- **网络安全协议**
  - 综述
    - 概述
      - **网络安全基础**
      - 3A标准
      - 不安全因素
      - **不安全原因**
      - **网络安全特征**
      - 主要任务
      - **攻击手段**
    - 网络体系结构
      - 安全体系结构概述
      - **安全攻击**
      - **安全机制（8种）**
      - **安全服务（系统提供，5类）**
    - 安全模型
      - 网络安全模型
      - 网络访问安全模型
    - 安全协议概述
      - 链路层
      - 网络层
      - 传输层
      - 应用层
    - 密码学基础
      - 分类
      - 对称密码
      - 非对称密码
    - 数字签名与认证
    - 网络安全标准组织
  - 公钥基础设施PKI
    - 概述
      - 定义
      - 功能
      - 证书结构
      - 产生背景（帮助理解）
      - 组成（基本组成包括：CA、RA、证书库）
      - 版本更替
    - PKI密码技术
      - 数字信封
    - 功能操作
      - PKI运行过程概述
      - 证书操作
  - IPSec：AH和ESP
    - IPSec背景
    - IPSec概述
      - 定义——1组协议的集合
      - 安全协议——AH、ESP
      - 安全算法
    - 安全组合（SA）
      - **背景需求**
      - SA概述
      - SA标识——SPI，IP，Protocal
      - **SAD——安全关联数据库**
      - SPD——安全策略数据库
    - 认证头标AH

- AH概述
- AH作用
- AH组成
- 封装安全载荷头标ESP
- ESP概述
- ESP作用
- ESP组成
- 封装模式
- NAT冲突
- IPSec全流程实例
- 背景介绍
- 步骤1——设置SPD规则
- 步骤2——IKE协议协商SA
- 步骤3——发送数据包
- 步骤4——接受数据包
- 源端/目的端处理流程
- 源端外出
- 目的端进入

IPSec：IKE（主要注意原理，不要拘泥于实现细节）

- IKE背景
- IKE定义
- IKE功能
- IKEv1流程
- 阶段一（建立ISAKMP SA 双向）
- 阶段二（建立IPSec SA 单向）

SSL/TLS基本协议

- SSL背景
- SSL概述
- 定义
- 功能
- 服务
- 协议构成
- 原理（数据处理过程和序列号使用）
- 组成
- TLS
- 会话
- 连接
- 会话重用

防火墙

- 防火墙概述
- 定义
- 功能
- 分类
- 局限性
- 包过滤防火墙
- 思想：
- 过滤规则：
- 优缺点：

NAT技术

- 分类——SNAT 和 DNAT
- NAT方式

虚拟专用网VPN

- VPN五大安全技术

应用层安全协议

- PGP
- 签名->压缩->加密
- SET
- 背景

三类四种+kerberos协商会话密钥

随机数 时间戳 计数器

P139 压缩长度 8bit改为16bit

POP3 MAP SMTP这些需要知道

网络安全协议

综述

概述

网络安全基础

理论角度：网络安全是建立在**密码学以及协议设计**的基础上的
技术角度：决定性因素：①网络设备的硬件和软件的安全。②设备的访问控制。

3A标准

- **认证**（Authentication）
- **授权**（Authorization）
- **计费**（Accounting）　*注意不是审计！*

不安全因素

1. **物理因素：**设备在物理防护上不安全，电磁波泄漏等。
2. **系统因素：**系统软、硬件漏洞，病毒感染，漏洞入侵。
3. **网络因素：**网络协议流程漏洞，会话劫持，数据篡改，故意的网络拥塞，（分布式）拒绝服务。
4. **管理因素：**（管理员）安全意识淡漠，误操作。

不安全原因

- **自身原因：**系统软硬件、协议存在缺陷
- **开放性：**
    1. **系统开放：**计算机及计算机通信系统是根据行业标准规定的接口建立起来的。
    2. **标准开放：**网络运行的各层协议是开放的，并且标准的制定也是开放的。
    3. **业务开放：**用户可以根据需要开发新的业务。

网络安全特征

1. 机密性（C）：信息**不泄露**给非授权用户、实体、过程

   攻击：窃听

2. 完整性（I）：数据**存储、传输**过程不被修改、破坏

   攻击：修改

3. 可用性（A）：需要时能**存取所需信息**，安全功能**不明显影响服务使用。**

   攻击：中断

4. **可认证性：** 与完整性存在关联，要求数据来自所声称的实体，或者合法用户。

   攻击：伪造

主要任务

1. 保障网络和系统
2. **保证信息：机密、完整、可认证、不可否认**地传输和使用

攻击手段

1. **社会工程**：攻击者可通过各种社交渠道获得有关目标的结构、使用情况、安全防范措施等有用信息从而提高攻击成功率。
2. **口令破解**：攻击者可通过获取口令文件，然后运用口令破解工具获得口令，也可通过猜测或窃听等方式获取口令
3. **地址欺骗**：攻击者可通过伪装成被信任的IP 地址，邮件地址等方式来骗取目标的信任
4. **会话劫持**：在合法的通信连接建立后，攻击者通过阻塞或摧毁通信的一方来接管已经过认证建立起来的连接，从而假冒被接管方与对方通信
5. **网络窃听**：网络的开放性使攻击者可通过直接或间接窃听获取所需信息
6. **数据篡改**：攻击者可通过截获并修改数据或重放数据等方式破坏数据的完整性
7. **恶意扫描**：攻击者可编制或使用现有扫描工具发现目标的漏洞，进而发起攻击
8. **基础设施破坏**：攻击者可通过破坏DNS 或路由信息等基础设施，使目标陷于孤立
9. **数据驱动攻击**：攻击者可通过施放病毒、特洛伊木马、数据炸弹等方式破坏或遥控远程目标
10. **拒绝服务**：攻击者可直接发动攻击，也可通过控制其它主机发起攻击，使目标瘫痪，如发送大量的数据洪流阻塞目标

网络体系结构

安全体系结构概述

- **X.800：**（OSI组织制定的安全架构）
  1. **安全攻击：**损害机构所拥有信息的安全的任何行为。
  2. **安全服务：**系统提供的对资源进行特殊保护的进程或者通信服务。
  3. **安全机制：**设计用于检测、预防安全攻击或者恢复系统的机制。

     安全机制—→实现—→**安全服务**—→抵御—→**安全攻击**

     接下来详细介绍上述的1、2、3

安全服务与攻击的关系

	攻　　击					
服　务	报文分析	流量分析	伪装	重放	篡改	拒绝服务
对等实体认证			Y			
数据源认证			Y			
访问控制			Y			
机密性	Y					
流量机密性		Y				
数据完整性				Y	Y	
非否认服务						
可用性						Y

安全服务与机制的关系

	安全机制							
安全服务	加密	数字签名	访问控制	完整性	认证	流量填充	路由控制	公证
对等实体认证	Y	Y			Y			
数据源认证	Y	Y						
访问控制			Y					
机密性	Y						Y	
流量机密性	Y					Y	Y	
数据完整性	Y	Y		Y				
非否认服务		Y		Y				Y
可用性				Y	Y			

安全攻击

- **主动攻击：更改或伪造**数据流。

  分类：

  ①**伪装**：伪装指的是攻击者以他人或**合法**实体的身份发送数据或执行操作，以便欺骗目标系统或**绕过安全检查**。这种伪装可能会导致未经授权的访问、数据泄露或其他恶意行为。

  ②**重放**：重放攻击是指攻击者通过**录制并重新播放**之前的有效通信数据或认证令牌来**模拟合法用户**的行为。

  ③**篡改**：篡改攻击是指攻击者在数据传输过程中**修改或篡改**数据包的内容，以**改变数据的意义**或**破坏数据的完整性**。

  ④**DoS**：攻击者利用大量**合理服务**请求来占用过多的攻击目标**服务资源**，从而使合法用户无法得到服务响应的过程。一般采取一对一的模式。

  DDoS：攻击者控制僵尸网络中大量**僵尸主机**向攻击目标发送大流量数据，耗尽攻击目标的系统资源，导致其无法正常响应服务请求。

- **被动攻击**：对传输进行**偷听与监视**，获得传输信息，但不对通信和数据做任何修改。

  分类：

  ①**窃听攻击**：窃听攻击是指攻击者在不被授权的情况下**监听通信传输**，以获取传输的**敏感信息或数据**。

  ②**流量分析**：流量分析是一种被动攻击技术，攻击者通过分析网络**流量模式和数据包之间的关系**来获取有关通信内容和参与者的信息。

③**破解弱加密数据流**：攻击者尝试破解使用弱加密算法或密钥的数据流。攻击者可能会**截获加密的通信传输**并尝试使用各种技术（如密码破解、密钥破解）来**破解加密数据**以获取敏感信息。

安全机制（8种）

1. **加密**：用加密算法对信息加密, 保护信息的**机密性**
2. **数字签名**：用签名算法对信息进行计算, 计算结果附加于信息单元。用于**身份认证、数据完整性**和**非否认服务**
3. **访问控制**：用于实施资源**访问权限**的机制
4. **数据完整性**：用于确保信息的**完整性**
5. **认证交换**：通过信息交换以**确保实体身份**，包括公知密码、特征、位置信息等
6. **流量填充**：填充信息，**防止流量分析**
7. **路由控制**：能够为特定数据选择特定**基于路由的安全通道**
8. **公证**：采用**可信任的第三方**以确保一些信息交换的属性

安全服务（系统提供，5类）

1. **认证**：提供某个实体的身份保证
   例：对等实体认证、数据源认证->需要了解两者差别
2. **访问控制**：保护资源, 防止对它的非法使用和操纵
3. **数据机密性**：保护信息不被泄露
   例：连接保密性、无连接保密性、选择域保密性、流量保密性
4. **数据完整性**： 保护信息以防止非法篡改
   例：具有恢复功能的连接完整性、无恢复功能的连接完整性、选择域连接完整性、无连接完整性、选择域无连接完整性
5. **不可否认性**：防止参与通信的一方事后否认
   例：源点的不可否认性、目的节点的不可否认性

