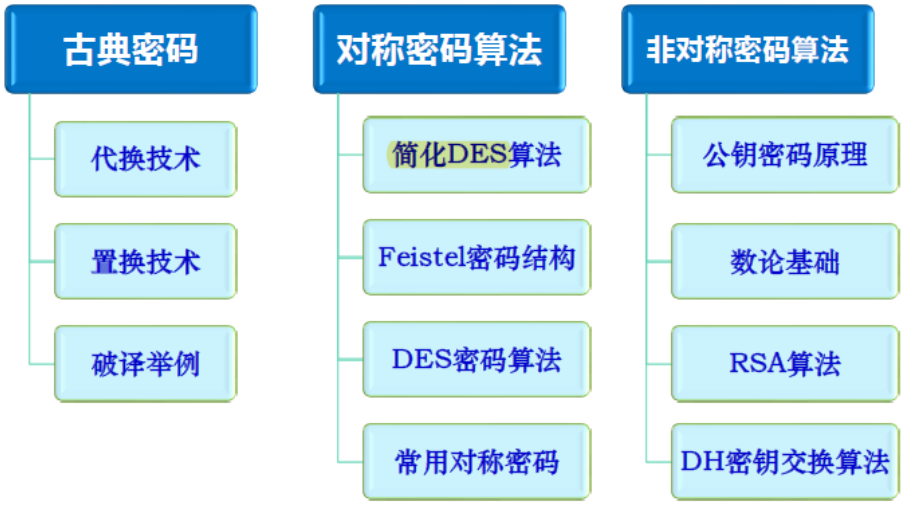
- 密码编码学的特征：**【！要背下来】**
    1. **转换明文为密文的运算类型**（①代换/置换，②数学函数）
        - 如果用了代换/置换，那么就是传统加密即（古典/近代）对称密码；
        - 如果用了数学函数，那么就是公钥密码
    2. **所用的密钥数**（Sender & Receiver用的密钥相同→传统；不同→现代）
    3. **处理明文的方法**（分组密码Block cipher、流密码Stream cipher）
- 加密机制追求**计算安全**（破译代价>数据价值，破译时间>密文生命期）

加密形式：

- 传统加密（=对称=单钥）：只有一个密钥，同时用于加解密
- 现代加密（=非对称=公钥）：多个密钥，分别用于加密＆解密

2、古典、近代（对称）、现代（非对称）密码的技术特点

密码学基础



古典：只有对称，代换置换

- 基本技术手段：代换（凯撒、单表代换、Playfair及之后都是多表、Hill、Vigenere、Verman）、置换（栅栏）
- 安全性：在于**算法**，因此需要对算法（Encode、Decode）保密，否则就失效了
- 破译：穷举法、频率分析法

近代（对称）：代换置换

- 新发明：计算机、有线电报（现代编码学）、无线电报（现代密码分析学）
- 例子：DES、IDEA、RC系列、AES、CAST-128、Blowfish
- 对称密码的安全性：**密钥**，要使用秘密信道分配（又叫对称密钥算法）
- 现在使用的**对称分组密码**通常用**Feistel分组密码**结构（见下）

现代（非对称）：数学函数

- 特点：不需要对算法保密；公钥密码是基于**数学函数**而不是代换置换。
- 例子
    - RSA公钥算法：攻击有蛮力攻击、数学攻击、计时攻击；抵抗穷举的办法是使用大密钥，但是速度会变得很慢；
    - DH密钥交换算法：只能用于密钥交换，原理是离散对数计算很困难；
    - DSA（数字签名算法）：只能用于数字签名，有数字签名标准（DSS）。
- 安全性：在于密钥（Ke、Kd）保密，算法可以公开
- 公钥密码算法：
    - **加密解密**：用别人公开的公钥加密，保证只有他可以解开，完成加密；
    - **数字签名**：用自己的私钥加密，使别人通过公开的公钥解开从而验证身份；
    - **密钥交换**：双方协商会话密钥，用于对称密钥数据加密（e.g. SET协议中customer用发卡行公钥对session key加密，装进电子信封）。

公钥密码算法	加密/解密	数字签名	密钥交换
RSA	Y	Y	Y
Diffie-Hellman	N	N	Y
DSA	N	Y	N

对称 vs 非对称密码

加密方法的安全性：依赖于密钥长度和破译计算量，不能简单说哪个更安全；

应用现状：都在用，加密还是用传统密码，公钥密码计算量比较大所以仅限于密钥管理和数字签名；

实现密钥分配的难度：不一定，传统密码和密钥分配中心握手很复杂，但使用公钥密码要通过中心代理等，也不简单；

3、流密码和分组密码、混淆与扩散、Feistel密码结构

流密码与分组密码

流密码：每次处理1 bit / 1 byte的明文（Caesar、密钥词、Vigenere类Caesar多表、Vernam**(?)**、转轮机如Enigma）

- 注意S-DES输入是8位，但不是流密码，因为流密码强调"原子"，如一个二进制位或者一个字符，而分组密码则是组内互相影响。

分组密码：每次处理一个明文组，得到**等长**的密文组（Playfair一次两个字母、Hill一次n个词和矩阵做乘法、置换举例用的按行写入按列读出、S-DES一次8位、AES、Blowfish、RC5、RSA）

混淆与扩散

扩散：使**明文的统计特征消散在密文**中（密文体现不出），让1个明文和n个密文相关联，n尽可能大

混淆：使**密文和加密密钥间的统计关系更复杂**，密钥难以从密文中反推出来（做出更复杂的密钥）

——**现代分组密码设计的里程碑：混淆＆扩散**（密文中看不出明文的统计特征 / 密钥）

Feistel密码结构

Feistel建议：

- 用乘积密码的概念替换简单代换（多轮迭代，最后看起来像一个代换，但已经很复杂了）
- 交替使用代换和置换

Feistel密码结构：（例如Lucifer算法）（DES如果去掉初始置换 IP 和末尾置换 IP^{-1} 就是典型的Feistel密码结构）

- 输入明文组长度为偶数，每次迭代结构相同，都是分左右两半，然后得到下一次迭代的输入。

  代换：$R_{i+1} <= L_i \oplus F(R_i, K_i),$
  置换：$L_{i+1} <= R_i.$

- 每次迭代中使用的轮函数 F 相同，但子密钥 K_i 是由整个密钥 K 推导出来的，互不相同。
- 重要参数有分组长度、密钥长度、迭代轮数、产生子密钥的算法、轮函数。

——和古典的Feistel密码结构**不同**的有：**Blowfish**（每轮都是左右两半同时运算，而不是只算左边然后下一轮置换）、**RC5**、**AES**（每轮都用了代换和置换并行处理，而不是开头代换结尾置换）

4、密钥分配的三种情况、密钥分配中心KDC模式

三种情况：**传统**的对称密码分配、非对称密码中的**公钥**分配、**公钥**密码用于**传统**密码体制的密钥分配。

传统的对称密码分配

适用于链路加密：A亲自交给B，第三方选择后亲自交给A和B，需要人工传送密钥；

适用于端到端加密（用户到用户）：使用密钥分配中心（KDC）

KDC：见下文

非对称密码中的公钥分配

1. 公开发布：直接公布，【缺点】任何人都能伪造。
2. 公开可访问目录：一个可信的组织维护一个公钥目录，目录项是<姓名，公钥>。【缺点】是目录管理员的私钥必须保证安全，否则目录就挂了。
3. 公钥授权：A和B通过管理员通信，消息用管理员的公钥加密，管理员用自己的私钥解密，分发B和A的公钥（很像KDC，但是A从管理员获得B公钥之后发消息给B，B还要从管理员那里获得A的公钥，更麻烦一点）；【缺点】是公钥管理员成为系统瓶颈，效率比较低。
4. 公钥证书：证书管理员产生证书（有公钥和其他信息），发给有私钥的通信方。通信方之间传递的是证书，可以验证其真实性（由证书管理员发出）＆当前性，从而可以读出公钥。

利用公钥分配传统密码的密钥

1. 简单的密钥分配：A把自己的公钥和消息发给B，B用A的公钥加密对称密钥Ks（封入电子信封）发给A，A用私钥解密了获得对称密钥Ks。【缺点】容易受到主动攻击，A未经身份认证，可能是别人假冒的。
2. 具有保密性（加密）和真实性（认证）的密钥分配：假设AB已经交换了公钥，
    1. A用B的公钥加密一条有A标识和临时交互号N1的消息给B；
    2. B用自己的私钥解密获得A标识和N1，再用A的公钥加密N1和临时交互号N2给A；（A收到后从N1可以判断对方是B，有B私钥）
    3. A用B的公钥对N2加密，返回给B；（B收到后从N2可以判断对方是A，有A私钥）

       ====== 双方完成互相的身份认证 ======
    4. A选择密钥Ks，用自己的私钥KRa加密、B的公钥KUb加密发给B；
    5. B用自己的私钥KRb解密、A的公钥KUa解密得到对称密钥Ks。
3. 混合方法：也需要KDC，通过主密钥实现会话密钥的分配。

KDC模式

假设：只有A和KDC知道主密钥Ka，只有B和KDC知道主密钥Kb（用户和KDC的通信是秘密的）

步骤：

1. A向KDC发送请求，发过去A和B的标识和临时交互号N1，希望获得一个会话密钥；
2. KDC回复给A一个用Ka加密的响应，包括了会话密钥Ks（一次性的）、原始请求信息（含N1）、一个用Kb加密的消息（含Ks和A的标识符IDa）；
3. A存下Ks，把用Kb加密的消息发给B。

——网络规模很大的时候，使用层次式的KDC，降低主密钥分配的代价，而且本地KDC出错不会影响区域外。

三、必会认证技术（消息认证、身份认证）

1、消息认证基本概念，MAC 码和 Hash 码的工作原理

网络环境中的攻击及其应对方案：

- **消息保密性**范畴：泄密、传输分析
- **消息认证**范畴：伪装、内容修改、顺序修改、计时修改
- **数字签名**：是一种对付发送方否认的认证技术，也可以用于接收方否认（还需相关协议）