安全模型

网络安全模型

信息在**传输过程**的安全

网络访问安全模型

网络或系统**本身**的安全

安全协议概述

链路层

链路隧道协议, 加密技术

网络层

包过滤机制, NAT, IPSec协议, VPN

传输层

SSL/TLS 协议

应用层

HTTPS, PGP, S/MIME, DNSSEC等

密码学基础

三类四种：

对称、非对称(加密、签名)、哈希

分类

- **发展进程**：
  1. **古典密码**：基于字符替换的密码，现在已很少使用了，但是它代表了密码的起源
  2. **对称密钥体制**：加密密钥和解密密钥相同，这些算法也叫作单钥密码体制
  3. **非对称密钥体制**：加密密钥和解密密钥不同，也叫公钥密码体制或双钥密码体制
- **加密模式**：
  1. **序列密码**：**按位或字节加密**。又称流密码，是手工和机器时代主流。
  2. **分组密码**：明文分成**固定长度的组**，用同一密钥和算法对每一块加密，输出也是固定长度的密文。

对称密码

- **DES：**
  - **数据分组**：64位
  - **密文分组**：64位
  - **密钥长度**：64位，有效长度56位，8位用于奇偶检验。
  - **解密**：密钥顺序与加密相反
- **攻击（对加密系统）**：
  - **已知明文攻击**：攻击者拥有部分密文和对应的明文，根据算法寻找密钥
    选择明文攻击：有选择地使用任意明文和与之对应的密文信息，根据算法寻找密钥
    选择密文攻击：具有CPA的能力之外，还可以有选择地使用密文和与之对应的明文信息，根据算法寻找密钥
- **其他对称算法（不说分组都是64位）**：
  1. **3DES：**算法开销太大
  2. AES：分组128位，密钥长度128、192、256
  3. **IDEA：**密钥长度128位
  4. **Blowfish：**密钥长度可变
  5. **RC5：**其中**RC4是流密码算法**
  6. **CAST-128：**密钥长度可变，40~128
- **分组密码工作模式：**

  ①电码本模式(ECB)

  ②密码分组链接模式(CBC)

  ③密码反馈模式(CFB)

  ④输出反馈模式(OFB)

  ⑤计数器模式(CTR)

非对称密码

- **分类：**①RSA ②ELGamal ③椭圆曲线 ④Diffie-Hellman密钥交换

- **DH密钥交换：**

  任意受到中间人攻击。

数字签名与认证

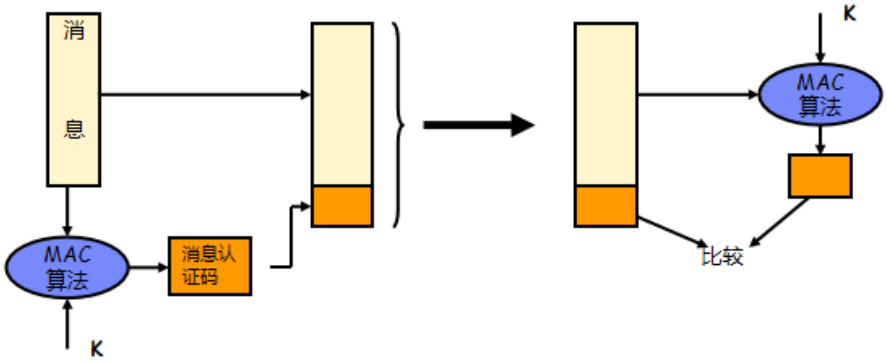
- **数字签名**
  - **目的：**保证消息的不可否认性。
  - **方法：**

    利用私钥生成签名，而用公钥验证签名。

    1. **签名算法：**利用私钥生成签名，称消息m的签名为sig（m），然后将（m，sig（m））发给接收方
    2. **验证算法：**利用签名者的公钥对sig（m）进行解密，如果解密输出与m一致，则为合法数据。
- **消息认证**
  - **目的：**保证消息的完整性。（数字签名的实例）
  - **方法：**

    1. **直接加密**
       - **对称密钥加密**
       - **公钥密码中的私钥加密**——速度太慢
    2. **消息认证码（MAC）**

       是对信源消息的一个编码函数, 以消息和密钥作为输入，定长输出

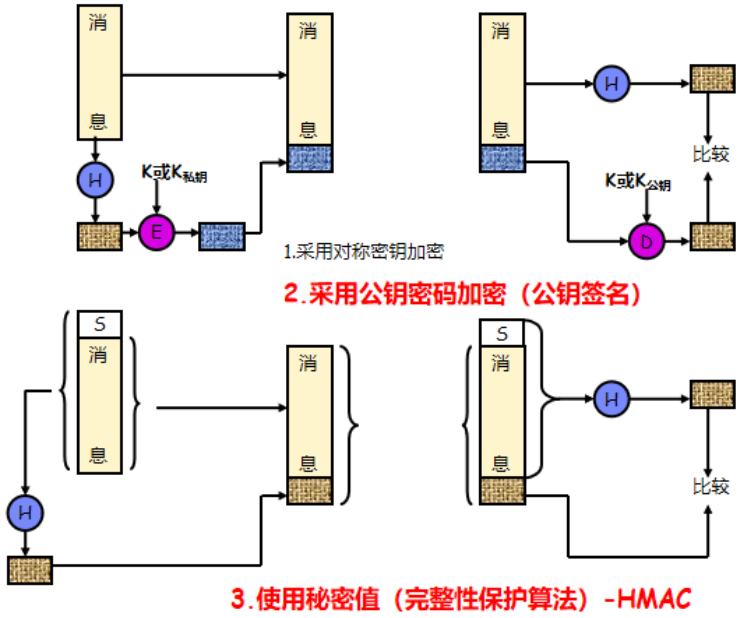
       

    3. [散列函数+签名]

       - **散列函数定义：**从很长的报文中计算出固定长度的比特串。这种散列函数称为报文摘要，用于**消息完整性检验。**
       - **散列函数特性：**

         ① **弱碰撞抵抗（弱hash函数）**：对于任意给定的数据X，寻找满足H(Y)=H(X),Y≠X,在计算上是不可行的。

         ② **强碰撞抵抗（强hash函数）**：寻找满足H(Y)=H(X)的(X,Y)对，在计算上是不可行的。
       - **标准：**MD5（**128比特**），SHA-1（**160比特**），SHA-2（包括6种）



加密和MAC顺序：

一般采用发送（E(M),HMAC(M)）而不是E（M,HMAC(M)）,原因是加密操作开销是HMAC的两个数量级，当消息出错时，前者只需要直接判断HMAC就行，消耗由101变为1。

Q：IPSec AH/ESP、SSL/TLS 中普遍不采用签名来提供完整性和数据源认证?

A：因为之前已经协商了对称密钥了，使用HMAC是基于对称密钥的更快，开销小。

网络安全标准组织

- **国际组织：**
    1. **国际标准化组织**（ISO）
    2. 国际电信联盟（ITU-T）
    3. **国际电工委员会**（IEC）
    4. 互联网协会 (ISOC)
        - 因特网工程任务组 (IETF)
        - 互联网研究任务组 (IRTF)
    5. **欧洲计算机制造商协会**（ECMA）
    6. **美国国家标准技术研究所(NIST)**
    7. **美国国家计算机安全中心**（NCSC）
    8. **美国国防部**（DOD）

公钥基础设施PKI

概述

定义

PKI是使用**非对称**密码算法技术，实现并提供**安全服务**的通用性**安全基础设施。**是为电子商务开展提供安全基础平台的**技术和规范。**能为所有网络应用提供**密钥和证书管理。**