消息认证：确保①发送方真实，②消息真实，方法是认证函数产生认证符，接收方计算并比较。

- 【注意】消息认证可以保证通信双方不受第三方攻击，但是不能处理**通信双方自身发生的攻击**（发送方否认、伪造对方消息等）
- => 在**收发双方不能完全信任**的时候要其他办法来解决，用数字签名是最好的（见后）。

认证函数分成三类：

- 消息加密：从整条消息生成密文
- 消息认证码MAC：
- Hash：将任意长的消息映射为定长的hash值

消息加密

对称加密：（保密+认证）A对消息M附加一个校验和FCS，对它们一起加密，B收到后解密并计算FCS，比对。【注意】不能先加密再算FCS，否则攻击者可以构造有正确FCS的消息，产生混淆。【例子】TCP协议。

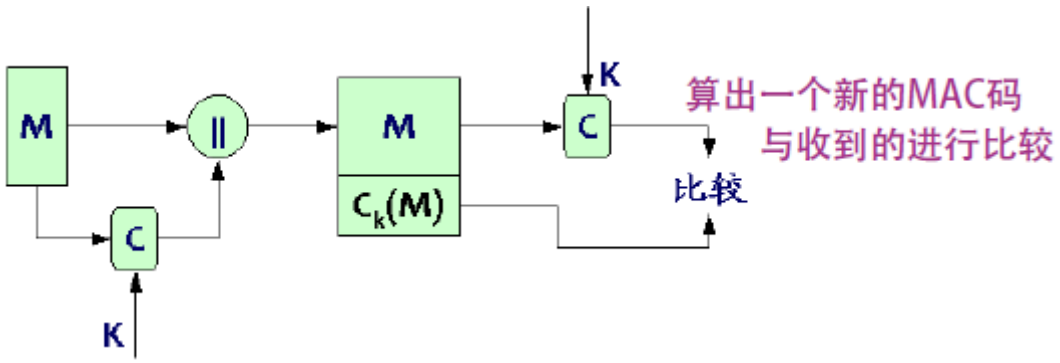
公钥加密：

- （保密）单纯用公钥只能提供保密性，不能提供认证；
- （认证+签名）A用私钥KRa加密，B用A的公钥KUa解密；
- （保密+认证+签名）A先签名再用KUb加密，B先KRb解密再KUa验证是A。

消息认证码 MAC

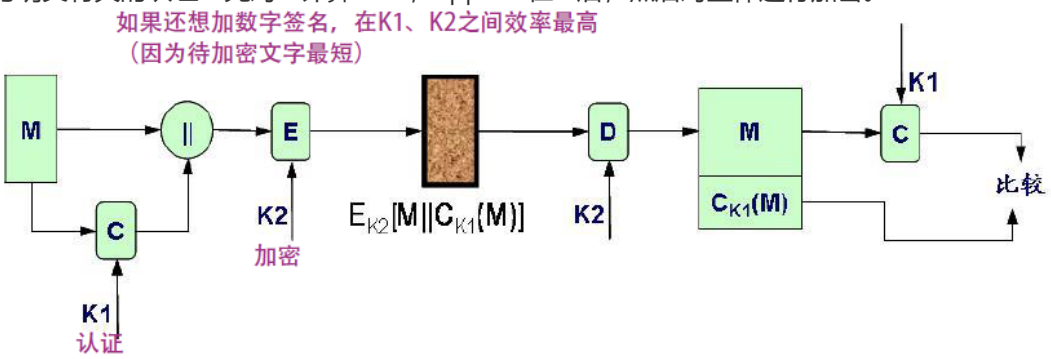
MAC算法（认证）：MAC=$C_k(M)$，产生固定长度的短数据块并append在M后，C是和密钥K有关的函数，收发双方**共享密钥K**【因此无法数字签名】。

——与加密类似，但**不要求可逆性**（加密则必须可逆）。

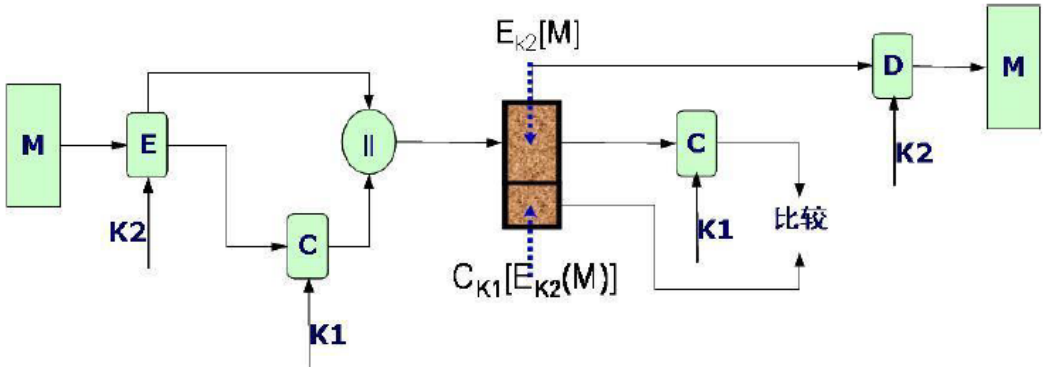


认证+保密：

- 与明文有关的认证：先对M计算MAC，append在M后，然后对整体进行加密。



- 与密文有关的认证：先加密，对密文计算MAC并append在其后。



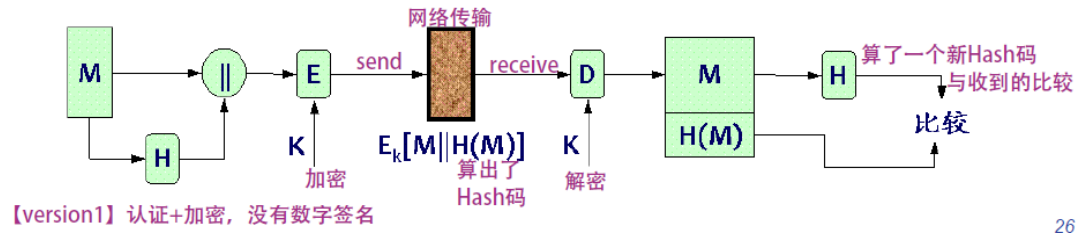
Hash 函数

是消息认证码的一种变形，输入可变长消息M，输出固定大小hash码H(M)。Hash码是所有消息位的函数，也称为消息摘要（Message Digest，MD）。

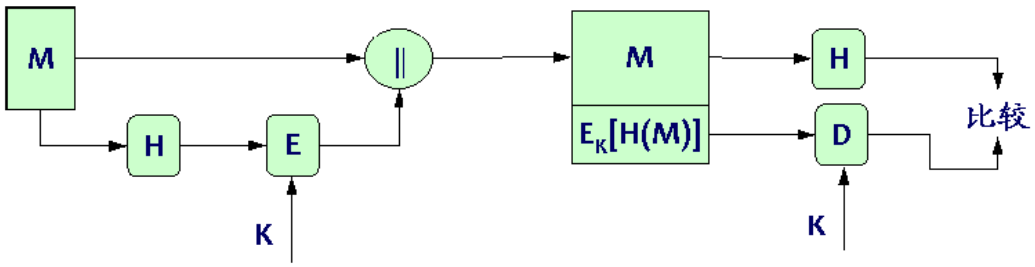
不同：**hash不用密钥。**

使用场景

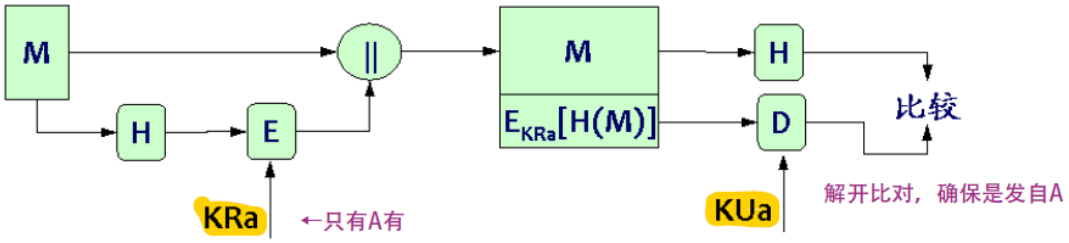
- 认证+加密：M拼接上Hash(M)，用对称密码K对它们加密——对称密钥只有AB共享，完成对A的确认，Hash码确保了消息未经篡改。



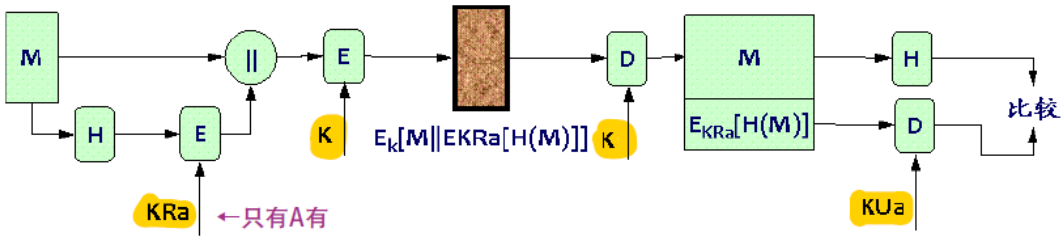
- 认证：用对称密码K仅对Hash(M)加密，M明文传输；（Hash函数和加密函数合成就是MAC，C_k）



- 认证+数字签名：用私钥KRa对Hash(M)加密，即把上面的对称密码K换成A的私钥，可以提供数字签名。解开时要用A的公钥，验证签名。



- 认证+加密+数字签名：先私钥KRa对Hash(M)加密，与M拼接后再用对称密码加密（缝合之前两种）。



- 认证+加密（加盐Hash）：在之前那种情况里，额外假定双方都知道一个秘密值S，可以把计算Hash(M)变为计算Hash(M||S)，即拼上S再算，更安全；
- 认证+加密+数字签名：在之前那种情况里，同样换成M||S的版本。

数字签名

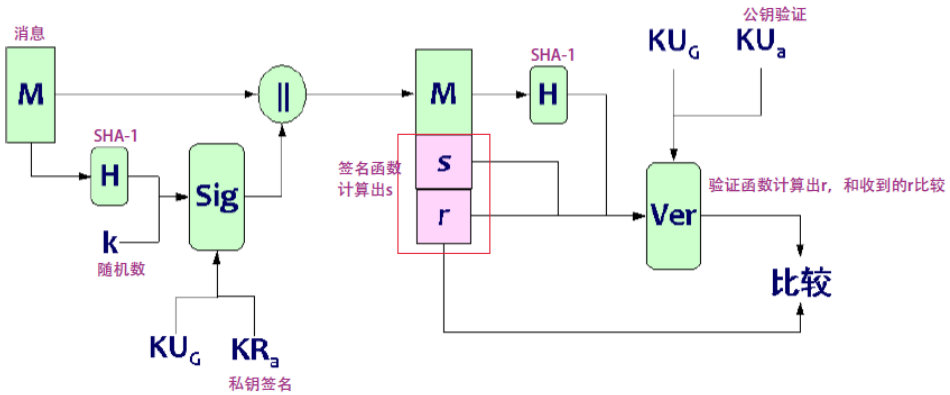
分类：直接数字签名、仲裁数字签名。

- 直接：只涉及双方，A发给B，先KRa签名，再KUb加密；【缺点】发送方私钥要安全；
- 仲裁：仲裁者检查消息、签名并加上日期和通过检验信息转发。
    - 数字签名标准DSS（SHA-1算法）给出了新的数字签名方法DSA。
    - （怪复杂的，放几张图，有空看看）

数字签名标准DSS

- DSS使用hash函数，它产生的hash码和随机数k作为数字签名函数Sig的输入，签名函数依赖于发送方的私钥KRa和一组通信伙伴共同拥有的参数构成的全局公钥KUG
- 签名函数保证只有拥有私钥的发送方才能产生有效签名
- 签名由两部分构成：s和r
- 接收方对接收到的消息产生hash码，这个hash码和签名一起作为验证函数Ver的输入，验证函数依赖于全局公钥KUG和发送方公钥KUa；若验证函数的输出等于签名中的r，则签名有效

8



参数说明

- 全局公钥（p,q,g）
  - p: 为L位长的素数。其中，L为512～1024之间且是64倍数的数。
  - q: 是160位长的素数，且为p-1的因子。
  - g: $g=h^{(p-1)/q} \bmod p$。
    其中，h是满足$1<h<p-1$且$h^{(p-1)/q} \bmod p$大于1的整数。
- 用户私钥x: x为在$0<x<q-1$内的随机数
- 用户公钥y: $y=g^x \bmod p$
- 用户每个消息用的秘密随机数k, $0<k<q$ 每次签名产生一个单独的k

参数p、q、g是公开的；x为私钥，y为公钥；
对于每一次签名都应该产生一次k；x和k用于数字签名，必须保密；

签名过程

用户随机选取k，计算：

- $r=(g^k \bmod p) \bmod q$
- $s=[k^{-1}(H(M)+xr)] \bmod q$

(r，s)即为消息M的数字签名

验证过程

接收者收到M,r,s后，首先验证0<r<q, 0<s<q，如通过则计算：

- $w=(s)^{-1} \bmod q$
- $u_1=[Mw] \bmod q$
- $u_2=[rw] \bmod q$
- $v=[(g^{u_1}y^{u_2}) \bmod p] \bmod q$

如果v=r，则确认签名正确

2、安全Hash函数的一般结构、MD5 / SHA-1 / RIPEMD160的基本步骤

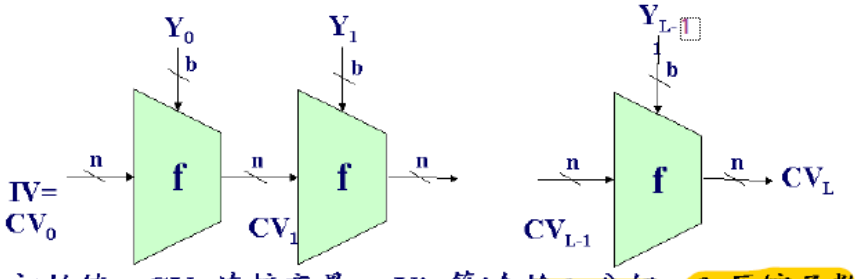
安全Hash函数的一般结构（又称为**迭代hash函数**）：MD5 / SHA-1 / RIPEMD160都是这种结构

- 输入消息分为L个固定长度（=b位，最后一组可以有padding）的分组，迭代L轮：

$$CV_0 = IV = \text{初始值},$$
$$CV_i = f(CV_{i-1}, Y_{i-1}), 1 \le i \le L, \text{即第}i\text{个输入分组}$$
$$H(M) = CV_L$$

Hash函数公式

- CV0=IV=初始n位值
- CVi=f(CVi-1,Yi-1)　1<=i<=L
- H(M)=CVL (其中输入M由Y0，Y1，...，Yl-1组成)



IV=初始值；CV=连接变量；Yi=第i个输入分组；f=压缩函数
L=输入分组数；n=hash码的长度；b=输入分组的长度

- 重复使用了压缩函数f，输入是上一步输出的n位结果 + 新一个b位分组，输出为一个n位结果，最终值即hash值。

- 设计安全hash函数可以归纳为设计具有**抗碰撞能力**的压缩函数（输入是定长的）问题。

MD族

MD2：先数据补位到16的倍数字节长，追加16位校验和，计算出hash值。

MD5：输入任意长消息，以b=512bit分组，输出n=128bit。

- 目标：计算安全，利于快速软件实现；
- 步骤：
    1. 增加填充位，让最后1个分组也padding到b=512-61=448位，留64位给填充长度；
    2. 填充长度，在填充位padding的后面，把末尾64位填上（大于则取模，小端序）；
    3. 初始化MD缓存（保存Hash函数迭代的中间结果和最终结果，4个32位寄存器）；
    4. 计算Hash值，以512bit的分组处理，压缩函数HMD5的输入是当前要处理的512bit分组Y_q + n=128bit缓冲区$ABCD$的内容，进行四轮运算，利用一个16×4的随机矩阵T，每轮用一列16个元素；第四轮的输出与第一轮CV_q的输入相加得到CV_{q+1}，即$CV_{q+1} = \mathrm{SUM32}(CV_q, \mathrm{RFI}[Y_q, \mathrm{RFH}[Y_q, \mathrm{RFG}[Y_q, \mathrm{RFF}[Y_q, CV_q]]]]).$
    5. 输出，当所有L个512bit分组处理完之后，第L个分组的输出CV_L就是128bit长度的消息摘要。

SHA

SHA-1：输入长度L小于264位的消息，以b=512位进行分组，输出n=160位的MD。

步骤：

1. 增加填充位，到比512的整数倍少64位；
2. 填充长度，64位表示填充前的报文长度（大端序）；
3. 初始化MD缓存：保存在160位的缓冲区中，用5个32位寄存器ABCDE表示；
4. 以512位的分组为单位处理消息，压缩函数共进行十轮运算，十轮运算分成两组（每组五轮？），每轮执行16迭代20步。
   $CV_0 = IV, CV_{q+1} = \mathrm{SUM32}(CV_q, ABCDE_q), MD = CV_L$

RIPEMD

RIPEMD-160：输入任意长，以b=512位的分组单位进行处理，输出n=160位的消息摘要。

步骤：和SHA-1相同，但是RIPEMD-160的十轮运算每轮执行16步迭代。

【比较】

- 抗强碰撞性：MD5由于MD长度n=128 < 160比较短，易受攻击；
- 抗密码分析：MD5 < SHA-1 < RIPEMD-160
- 速度：MD5迭代次数少，最快
- 端序：MD5和RIPEMD-160都是小端序，SHA-1是大端序

比较hash码只需看以下几个，就能知道hash加密的强度和长度	MD5	SHA-1	RIPEMD-160
摘要长度	128 bits	160 bits	160 bits
基本处理单元	512 bits	512 bits	512 bits
步数	64(4 轮，每轮 16 步)	80(4 轮，每轮 20 步)	160(5 轮，每轮 16 步)
最大消息长度	∞	$2^{64}-1$ 位	$2^{64}-1$ 位
基本逻辑函数	都很简单 4	4	5
使用的加法常量	64 矩阵	4	9
低端位/高端位结构	低位在前	高位在前	低位在前

——现在未被攻破的有SHA-2，RIPEMD-160，其他的MD、SHA、RIPEMD系列都被攻破了。

3、对于网站身份认证，基于Basic认证和基于表单认证的工作原理和各自特点

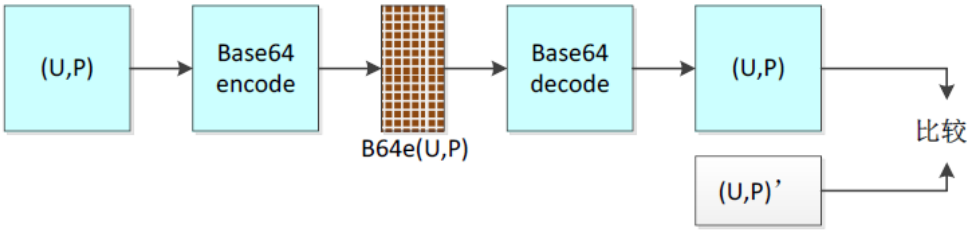
（电子）身份认证需要大量使用消息认证技术（Message Authentication）。

网站用户认证：

- 最简单的：HTTP的Basic认证（HTTP是面向一次连接的无状态网络协议）
- 改进解决Basic认证的问题：基于表单的身份认证
- 若干增强认证技术：手机口令（短信验证码）、动态口令（网易将军令）、USB KEY（银行U盾，存储了用户密钥 / 数字证书）、数字证书（由CA发行）