简言之，PKI是用于管理数字证书和公钥-私钥对的技术框架。

值得注意的是，PKI是技术框架，后续学到的IPsec，SSL/TLS都是利用PKI框架来实现安全通信的具体协议。

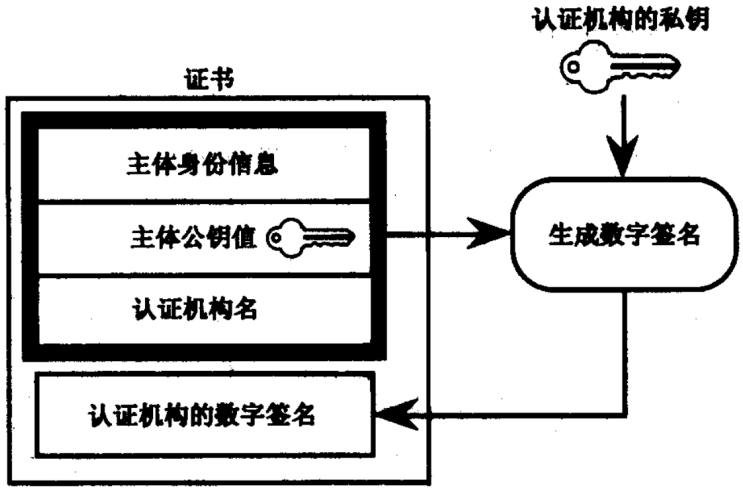
功能

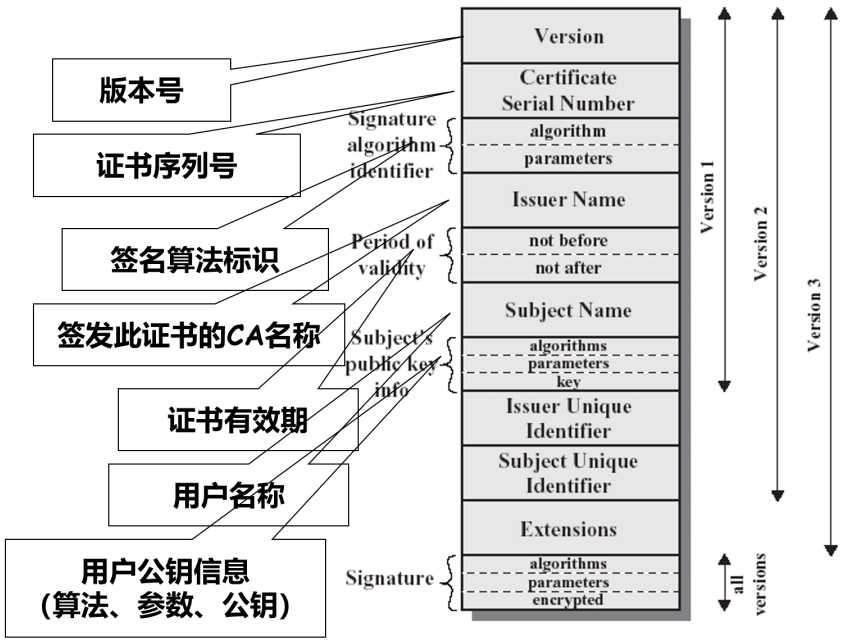
通过可信CA的签名实现**主体身份**和**主体公钥**之间的**绑定关系**

证书结构

- **最基本结构：**① 主体公钥值 ②主体身份信息 ③认证机构名 ④认证机构数字签名
- **其他结构：**版本号、有效期、序列号、用户名、签名算法标识等

证书的基本结构





产生背景（帮助理解）

PKI（公钥基础设施）的背景可以追溯到数字加密和网络通信的发展。在早期的网络通信中，安全性和身份验证是一个严重的问题。传统的加密技术使用相同的密钥进行加密和解密，这需要在通信双方之间提前共享密钥，**存在密钥分发和管理的困难**。此外，没有有效的机制来验证通信双方的身份，容易受到**中间人攻击和数据篡改**的威胁。

下面是一个PKI的实例：

浏览器与网站之间的安全通信，使用HTTPS协议

1. **浏览器请求安全连接：**浏览器输入HTTPS开头网址，浏览器会向服务器发送一个连接请求，要求建立一个安全的连接。
2. **服务器发送数字证书：**作为响应，服务器会发送其数字证书给浏览器。这个证书包含了**网站的公钥、CA的签名（私钥）、网站的身份信息、认证机构名。**
3. **浏览器验证数字证书：**浏览器接收到数字证书后，会执行以下操作来验证证书的有效性，若证书验证成功，浏览器就会信任该网站：
    1. **验证 CA 的签名**：浏览器会检查数字证书上的签名是否由其信任的 CA 签发。浏览器内置了许多受信任的 CA 的根证书，用于验证签名。
    2. **检查证书的有效性**：浏览器会检查证书是否在有效期内，且未被撤销。
4. **加密通信：**一旦浏览器验证了服务器的证书，它会使用服务器的**公钥**来加密一个随机生成的**对称加密密钥**，即**数字信封**，并将这个加密后的密钥发送给服务器。
5. **安全通信：**之后，浏览器和服务器就可以使用对称加密密钥安全地传输数据了。

假设没有PKI：

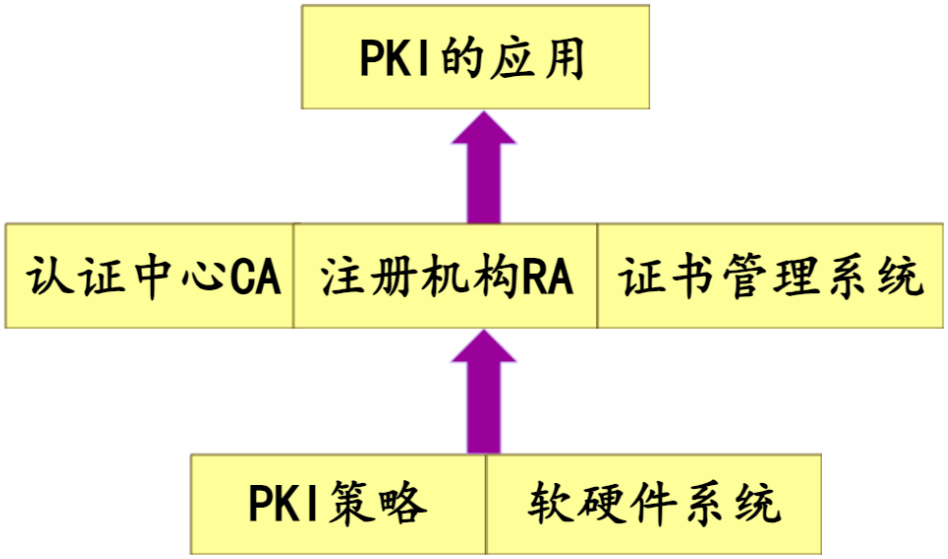
服务器会受到中间人攻击，例如：

服务器没有CA颁发的证书，浏览器无法验证服务器身份，无法保证用户试图访问的服务器不是一个冒充的服务器。进一步地，服务器发送的公钥可能是中间人伪造的，这就导致通信缺乏保障。

组成（基本组成包括：CA、RA、证书库）

1. **认证中心（CA）**：负责**签发**数字证书并管理证书的生命周期。
2. **注册机构（RA）**：协助CA执行**身份验证**，并进行证书请求的审批和管理。
3. **证书管理系统：**
    - **证书库**：集中**存储**已签发的数字证书和证书吊销列表（CRL），提供公共查询服务。
    - **密钥备份及恢复系统**：用于备份和恢复密钥，确保密钥的安全性和可用性。
        - **签名密钥对：**签名私钥相当于日常生活中的印章效力，为保证其唯一性、抗否认性，**签名私钥不作备份。**签名密钥的生命期较长。
    - **加密密钥对：**加密密钥通常用于分发会话密钥，为防止密钥丢失时无法解密数据，**解密密钥应进行备份。**这种密钥应频繁更换。
    - **证书作废处理系统**：记录和管理已经被吊销的证书。
    - **自动密钥更新**：确保证书及其关联的密钥在失效时自动更新。
    - **密钥历史档案**：存储用户的密钥历史记录，用于管理密钥更新和维护密钥历史记录。

PKI的组成



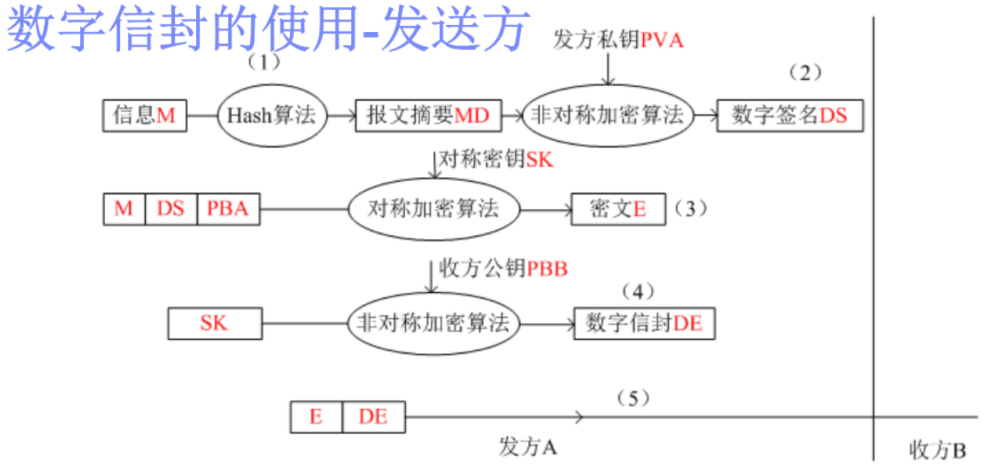
版本更替

- v2：引入主体和签发者唯一标识符的概念（选项）
- v3：支持扩展（属性）

PKI密码技术

[数字信封]

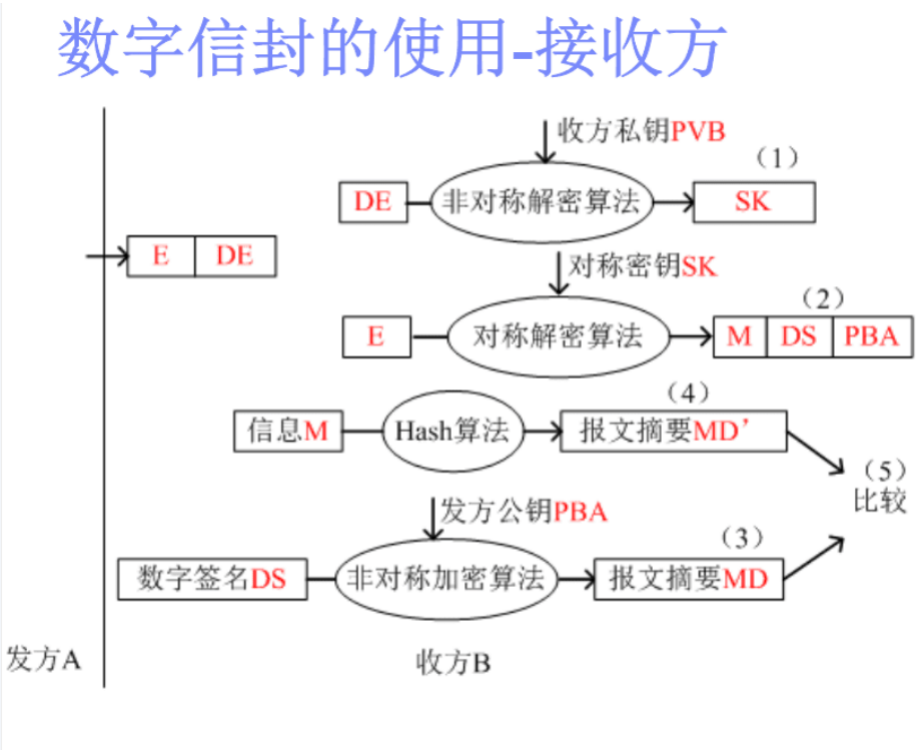
- **概述：**①发送端：用接收端的**公钥**，将通信密钥（**对称密钥**）加密，生成**数字信封。**②接收端：用私钥打开数字信封，获取对称密钥SK。
- **使用：**
  - **发送方：（客户端）**
    1. **信息hash：**生成数据摘要，防止传输过程数据被篡改。
    2. **签名摘要：**①信息长度可能非常长，直接签名非常耗时，因此签名摘要。②签名目的是防止中间人攻击，以保证信息的可靠性。③抗否认。
    3. **明文加密：**将明文、2的数字签名、发送方证书上的公钥，加密生成密文。发送方的公钥用于接收方验证签名。
    4. **生成数字信封：**使用证书中的**公钥**加密**对称密钥**，生成数字信封。
    5. **发送密文和信封**