Basic认证

用户身份凭证：账号+静态口令，明文传输并比较。



【缺点】本地保存可能被盗，明文传递容易被监听，安全性低；服务器每次都要进行身份验证，效率低。

改进：

- 使用加密技术，把password作为对称密码，传输(U || E_k[U])，服务器解密后与U比对。【缺点】无法防范重放攻击（attacker偷了user的数据，自己发给服务器以通过认证）
- 使用消息认证的MAC技术，共享密码K（口令）。缺点同上。
- 采用挑战/响应机制，第一次server向client返回一个挑战码M，第二次client才用MAC认证发送信息，给server认证。

表单验证+session

解决的问题：账号口令长期本地保存，服务器每次都进行身份验证。

session首次启动会产生唯一标识符（通常用Cookie技术存储），client每次发送HTTP请求都会附带它，使server关联前后多次请求。结束session时销毁。

基于表单的身份认证：

1. client向server发送请求，获得包含表单的页面
2. 用户填好表单（**明文**），client发到server，server验证通过则启动session并返回给client
3. client后续请求都包含session的唯一标识符，server验证这个标识符

改进：

1. 挑战/响应机制，用口令K对称加密计算$E_k[M||U]$，发送密文和U，client比对。
2. 传输时使用传输层SSL协议传输HTTP请求（HTTPS）

常见的不安全做法：浏览器保存账号口令Cookie（口令泄露），或者加密保存账号口令Cookie（重放攻击）。非要保存，就要加上时间戳t，server验证t。

四、必会访问控制技术（Lecture5）

1、防火墙的设计目标和局限性

防火墙：在被保护网络和其他网络之间（内/外网，专用/公共网）实施访问控制策略的一组设备（软＆硬件）。作用有隐藏内部网络结构和资源、保护不安全的网络服务、执行网络间的访问控制策略、统一集中安全管理、记录并统计网络的使用情况、监视和预警。

设计目标：所有的通信（内→外 / 外→内）必须经过防火墙，只有授权的通信（根据防火墙的安全策略）才能通过。防火墙本身对于渗透必须免疫。

常用技术：服务控制、方向控制、用户控制、行为控制。

技术分类：

- 网络层-包过滤技术（典型例子是ACL）
- 网络层-地址转换NAT，隐藏内部网络，节省IP地址空间
- 传输层-电路层网关（现在很少用）
- 应用层-应用层代理Proxy（可以翻墙）

【局限性】

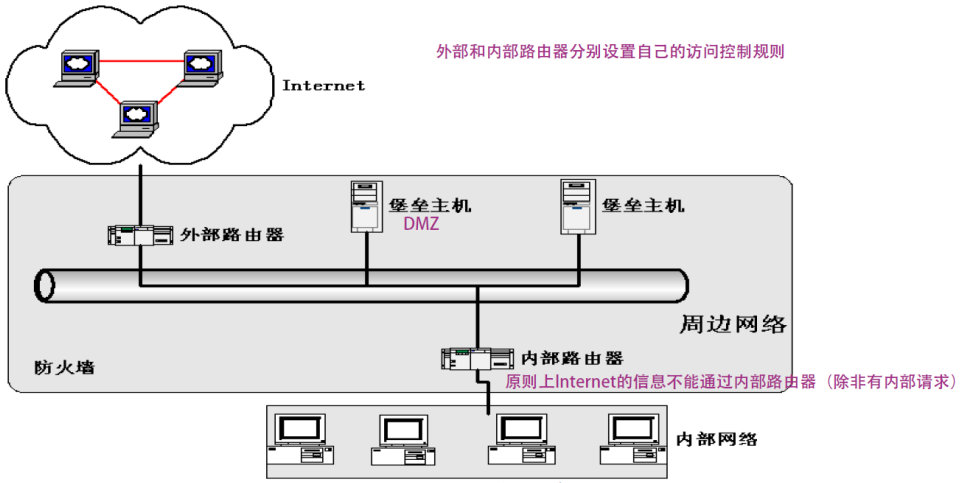
- 需要用户定义访问控制规则（ACL）
- 防外不防内，即不能防止内部恶意的攻击者
- 无法控制没有经过它的连接
- 不能很好防止病毒和信息扩散（只能对流量进行控制，但无法杀毒）
- 不能替代内部网络系统的安全管理
- 无法防范全新的威胁和攻击

2、防火墙屏蔽子网结构

防火墙的配置结构分类：

- 单盒（简单的包过滤）
- 屏蔽主机（包过滤路由器+堡垒主机）：外部只能到达堡垒主机，需要内部主机性能好+对外连接少，否则就选择屏蔽子网。
- **屏蔽子网**：

- 不设防区DMZ（堡垒主机放在这里）
- 外部路由器（只允许外部访问DMZ），保护周边网络和内部网络，主要针对外部网络，是第一道屏障
- 内部路由器，隔离周边网络和内部网络，是第二道屏障



- DMZ(Demilitarized zone), 不设防区
  - 通常放置DNS/Web/Email/FTP/Proxy Server等
- 外部路由器
  - 只允许互联网对DMZ的访问 堡垒主机
  - 拒绝所有目的地址为内部网络地址的包
  - 拒绝所有不以内部网络地址为源地址的包进入互联网

外来者无法偷偷加入内部网络主机

- 内部路由器
  - 保护内部网络，防止来自Internet或DMZ的访问 万一DMZ被攻陷，也无法通过内部路由器访问内部主机
  - 内部网络一般不对外部提供服务，所以拒绝外部发起的一切连接，只允许内部对外的访问

36

3、对防火墙的争论

破坏了Internet端到端的特性，阻碍新应用发展；

有先天不足，不能解决网络内部的安全问题，防外不防内；

给人一种误解，降低了人们对主机安全的意识（……）。

【其他】虚拟局域网VLAN作用：防止广播风暴，降低泄密风险，有物理层（管理员将交换机端口分组）、数据链路层（根据MAC地址配置，用户搬家也没关系）

五、必会互联网安全协议

1、CIA基本概念

计算机网络安全体系结构如下：

安全目标：CIA

C：Confidentiality，保密性（防范被动攻击，信息不会泄露给未授权实体）

I：Integrity，完整性（防范主动攻击，信息不被篡改）

A：Availability，可用性（防范拒绝服务攻击）

安全服务

X.800有认证服务、保密服务、数据完整性、访问控制、抗抵赖、可用性服务

安全机制及其算法实现

普通（不属于任何层）+特定（8种）

2、网络层IPsec：工作原理、AH/ESP、安全关联SA、模式、安全关联组合

功能：认证（可选）、保密

原理：可以在IP层加密 and / or 认证所有流量，在IPv4（可选）和v6（必选IPsec）中都适用。

应用：建立虚拟专用网VPN使分支机构安全互联、远程安全访问互联网

实施载体：

- 可以在主机上实施（保障端到端安全，每个会话）
- 在防火墙上实施（内部所有应用）
- 在路由器上实施（VPN）

AH/ESP

认证头AH（Authentication Header）：**包认证**的扩展报头。

封装安全载荷ESP（Encapsulated Security Payload）：**包加密**的扩展报头，可以分成支持/不支持认证两种。

它们提供的安全服务包括：

- 访问控制
- 无连接完整性
- 数据源发认证
- 拒绝重放数据包
- 保密性（加密）
- 有限的信息流保密性

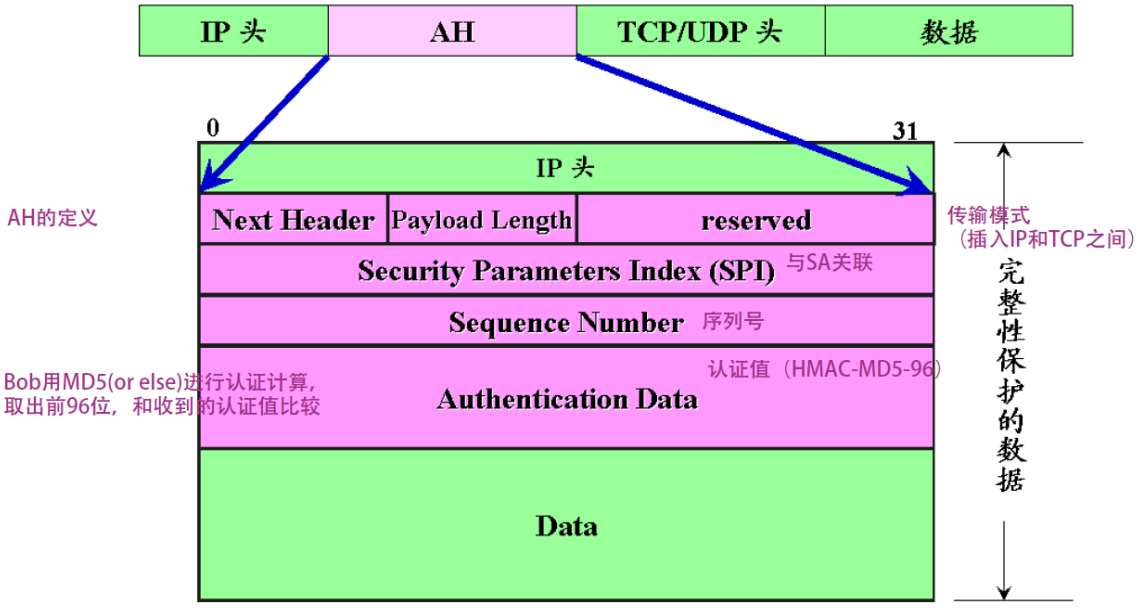
三类报头：
只认证、只加密、认证并加密

	AH	ESP(只加密)	ESP(加密并认证)
访问控制	✓	✓	✓
无连接的完整性	✓		✓
数据源发认证	✓		✓
检测重放攻击	✓	✓	✓
机密性		✓	✓
有限的通信流保密		✓	✓