考虑接收端开销的角度，这里是有错误的，不能够先签名后加密，否则接收方每次都需要解密才能判断有没有出错。

建议密文签名和密文分开发过去。

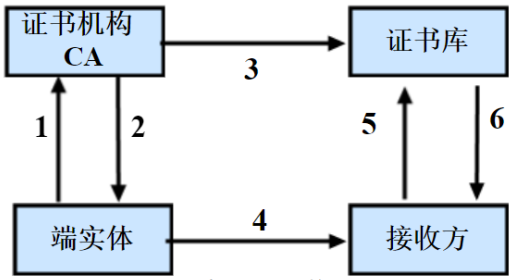
- **接收方：（服务器）**
    1. **解密信封：**私钥解密信封获取对称密钥。
    2. **解密密文：**对称密钥解密密文，获取明文，发方公钥，签名摘要
3. **验证摘要：**使用发方公钥解密数字摘要。
4. **验证信息：**将明文hash值与摘要比对，判断是否相等，相对则接受。



功能操作

PKI功能操作概述

PKI的运行操作主要经过六个步骤：



① 端实体向证书机构（CA）提出数字证书申请；

② CA验明端实体身份，并签发数字证书；

③ CA将证书公布到证书库中；

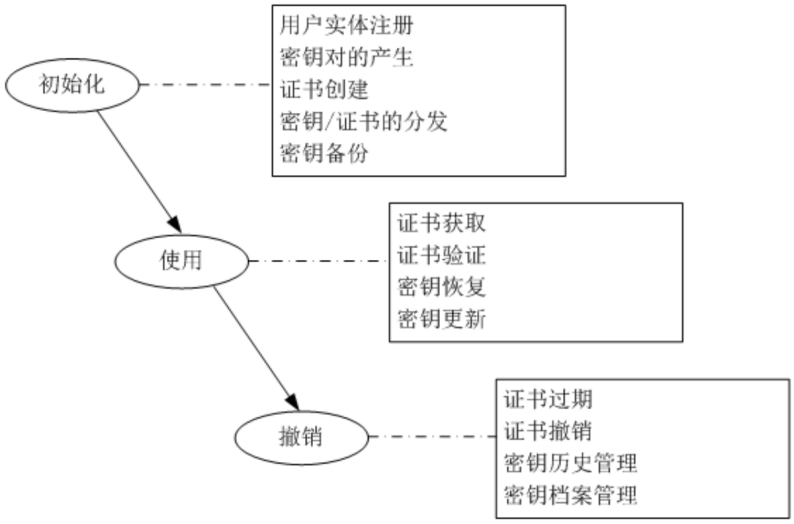
④ 假设为电子邮件应用，端实体对电子邮件数字签名作为发送认证，确保邮件完整性，不可否认性，并发送给接收方。

⑤ 接收方接收邮件，用端实体的公钥验证数字签名，并到证书库查明端实体证书的状态和有效性；

⑥ 证书库返回证书检查结果（CRL检查等）。

证书操作

证书的初始化、使用和撤销



- **初始化**：终端实体在使用PKI提供服务之前，必须先初始化。
- **使用**：
  - **证书获取**：
    - 发送者发送签名时，附加发送自己的证书。
    - 单独发送证书
    - 从访问发布证书的目录服务器获得
    - 从其他公共站点的共享位置获得
  - **证书验证**：（三点，见下述Q&A）
  - **密钥恢复**：终端实体丢失**加密密钥**可以恢复
  - **密钥更新**：当一个合法的密钥对将过期时，进行新的公/私钥的自动产生和相应证书的颁发
- **过期**：证书生命周期自然结束。

- **撤销：**

  撤销场景：

  - **证书撤销**：私钥泄露、关系终止、CA私钥泄露等

  撤销阶段：

  - **撤销：**证书在过期之前被撤销（非过期）
  - **存档：**维持一个**CRL**和有关历史证书的记录，以便被**过期的密钥资料**所加密的数据能够被**解密**
  - **审计：**出于对密钥历史恢复、审计和解决争议的考虑所进行的密钥资料的安全**第三方长期储存**
- **验证：**（三个方面，不要求顺序）
  - 验证签名
  - 验证CRL列表，查看证书是否撤销
  - 查看字段是否符合，例如有效期、用户ID、签名是否是加密密钥对等

Q.如何产生证书？（超级关键）

A.（不要回答用户提交申请RA审核CAxxx这些，问的是产生证书的流程，对应的是**对签名的描述**！）

CA用自己私钥签名证书摘要附在证书之后

Q.获得证书的流程

A.用户<->RA<->CA（略，见ppt）

IPSec：AH和ESP

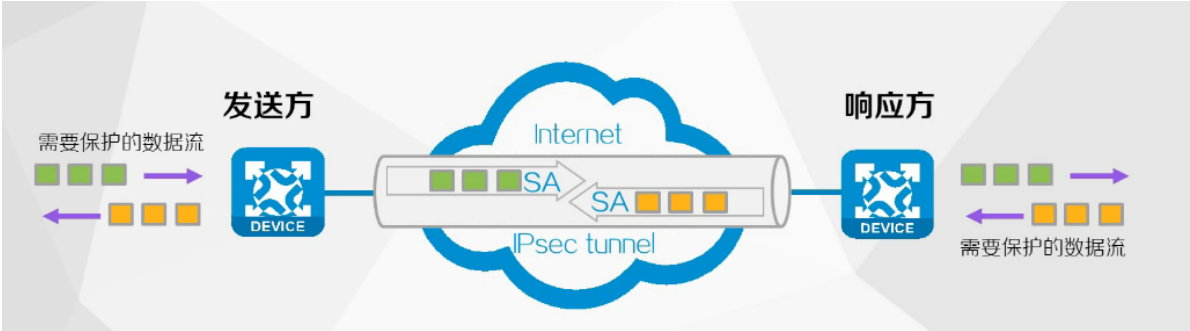
IPSec背景

- **仅有SSL/TLS的缺陷**

  IPSec提供网络层端到端的保护，SSL/TLS在应用层起作用，没有网络层的保护会出现很多问题，例如：

  - **元数据泄露**：即使应用数据被加密，但如果网络层不加密，关于通信本身的元数据（如IP地址、端口号、协议类型等）仍然可以被窃听和分析。攻击者可以利用这些信息来识别网络流量的模式、确定通信双方、甚至推断出正在进行的活动类型。
  - **内部网络安全：**基于元数据泄露，攻击者可能嗅探到服务器IP地址等相关讯息，进而攻击服务器和系统，获取权限。
- **IPSec诞生**

IPSec概述



定义——1组协议的集合

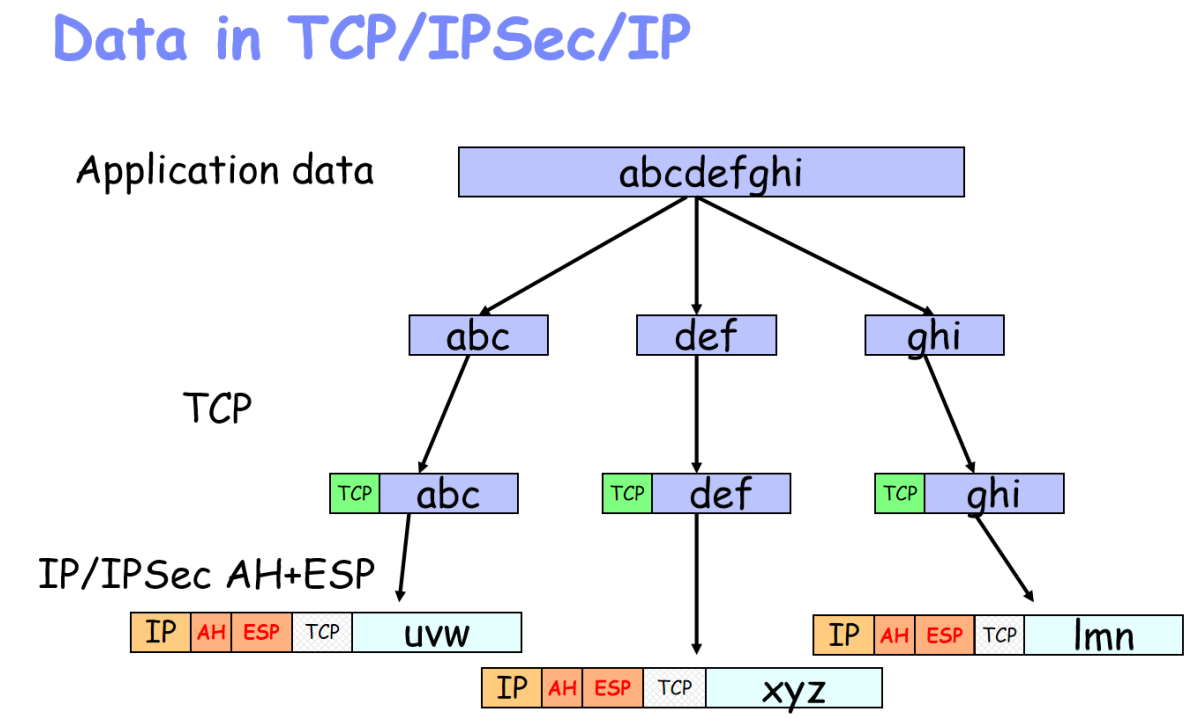
IPSec是一组协议的集合，用于在IP网络层上保障通信的安全。它旨在通过加密和认证手段，保护网络数据包在互联网或其他网络传输过程中的安全，从而确保数据的完整性、认证以及机密性。IPsec广泛应用于虚拟私人网络（VPN）中，为远程用户提供安全的连接到企业网络。