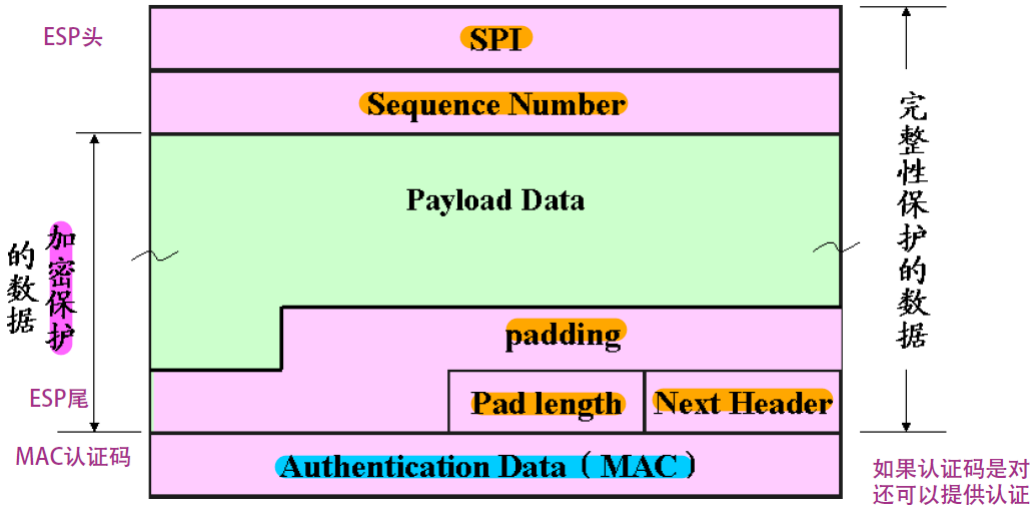
完整性

保密性

AH：基于MAC，双方共享公钥。



ESP：可选是否认证，能完成加密。



安全关联SA

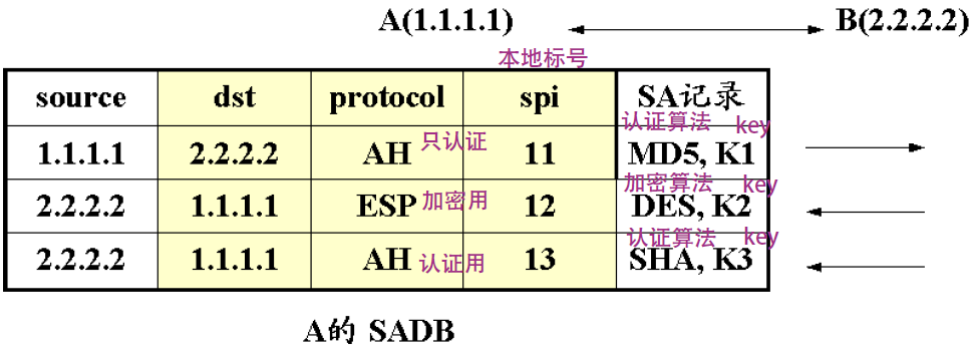
安全关联SA（Security Association）：IPsec通信双方对安全信息参数的协商，是**有方向**的，发送方和接收方之间的单向关系（如果要双向就要两个SA）。

SA由三个参数唯一确定：

- 安全参数索引SPI：由AH和ESP携带
- IP目的地址IPDA：只允许使用单一地址
- 安全协议标识：区分是AH还是ESP的SA

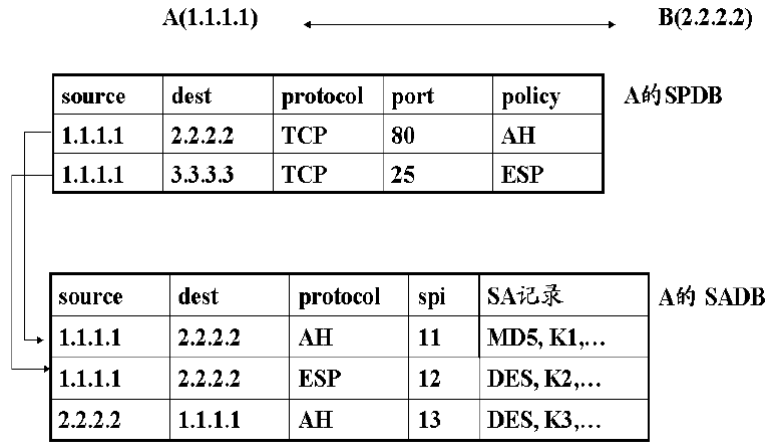
SA的其他参数：序号计数器、序号溢出标志、反重放窗口、……、IPsec协议模式、Path MTU

SA表项的集合就是安全关联数据库SADB，Entry由source IP唯一标识。



SA选择子：IP流量与特定SA相关是通过安全策略数据库SPDB的。

SPDB：把IP信息流与SA联系起来的手段，决定了对流入和流出的哪些数据包进行安全操作。



模式

传输模式：为上层协议、IP包载荷提供保护，用于两台主机之间的端到端通信。

- 

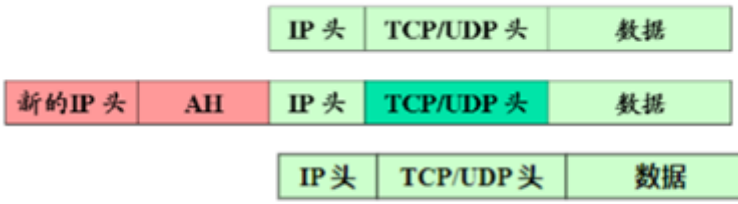
- AH： 

- ESP： 

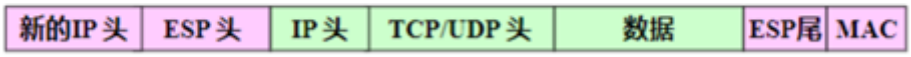
隧道模式：为整个IP包提供保护，有一个新的IP报头，途中路由器不能检查内部IP报头。

- 

- AH：



- ESP：



安全关联组合

单个SA可以实现AH或者ESP，但是不能两者都实现。

安全关联组合：一个SA序列，可以是

- 传输邻接：只用传输模式，一个IP包可以有多个安全协议（只能一级组合）
- 隧道迭代：通过隧道应用多层安全协议

【考法】IPsec：有一个需求，应该怎么设计；画上AH/ESP头的传输/隧道模式，指出哪些是明文，哪些是密文

3、IKE：工作阶段和工作模式、IPsec和IKE的工作过程

作用：密钥管理，解决了**在不安全的网络环境中安全地建立或更新共享密钥**的问题。非常通用，可以用于IPsec（目前只在这里用了）、SNMPv3、RIPv2、OSPFv2等协议协商安全参数。

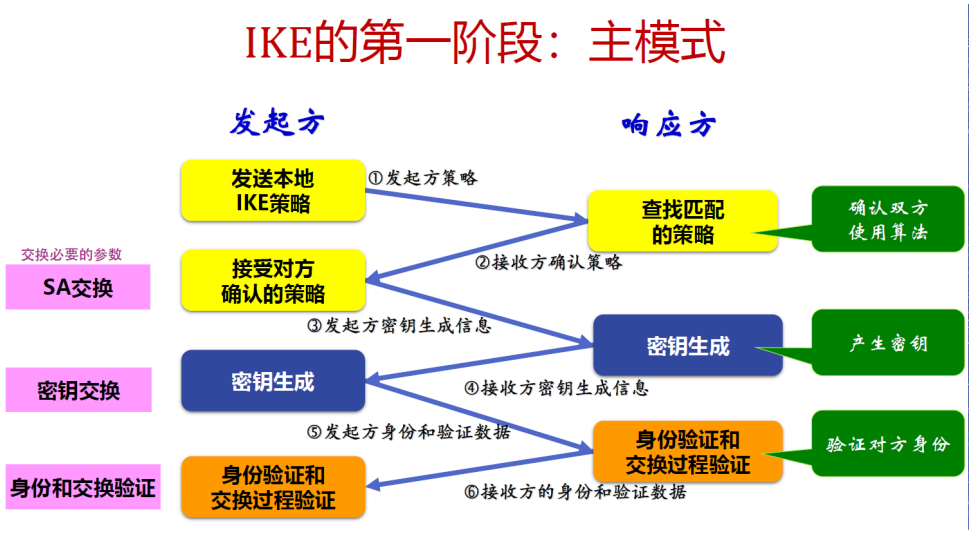
IPsec支持两种密钥管理类型（手工 / 自动），其中IKE用于自动协商交换密钥、建立SA、维护SADB。IKE的功能有密钥&SA定时更新、允许IPsec提供反重放、端与端之间动态认证服务。

（密钥只通过安全参数索引SPI与认证、保密机制联系。）

报文格式：和ISAKMP同，可以在任何传输层协议（TCP、UDP）/网络层协议（IP）上实现。

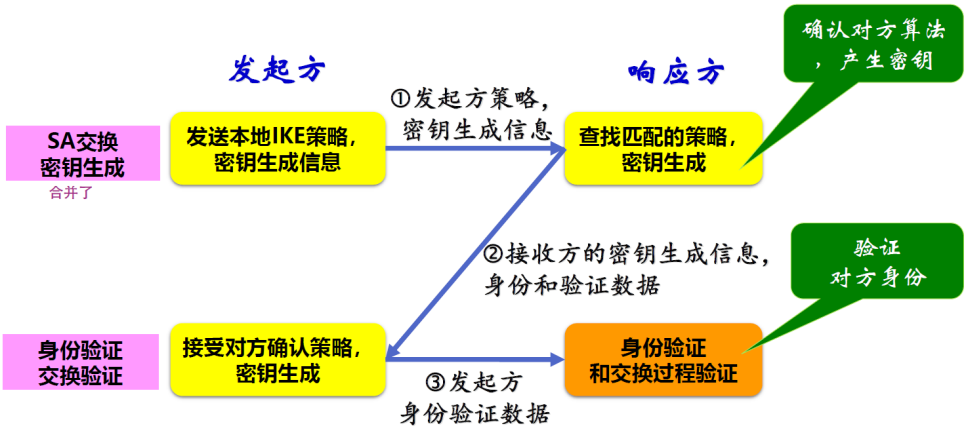
工作阶段

1. **协商IKE SA**（主模式 / 快速模式），建立一个经过验证的安全通道。协商内容包括加密算法、哈希算法等。
    - 主模式【只能采用IP地址作为ID协商】：对双方身份保护，协商之前双方必须计算产生Cookie（标识每个协商交换过程），写入ISAKMP报文的Cookie域中。如果⑤⑥中的hash值相同，则认证成功。