IPSec是在网络层每个节点（路由器）的保护，SSL是在传输层端到端的保护。

安全协议——AH、ESP

- **AH（认证头）**：
  - 无连接的**完整性、数据源认证**和**抗重放保护。**
  - **不提供保密性**
  - 使用MAC对IP认证
- **ESP（封装安全载荷）**：
  - **数据保密、无连接完整性、抗重播服务**
  - 对称密码加密
  - 使用MAC认证

网络安全协议



安全算法

- **加密算法：**3DES、AES、SM4
- **认证算法：**MD5、SHA1、SM3
- **密钥协商算法：**DH

安全组合（SA）

背景需求

认证、加密算法及参数、密钥、使用的序列号等。

SA概述

为使通信**双方的认证、加密算法、密钥**的**一致**，而相互建立的**联系**被称为**安全组合**或**安全关联。**

SA是单向的（目的端确定），因为数据传输是双向的，不同方向可能策略不同。因此双向通信要建立两个SA。

表现形式：SAD中的一条条记录。

简单来说，SA就是一组规则条目，包含了以下内容：

- 协商的密钥、序列号、加密算法、生存周期

SA标识——SPI，IP，Protocal

SA由一个**三元组**唯一地标识，该三元组为**安全参数索引SPI、目的IP地址（用于输出处理目的）、协议（AH或ESP）**

因此只需要SPI、IP、Protocal组合即可确定SA，以确定相关参数。

- **SPI：**是为了**唯一标识SA**而生成的**32位整数**，由目的端确立。包含在AH头标和ESP头标中。

  注：SPI由目的端确定，这是因为：

  发送端需要先根据规则在SPD匹配，匹配后才能根据指针去SAD分配到SA，这样才能给数据包分配SPI。如果SPI由发送端确定，那么发送端选择的SPI可能会与目的端的冲突（不同发送端协商了同一个SPI）。由目的端选择则不会（目的地址不一样）。

SAD——安全关联数据库

SA通过IKE协商，协商完毕后在SAD中存储SA参数。

SPD——安全策略数据库

用于确定策略，定义了哪些流量需要通过IPSec进行处理。

注：接收端可以查SPD（可选），防止发送端乱发（譬如数据需要加密，但是发送端用了认证的SA，不查SPD就没法判断了）。

Q.IPSec保护一般需要几个SA?

A.四个（单向2，加密和完整性保护2，这里首先一般不用ESP认证，同时一般都需要认证和加密、只加密的情况下可以去减）

认证头标AH

AH概述

AH是IPSec框架中的一部分，它为IP数据包提供无连接的源认证、数据完整性和抗重放保护服务，但不提供保密性服务。AH使用消息认证码（MAC）对IP进行认证。

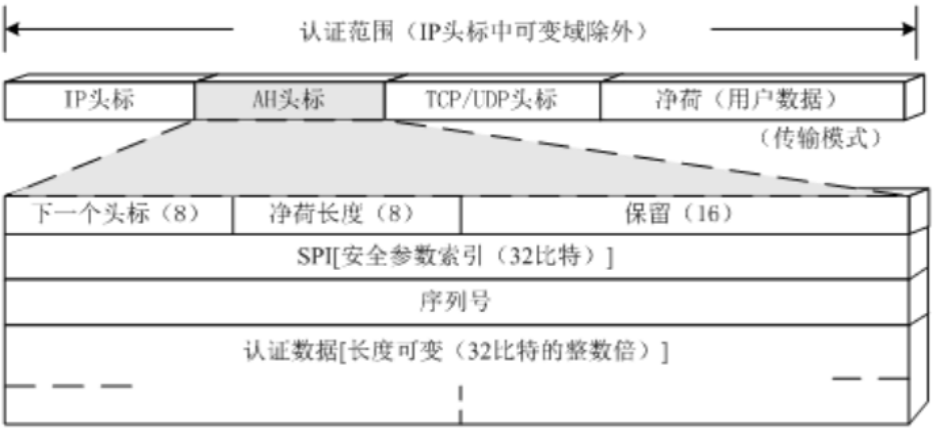
AH作用

1. **完整性：**使用MAC进行完整性检查。
2. **认证：**使用MAC进行认证。
3. **抗重放：**利用序列号单调增来抗重放。

AH组成

1. **安全参数索引SPI：**即SA中的SPI。
2. **序列号：**单调增加，利用其抵抗重发攻击。
3. **认证数据：**存放整个IP数据包的MAC值，长度为32的整数倍。MAC过程中，数据包的可变域取0，如AH的认证数据部分.

   AH使用MAC进行认证，这意味着在认证数据部分包含对AH头和载荷进行的MAC，MAC过程的密钥通过IKE分发。



封装安全载荷头标ESP

ESP概述

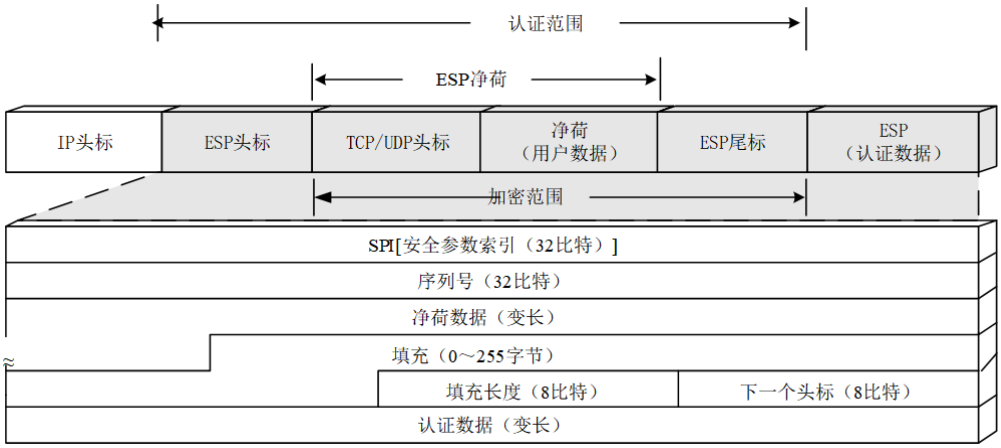
ESP是IPsec框架中的一部分，用于提供数据保密、无连接完整性（可选，不覆盖IP头标）、数据源认证、抗重放攻击。

- 填充域：为了对齐待加密数据、掩盖实际长度而根据需要将其填充到4字节边界。

ESP作用

1. **保密性**：利用对称密码。
2. **完整性**：使用MAC
3. **认证**：使用MAC
4. **抗重放**：使用序列号

ESP组成



封装模式

- **传输模式**：为**IP载荷**提供认证、完整性和机密性。
- **隧道模式**：保护**整个IP分组**，提供认证、完整性和机密性。

  （这就是传输和隧道的区别！）

补充：

①认证：AH认证全部，ESP只认证从ESP头标到ESP尾标。

②加密：AH不加密，ESP只加密ESP头标后到ESP尾标。

因此只有ESP是不能够认证IP头的，需要AH结合才行。

NAT冲突

IPSec -> SNAT（修改源地址/端口） -> DNAT（修改目的地址/端口） ->IPSec

- **AH认证：**地址的修改使得接收端的AH认证失败
- **ESP加密：**上层端口号信息的ESP加密，使得端口无法被得知，无法进行NAT-PT（目的地址 + 端口[加密] <-> 内网地址+端口）
- **ESP认证：**上层TCP/UDP中校验和计算涉及伪头标，包括IP地址和端口，通过ESP认证，校验和字段不能被修改，上层会校验验证失败

解决方案：（可能会在考试时现场手搓解决办法）

针对ESP的问题，在ESP头标前插入一个UDP头标（端口得知，就可以映射了）

IPSec全流程实例

背景介绍

假设公司的两个分支机构A和B，希望通过互联网安全地交换数据。它们决定使用IPSec来加密和认证数据。

步骤1——设置SPD规则

A的网络管理员配置SPD，所有目标为B的IP地址的HTTP流量必须通过IPSec保护。具体规则如下：

- 选择符：
  - **源地址**：10.1.1.0/24
  - **目的地址**：10.2.2.0/24
  - **源端口号**：任意（HTTP客户端通常使用临时端口）
  - **目的端口号**：80
  - **协议**：TCP
- 条目：
  - **策略**：加载IPSec。
  - **IPSec协议**：选择ESP
  - **操作模式**：传输模式，加密HTTP流量的有效负载。
  - **算法**：使用AES算法加密，SHA-256认证。
  - 对外出处理，**应在SPD中查找指向SAD中SA的指针**

这样，A中任何主机通过HTTP访问B的服务器时，其数据包会经过SPD检查。匹配上述规则后，系统会根据指向SA的指针，在SAD中寻找对应规则的SA。

若没找到SA，则使用IKE建立SA。这个具体实例中，是A和B的第一次通信，SA还没建立。

步骤2——IKE协议协商SA

①建立安全通道（IKE SA）

②建立IPSec SA，**结果如下：**

- 使用**ESP协议**加密数据流。
- 选择**AES-256**作为加密算法，密钥为666...（共256bits）。
- 使用**SHA-256**作为完整性检查算法。
- 为特定的数据流**分配SPI**。比如0x12345678，用于标识经过处理的数据包属于哪个SA。
- 设置SA的**有效期限**。例如1小时，之后需要重新协商。

IPSec SA建立完成之后，A、B将该SA加入自己的SAD中。

步骤3——发送数据包

步骤2中，根据SPD的策略，在SAD中找到了符合策略的具体SA。 **（SPD存储保护策略和保护方式，SAD存储执行策略的具体参数）**

系统会使用该SA的参数来为数据包进行加密，具体内容为：

1. **匹配SA规则：**数据包到达网关，网关中的SPD对数据包进行匹配，根据选择符匹配需要使用的SA规则。
2. **具体SA加密：**系统对SPD对应的SA指针找到具体的SA，利用SA对数据包进行加密。
    - **ESP处理**：SA提供加密参数，而ESP负责具体执行加密过程。
        - **SPI**：在此例中，SPI被设定为0x12345678。
        - **序列号**：用于抗重放攻击，每发送一个数据包，序列号就增加。
        - **加密**：通过SA的加密参数和算法，来加密数据。
        - **ESP尾部**：包含必要的填充（如果加密算法需要），填充长度，和下一个头部字段（指明封装前的原始IP数据包的协议类型，例如TCP）。

步骤4——接受数据包

根据收到数据包的三元组（SPI，目的地址，协议），B可以在自己的SAD查找对应的SA参数，从而对数据包解密。具体内容为：

1. **匹配SA参数：**通过三元组获取在SAD中对应的SA，使用该SA解密。若没找到SA则丢弃。
2. **策略验证：**在SPD中寻找分组对应的策略，根据策略检查该分组是否满足IPSec处理要求。（确保数据包是被授权的）
3. **检查序列号：**序列号不在范围内则丢弃。
4. **认证：**有认证需求则对数据包进行认证。
5. **解密**

源端/目的端处理流程

使用ESP加密、AH认证情况的源端/目的端处理流程（两个SA都是隧道模式）。

源端外出

1. 根据选择符（五元组）查找**安全策略数据库**（SPD）获取策略。策略显示分组需要进行IPSec处理，使用ESP加密、AH签名。若到目的主机的SA已经建立，对应策略将指向**外出SA数据库**（SAD）对应的用于AH和ESP的两个SA。若SA还未建立，IPSec将调用**IKE**协商一个SA，并将其连接到SPD条目上。
2. 产生或增加序列号，当一个新的SA建立时，序列号计数器初始化为0，以后每发一个分组，序列号加1
3. 根据加密SA，加密整个IP分组，SA指明加密算法，一般采用对称密码算法
4. 根据认证SA，添加计算ICV（完整性校验值）
5. 封装新IP头，转发分组

目的端进入

1. 根据选择符进入SPD查找一条与选择符匹配的策略，**检查策略是否相符**，策略不相符则丢弃，策略相符或无需IPSec处理则放行。
2. 根据IP分组头中的SPI值、目的IP地址以及IPSec协议在进入的SA数据库中查找SA，如果查找失败，则抛弃该分组，并记录事件。
3. 检查序列号，确定是否为重放分组
4. 使用认证SA指定的MAC算法计算ICV，并与认证数据域中的ICV比较，如果两值不同，则抛弃分组
5. 使用加密SA解密分组

IPSec：IKE（主要注意原理，不要拘泥于实现细节）

针对IKE的攻击：

IKEv1：在第一步给出较弱列表降低安全等级（第三步认证把list哈希发过来即可解决）

协商避免从0开始（避免了开销但是会带来安全问题）

IKE背景

见课本243页。

DH密钥交换的不足：

1. 不能标识各方身份信息
2. 易受中间人攻击
3. 计算密集型（DoS攻击）

Diffie-Hellman 算法有两个优点：

- 仅在需要时生成密钥，而不需要长时间地存储密钥，从而增加了安全性。
- 交换仅需要全局参数达成一致，不需要其他预先存在的基础设施。

然而，正如[HUIT98]中所指出的那样，Diffie-Hellman 也有很多弱点：

- 它没有提供标识各方身份的任何信息。
- 它易受中间人攻击。在中间人攻击中，第三方 C 当与 A 通信时冒充 B，当与 B 通信时冒充 A。这样 A、B 都与 C 协商了密钥，第三方 C 能窃听和传递流。中间人攻击过程如下：

（1）B 发送其公钥 Y_B 给 A（如图 3.13 所示）。

（2）攻击者 E 截获此消息，保存 B 的公钥并用 B 的用户标识和 E 的公钥 Y_E 向 A 发送消息，用此方式发送，该消息好像来自于 B 的主机系统。A 接收到 E 的消息，并存储了带着 B 用户标识的 E 的公钥。同理，E 给 B 发送带有 E 的公钥声称来自 A 的消息。

（3）B 基于 B 的私钥和 E 的公钥 Y_E 计算一个密钥 K_1；A 基于 A 的私钥和 E 的公钥 Y_E 计算一个密钥 K_2；E 使用其私钥 X_E 和 Y_B 计算 K_1，使用 X_E 和 Y_A 计算 K_2。

（4）从现在起，E 能转发从 A 到 B 的消息和从 B 到 A 的消息，用适当的方式适当地改变途中的密文，而 A 和 B 都不知道他们正在和 E 共享通信。

- 算法是计算密集型的。后果是，该算法易受拥塞攻击。在这种攻击中攻击者申请很多密钥。这样遭到攻击的主机就会消耗大量的计算资源做无用的模幂运算，而不是

IKE弥补了DH密钥交换的不足：

IKE 密钥确定的特性。IKE 密钥确定算法有如下 5 个重要的特性。

（1）它运用了一种称为 Cookie 的机制来防止拥塞攻击。

（2）它允许双方协商得到一个组，在本质上，这就是详细列出 Diffie-Hellman 密钥交换的全局参数。

（3）它使用随机数来阻止重放攻击。

（4）它允许交换 Diffie-Hellman 的公钥值。

（5）它认证 Diffie-Hellman 交换，以此阻止中间人攻击。

我们已经讨论了 Diffie-Hellman 算法，下面逐个讨论这些剩下的问题。首先，考虑拥塞攻击的问题。在此攻击中，对手伪造合法用户的源地址并向受害者发送一个 Diffie- Hellman 公钥。受害者执行模幂运算来计算密钥。重复这类消息可以利用无用工作拥塞受害者的系统。**Cookie 交换**要求各方在初始消息中发送一个伪随机数 Cookie，此消息要得到对方的确认。此确认必须在 Diffie-Hellman 密钥交换的第一条消息中重复。如果源地址被伪造，那么攻击者就不会收到应答。这样，攻击者仅能让用户产生应答而不进行 Diffie-Hellman 计算。

ISAKMP 规定 Cookie 的产生必须满足三个基本要求：

（1）Cookie 必须依赖于特定的通信方，这能防止攻击者得到一个正在使用真正的 IP 地址和 UDP 端口的 Cookie，因此也就无法用该 Cookie 向目标主机发送大量的来自随机选取的 IP 地址和端口号的请求，以达到浪费主机资源的目的。

（2）除了发起实体以外的任何实体都不可能产生被它承认的 Cookie。这就意味着发起实体在产生和验证 Cookie 时，要使用本地的秘密信息，并且，根据任何特定的 Cookie 都不可能推断出该秘密信息。实现这个要求的目的在于发起实体不需要保存它发行的 Cookie 的副本，仅在必要时能验证收到的 Cookie 应答，所以就降低了泄露的可能性。