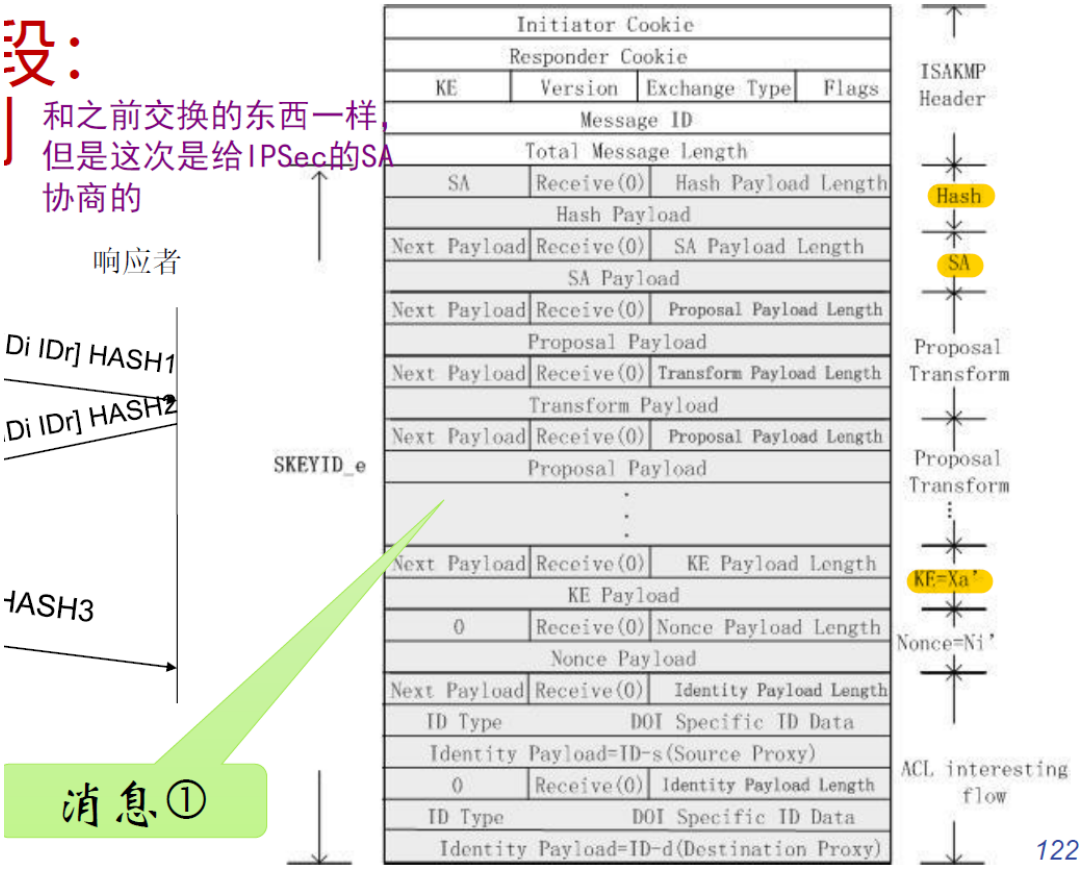


    - 快速模式（积极）【还能适用于一方地址为动态的情况】：无需保护，效率更高。消息②就开始计算Hash，消息③计算Hash值并比对，一致则认证成功。

IKE的第一阶段：快速模式



2. 使用已建立的IKE SA**协商IPsec SA** （只有快速模式），可以建立多个SA，存入SADB。



工作模式

传输：适用于**端节点**到端节点，IP头与数据之间插入IPsec头，保护数据载荷。

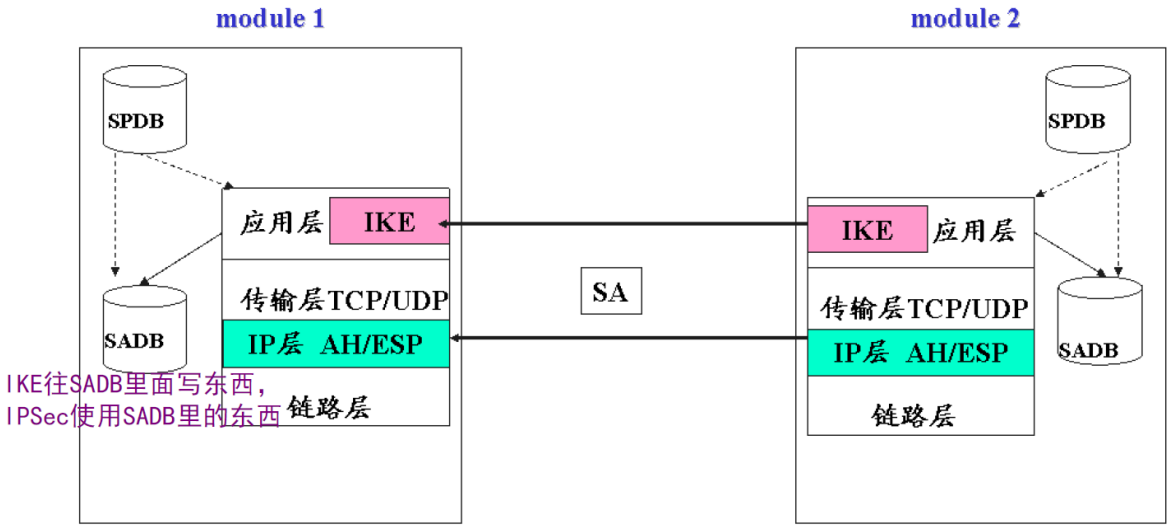
隧道：适用于**安全网关**到安全网关，端系统不需实现IPsec。

工作过程

IKE：

- 以**守护进程**的方式在后台运行。两个守护进程之间通过**UDP**通信。
- 使用SADB、SPDB，两个数据库都保存在OS内核。
- 对于每个出 / 入的IP包，先**查SPDB表**确定是丢弃 / 绕过 / 应用**IPsec**；
    - 若应用，**IPsec查询SADB**看是否有合适的SA；
        - 若有，则进行相应的IPsec处理；

        - 若无，则向**IKE**守护进程发出**创建SA**请求。
            - IKE收到后**查询SPDB**，得到所有协商参数，然后向**远程另一IKE进程**发出协商请求，开始协商创建SA。
            - 协商成功后，**新协商的SA被加入SADB。**
- 当管理员指示IKE守护进程不再使用某SA时，
    - IKE守护进程从SADB中删掉对应SA，并向远程IKE守护进程发送删除信息；
    - 远程IKE收到后可以删除SA，也可以保留但标记不允许用于通信。

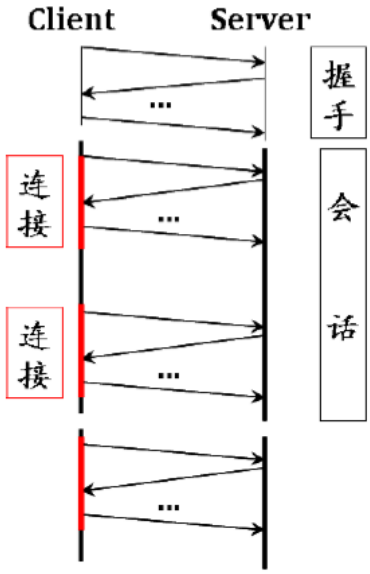


4、传输层SSL：SSL体系结构、记录协议

SSL（Secure Socket Layer，安全套接字协议）：工作在**TCP**协议上，不支持UDP；与应用层协议无关，为端到端的应用提供保密性、完整性、身份认证等安全服务。

SSL的相关概念

- 会话（session）：client和server之间的一个虚拟连接关系，称为关联（association）。通过**握手协议**建议，用来**协商**密码算法、主密钥等信息，这些信息可以**被多个连接共享。**参数有session ID等。
- 连接（connection）：一个特定是通信信道，一般很短。一次访问中的**多个连接可以共享同一个会话协商的信息。**参数有序列号等。