（3）Cookie 的产生和验证方法必须很快，以阻止企图占用处理器资源的攻击。

推荐的创建 Cookie 的方法是根据 IP 的源地址、目的地址、UDP 的源端口、目的地端口和本地产生的秘密值来进行快速散列运算（比如 MD5）。

IKE定义

IKE是**因特网密钥交换协议**，是一个以受保护方式**动态协商IPsec SA**的协议。运行在应用层，基于UDP。

目的：基于长期密钥协商得到短期密钥

IKE功能

使用某种**长期密钥**进行双向认证并建立**会话密钥**。

- **协商内容**：通信参数，安全特性等
- **长期密钥**：
  - 共享秘密密钥
  - 加密密钥
  - 签名密钥

IKEv1流程

阶段一（建立ISAKMP SA 双向）

步骤：

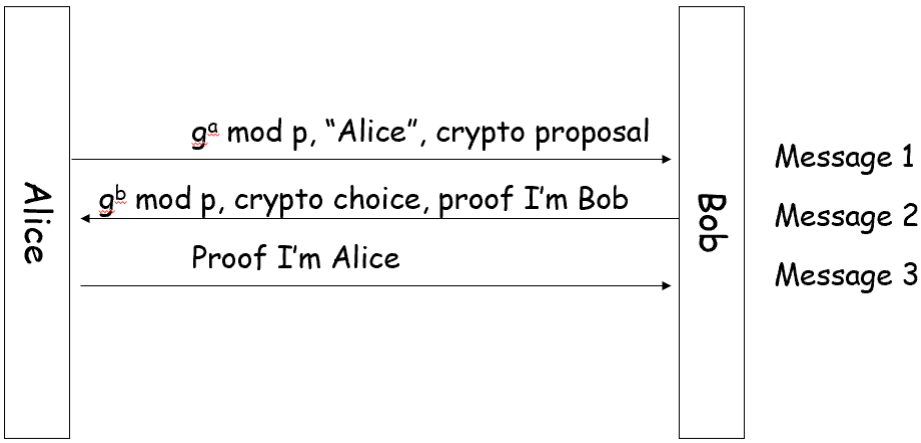
① 协商安全参数

② DH密钥交换

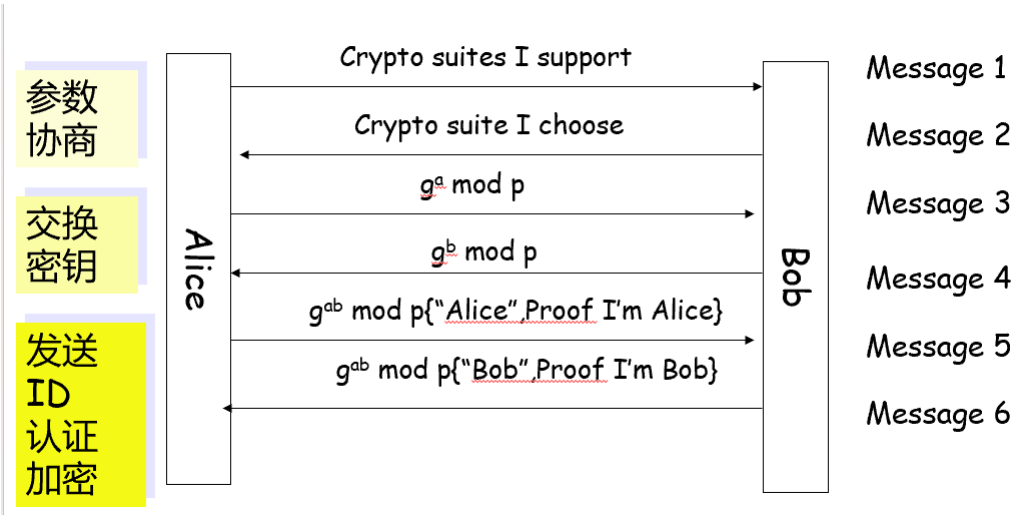
③ 认证实体

交换模式（2种）：

- **野蛮模式——简化规程和提高处理效率**



- **主模式**



每次传输都需要附加IKE Header，其中包含CKY-I/CKY-R（cookie），用于防止拥塞攻击。

①参数协商：

发送方（Initiator）发送ISAKMP SA，这是一个包含加密算法和认证算法列表。接收方从中选择算法。（攻击：把破解难度大的算法剔除）

②密钥交换：

采用DH密钥交换

③实体认证：

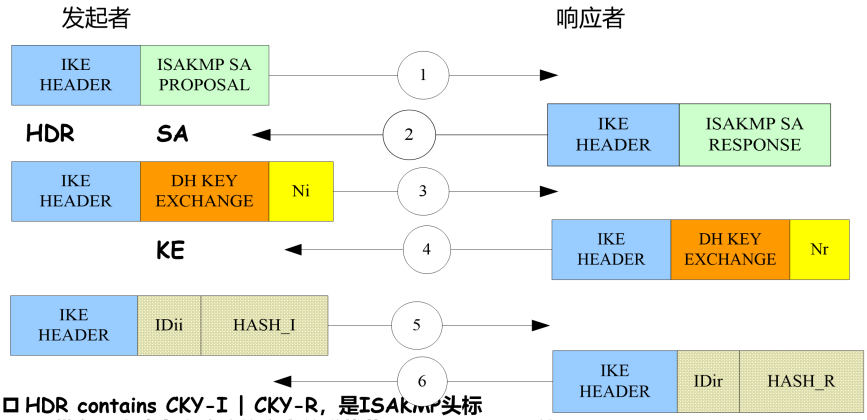
- **预共享密钥：**②中加上随机数Nx。③中采用HMAC加密HASH_I/R，其中 **SKEYID=PRF(preshared key, Ni|Nr)**
- **基于数字签名：**②中加上随机数Nx。③中采用证书+数字签名加密HASH_I/R，其中 **SKEYID=PRF(g^ir, Ni|Nr)**
- **基于公钥加密：**②中直接用对方公钥加密ID（防止伪造身份）和Nx。③中直接发送HASH_I/R（因为Nx公钥加密可作为PSK）

  补充，这里**必须要加密ID以防止中间人攻击**：

  假设Eve能够让A向其发起通信。①收到A的第一条消息后，Eve通过伪造cookie为CKY_A，向B发起通信。②Eve将含CKY_B的头发送给Na，并从A获得Na。③Eve将Na发送给B，并从B获得用A公钥加密的Nb。④Eve把加密的Nb发送给A，A计算hash_I，得到的结果经过Eve转发给B，此hash_I中，所有信息均无误（Na，Nb，CKY_A，CKY_B，IDiA，SKEYID），从而成功伪造身份。
- **公钥加密改进：**原先采用四次公钥加/解密操作，耗费计算资源。改进方案：②中使用对方公钥加密Nx，而其余信息（包含证书CERT）使用DH Ki/Kr加密。

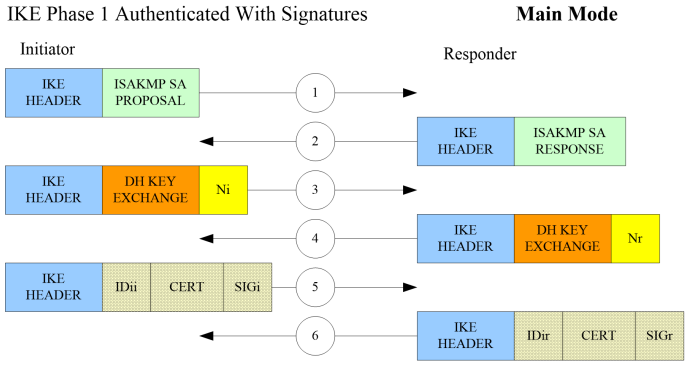
  补充，中间人因为无法破解Kb，从而无法修改IDib，进而无法让A发送HASH_A。

主模式：用预共享密钥认证



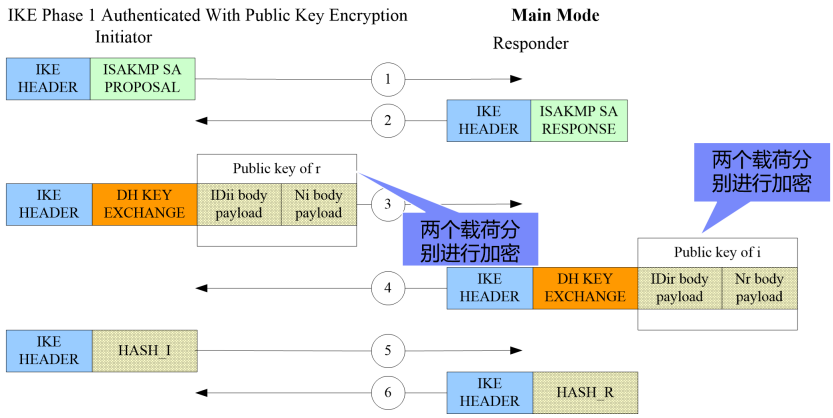
- HDR contains CKY-I | CKY-R，是ISAKMP头标
- SA带有一个或多个建议的安全关联载荷。Ni/Nr是nonce值
- KE = g^i (Initiator) or g^r (Responder)，是密钥交换载荷
- IDii/IDir是发起者/响应者的标识载荷
- HASH是杂凑载荷

主模式：用数字签名认证

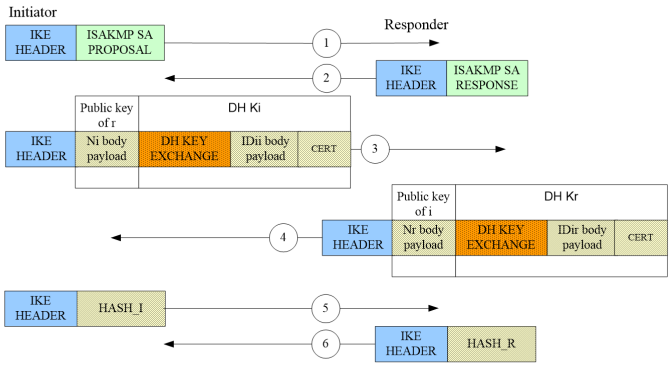


- HDR contains CKY-I | CKY-R
- KE = g^i (Initiator) or g^r (Responder)
- SIG_I/SIG_R = digital sig of HASH_I/HASH_R
- CERT是证书载荷

主模式：用公钥加密认证



主模式：改进的公钥加密认证



HDR contains CKY-I | CKY-R
KE = g^i (Initiator) or g^r (Responder)
Ki = prf(Ni, CKY-I), Kr = prf(Nr, CKY-R)

阶段二（建立IPSec SA 单向）

IKE SA

Child SA

SSL/TLS基本协议

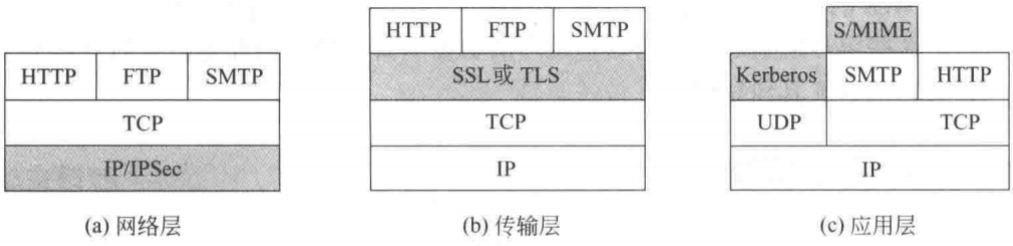
SSL背景

上世纪九十年代，IETF主导提出了IPSec规范。IPSec位于网络层，在网络层提供了安全策略，也可以提供Web安全，好处在于IPSec对上层透明。

然而提供安全的方法不止一种，为什么非得吊死在网络层上呢?