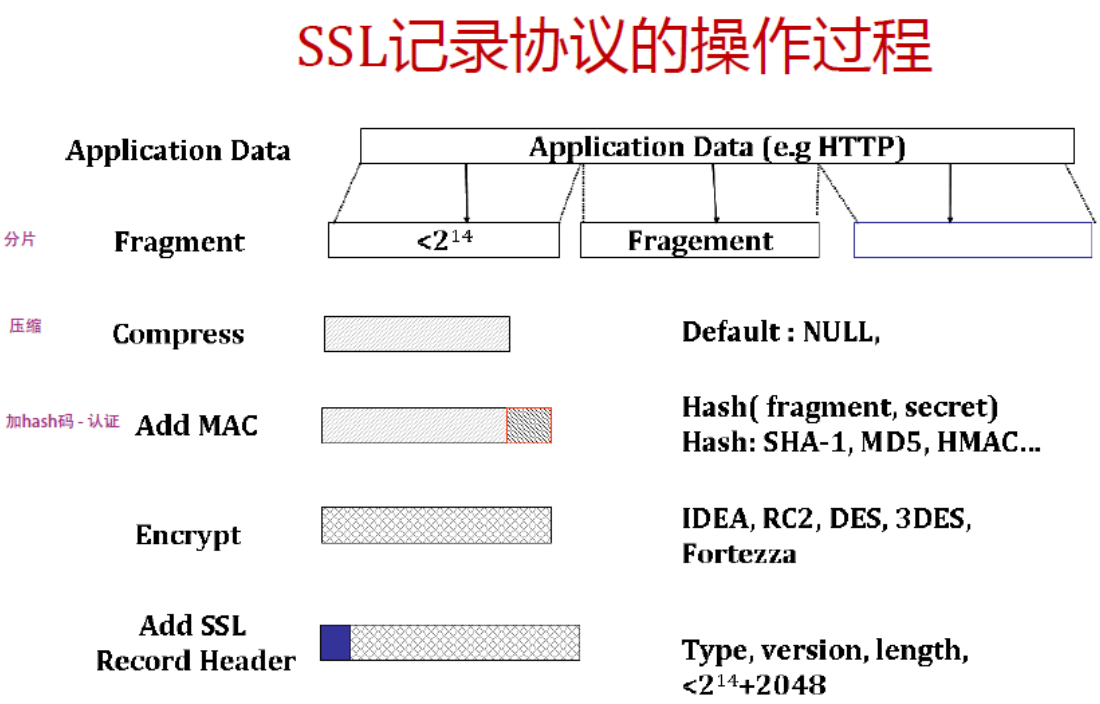


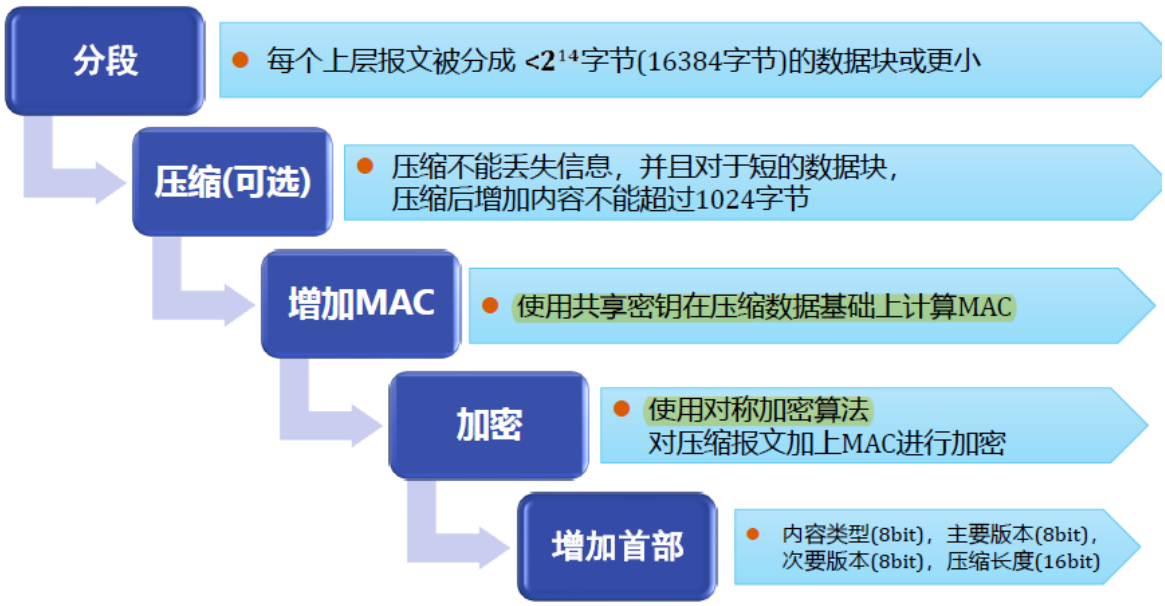
SSL体系结构

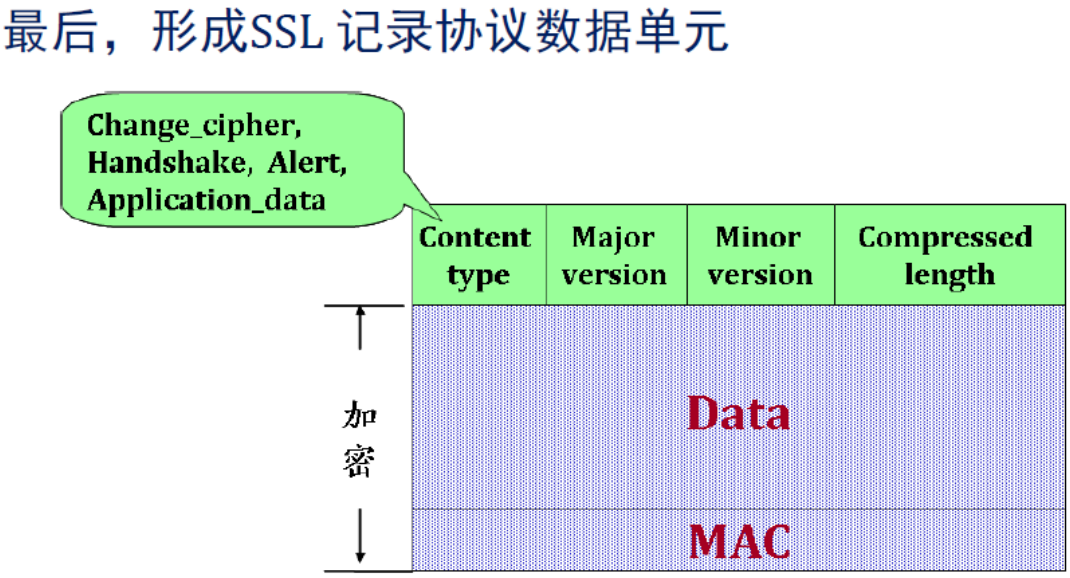
SSL协议的组成：

- SSL握手协议，协商密码**算法**、主会话**密钥**，进行**server**和client（可选）的身份**认证。**（主密钥不直接用于加密/认证，而是用来产生一系列密钥用于MAC、加密、IV等）有四个阶段——
    1. 建立安全能力：相互说hello，协商加密算法、密钥交换方法、MAC算法、随机数等一系列安全参数
    2. 服务器认证和密钥交换：server发送证书（取决于密钥交换算法，匿名DH不用发证书）、还可能需要发交换密钥相关的报文`server_key_exchange`，等待client响应
    3. 客户认证和密钥交换：client验证server的证书，发送`client_key_exchange`（包含pre master secret和消息认证码密钥），可选是否发送客户端证书
    4. 结束：将pre master secret计算转化为master secret，然后产生用于加密和消息认证的所有密钥。
- SSL修改密码规约协议：握手协议的结束阶段发送，通知接收方以后就使用刚才协商好的密码算法和密钥进行加密/认证。
- SSL记录协议，提供保密性、报文完整性服务（用握手协议定义的共享密钥对SSL载荷进行加密，以及计算MAC码，注意是**先对明文算MAC码再整体加密**）
- SSL告警协议，向对等实体传递警报，如标识为致命会使SSL终止连接。

SSL记录协议







安全性

安全性：

- 建立在**RSA算法**的安全性上。
- 可以**防范重放攻击**（每次安全连接都有一个128位的随机数作为连接序号）。

5、应用层HTTPS：基本工作原理以及防范ARP欺骗、报文篡改的技术保障

HTTP：超文本传输协议，用于发布资源（支持超文本链接）

HTML：超文本标记语言，用于编写文档

URL：统一资源定位，通过互联网引用其他可访问文档或资源

Web安全目标：

1. 确保web服务器存储数据的安全
2. 确保web客户端的计算机是安全的
3. 确保服务器和浏览器之间信息传输的安全

Web安全威胁：按威胁方式分有主动 / 被动攻击，按威胁位置分有web服务器（Basic、MD、SSL、访问控制） / 客户端 / 服务器和客户端之间通信的安全（IPsec、SSL、SET都是通信安全）。

安全方法：

- IP级：IPsec
- TCP级：SSL / TLS
- 应用级：SET、PGP、S/MIME等

HTTP的缺陷：

- 明文传输（没有加密），不校验数据完整性（没有消息认证）
- 无状态连接，无法验证双方身份的真实性（没有数字签名）

针对HTTP的攻击：监听嗅探、篡改劫持、伪造服务器（中间人攻击如ARP欺骗改变IP-MAC映射表、DNS欺骗改变域名-IP地址映射等）

基本工作原理

HTTP层：翻译成HTTP请求

SSL层：借助下层协议的信道，安全地协商出一份加密密钥，以此加密HTTP请求

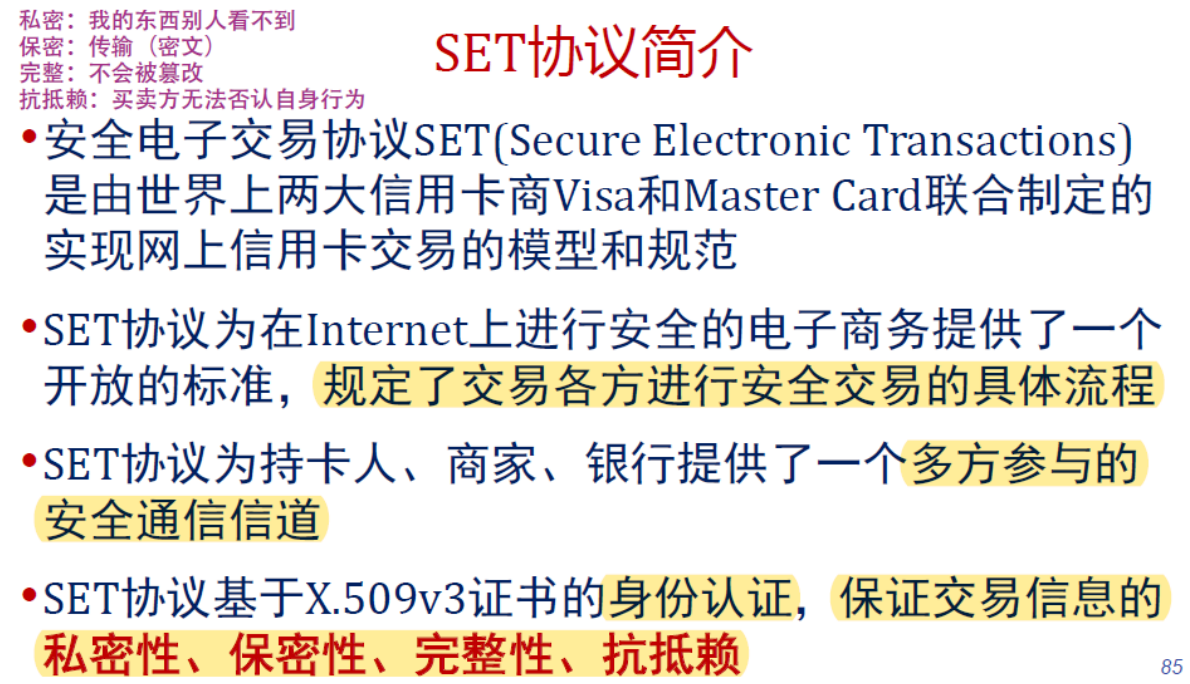
TCP层：与443端口通信，传递SSL处理后的数据；接收端相反

防范ARP欺骗、报文篡改的技术保障

ARP欺骗和嗅探攻击：无法避免，但是嗅探到的是密文。

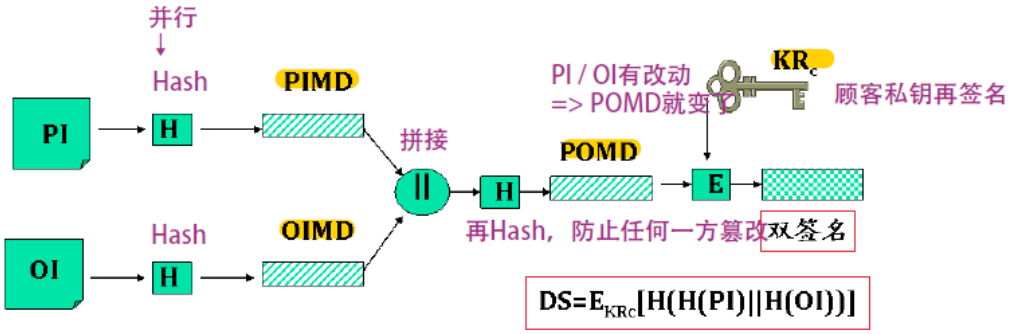
报文篡改：内容是无法识别的密文，攻击者无法对通信内容进行篡改。

6、应用层SET：电子交易工作原理，双签名机制的设计目标和工作过程

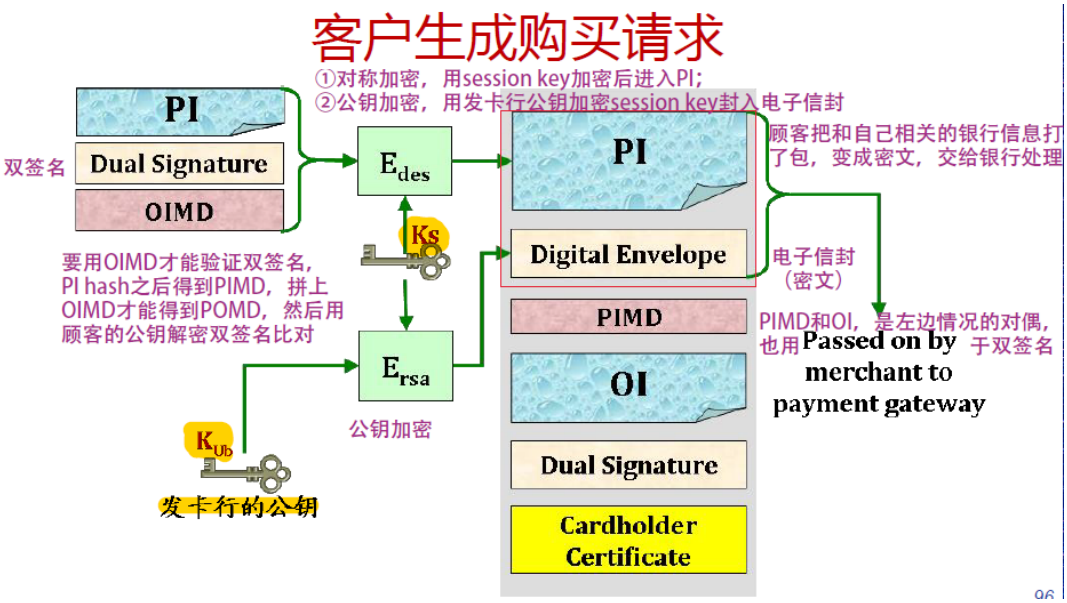


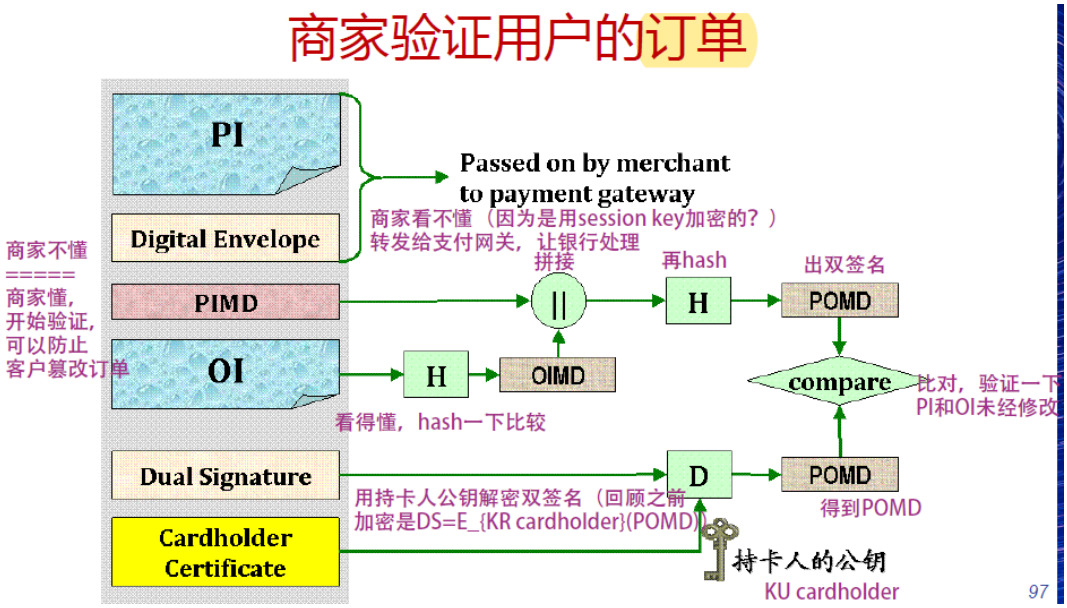
双签名机制

- 双签名的目的是为了连接两个发送给不同接收者的报文
  - PI=支付信息
  - PIMD=PI报文摘要
  - OI=定购信息
- OIMD=OI报文摘要
- H=散列函数（SHA-1）
- E=加密（RSA）
- ||=拼接
- KRc=顾客的私有签名密钥



1. 购买请求



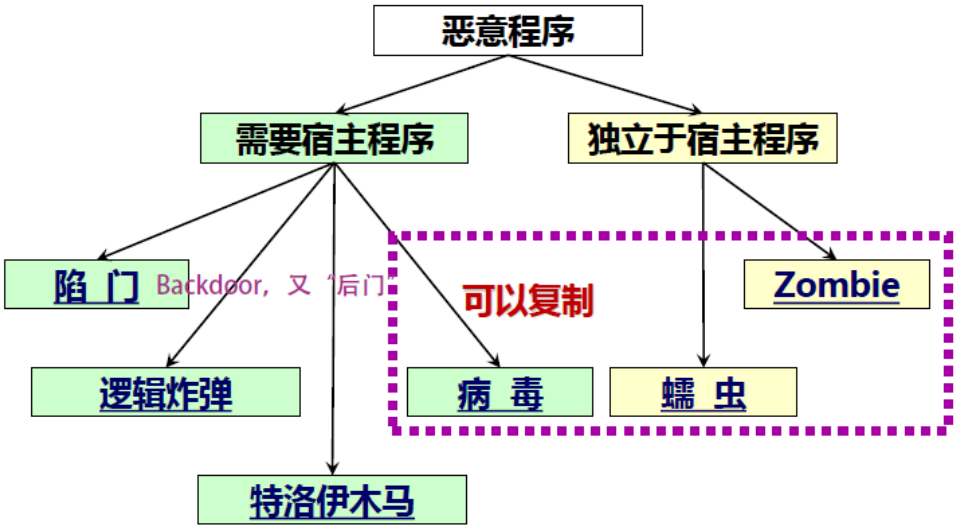


2. 支付授权：银行解开Payment Block，验证PI与OIMD生成的双签名与发过来的是否一致，确认PI未被更改，是同一个transaction ID。

3. 支付获取

六、必会软件入侵

1、陷门、逻辑炸弹、特洛伊木马、Zombie、病毒和蠕虫各类恶意软件属性



特性	病毒	蠕虫	木马
宿主	需要	不需要	需要
表现形式	不以文件形式存在	独立的文件	伪装成其他文件
传播方式	依赖宿主文件或介质	自主传播 自我复制	依靠用户主动传播 用户傻了才点一下
主要危害	破坏数据完整性、系统完整性 篡改	侵占资源 侵占CPU、内存、带宽	留下后门 窃取信息 偷东西
传播速度	快	极快	慢

特性	病毒	蠕虫	木马
宿主	需要	不需要	需要
表现形式	不以文件形式存在	独立的文件	伪装成其他文件
传播方式	依赖宿主文件或介质	自主传播 自我复制	依靠用户主动传播 用户傻了才点一下
主要危害	破坏数据完整性、系统完整性 篡改	侵占资源 侵占CPU、内存、带宽	留下后门 窃取信息 偷东西
传播速度	快	极快	慢