因此，同一时期（甚至比IPSec更早），有另一波人思考能否在传输层提供安全，在此背景下，Netscape公司于1994年提出了SSL 1.0（因被业内质疑有安全问题而未公开）。随后基于SSL，发展了TLS协议，目前TLS v1.3成为主流。

当然，还有存在于应用层之上的安全协议，以提供特定的安全服务。例如之前就学到的Kerberos协议。



(a) 网络层　　(b) 传输层　　(c) 应用层

图 6.1　TCP/IP 协议栈中的安全设施的相对位置

具体在应用中，TLS和IPSec很大一部分功能是重叠的，因此使用IPSec的大部分场景都可以使用TLS替代。

然而，TLS却应用的更加广泛，这是因为相比于IPSec，TLS有诸多优势，例如之前学到的IPSec在经过NAT处理时会遇到问题（ESP的作者反对使用NAT，因此在设计时没有解决该问题），为此提出了插入UDP头标的解决方法，但是TLS因为在传输层就避免了NAT的问题，自然更受欢迎。

具体优势可以见下表：

对比项	IPSec	SSL/TLS
适用环境	Site to Site	Client to Site
VPN层次	IP层	应用层
数据传输	隧道方式	SSL传输（TCP 443）
客户端	需专用软件	无需专用客户端软件
VPN部署	复杂	简单
远端维护管理	复杂，成本高	简单，成本低
部署成本	高	低
移动连接	不适用	适用
加密级别	高	高
复杂应用支持	容易	较容易
Intranet适用	较好	很好
Web应用	适合	非常适合
安全级别	高	高
NAT支持	不容易	容易
代理访问	不容易	容易
穿越防火墙	不容易	容易
供应商互操作	不容易	容易
即时消息传送、多播、视频会议及VoIP	容易	较复杂（采用L3VPN）
B/S应用	支持	支持
Legacy application	支持	支持
http应用	支持	支持
文件共享	支持	支持
无线设备	支持	支持
家庭、网吧、宾馆、其他企业接入	不好	很好，非常适用
代理级保护	不支持	支持
用户认证	不好	好
用户授权	有限	灵活
Web访问一次性认证	不支持	支持
URL级别的接入限制	不支持	支持
域名和IP地址的保护	不支持	很好

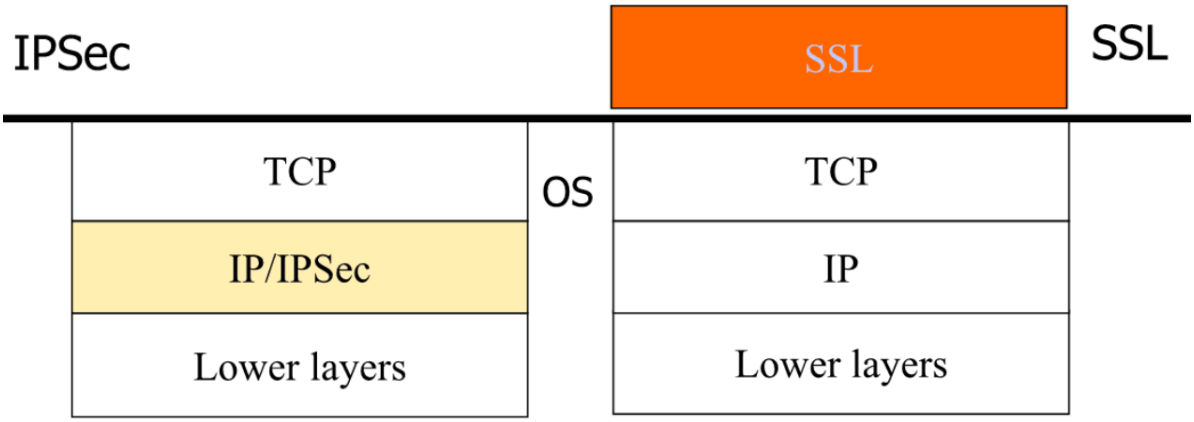
对比项	IPSec	SSL/TLS
根据用户访问的类别控制接入	不支持	支持
Session级保护	不支持	支持

SSL概述

定义

SSL是在TCP之上的两个端实体之间提供安全通道的协议。包括SSLv2/v3、TLS协议。

简单来说，IPsec，SSL都是PKI的应用实例。



功能

- **身份认证**
  - **客户对服务器身份认证：**证书
  - **服务器对客户身份认证：**用户名+密码，或证书
- **建立安全数据通道**

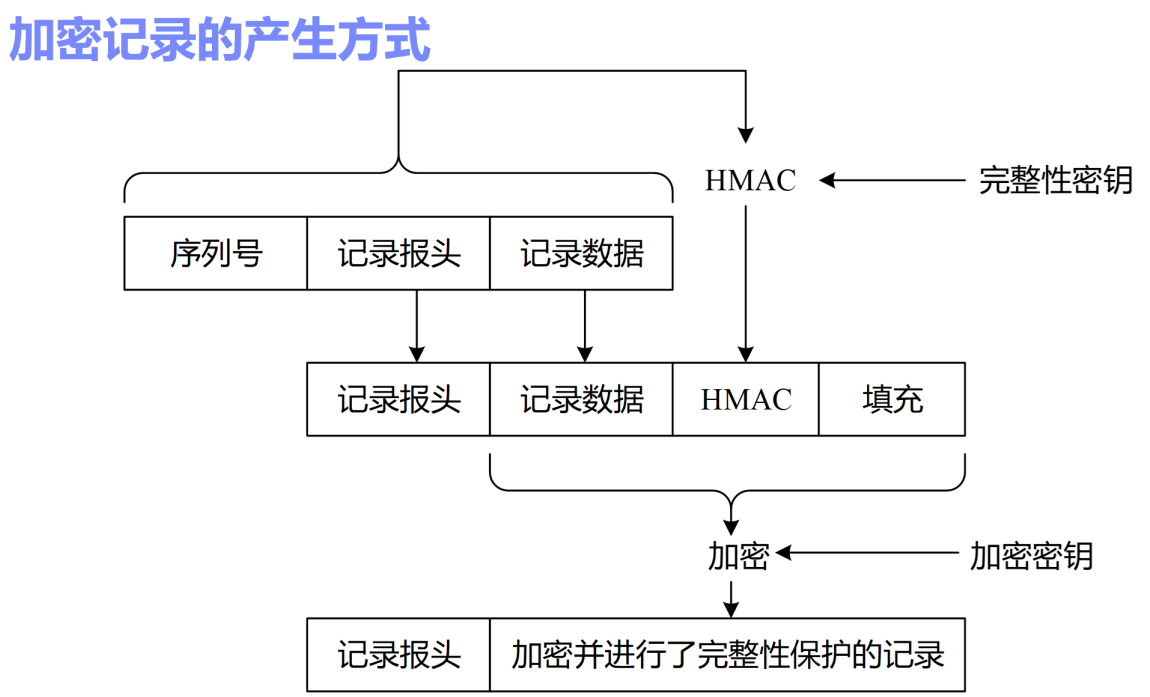
服务

1. **合法性认证**
2. **数据机密性**
3. **数据完整性**

协议构成

- **握手协议：**建立安全通道
- **告警协议：**指明安全错误
- **密码变化协议：**告知改变密码参数
- **记录协议：**封装上述三个协议+应用层数据

原理（数据处理过程和序列号使用）



注：序列号初始为0，随着数据包累加1，不需要进行传输！

组成

SSL协议分为两层：

- **底层**：**记录协议**
  1. 建立在TCP之上
  2. 用于上层协议封装
  3. 安全性：提供**保密性（对称加密）**和**完整性（HMAC）**
- **上层**：握手协议、密码变化协议、警告协议、用户数据
  - **握手协议**：安全连接之前交换安全信息
    1. 客户和服务器相互认证
    2. 协商加密算法和密钥
    3. 安全性：身份认证，协商密钥等

TLS

会话

假设你打开浏览器访问一个网上购物网站。你从打开网站开始，到你关闭浏览器窗口为止，这一段时间你和这个网站之间的互动就构成了一个会话。在这个过程中，你可能浏览商品、加入购物车、提交订单等等。

连接

在这个会话过程中，每当你点击一个链接或者提交一个表单，浏览器会和网站服务器之间建立一个连接。这就像你和朋友打电话，每一次通话就是一个连接。

会话重用

会话重用指的是在一次会话过程中，浏览器和服务器之间保持一个长时间的连接，而不是每次你点击一个链接都重新建立一个新的连接。比如你浏览商品、查看商品详情、加入购物车，这些操作都通过同一个连接完成，而不是每次点击都重新连接服务器。这就像你和朋友一直保持电话通话，不需要每次说一句话就挂断再拨打。

防火墙

防火墙概述

定义

防火墙是位于两个/多个网络之间，实施网间访问控制的一组**组件的集合。**它满足：

①内部和外部之间的所有网络数据流**必须**经过防火墙。

②只有符合安全政策的数据流才能通过防火墙。

③防火墙自身**抗攻击。**

功能

1. **负载均衡**：
2. **内容安全**：病毒扫描、URL扫描、HTTP过滤

分类

- 包过滤
- 状态检测
- 应用级网关
- 代理服务
- 复合型
    - 屏蔽路由器型结构：路由器+防火墙
    - 单宿主主机：防火墙+堡垒机
    - 双宿主主机：防火墙+堡垒机（串联，需要破俩）
    - 屏蔽子网结构：双防火墙+子网

局限性

面对以下攻击，防火墙体现出不足：

1. 绕开防火墙的攻击
2. 内部攻击
3. 全新威胁
4. 病毒感染程序或文件传输
5. 端到端加密时，收效甚微
6. 对用户不透明——传输延迟
7. 新WLAN、手持终端带来的挑战

包过滤防火墙

思想：

①依据规则

②双向配置（入站和出战）

过滤规则：

- **基础：**源/目的地址、源/目的端口、IP协议、接口
- **策略：**匹配规则，不匹配则根据**缺省策略**

  缺省策略：①未禁止则允许。②未允许则禁止（推荐）

优缺点：

- **优点：**简单、透明、效率高
- **缺点：**规则制定不容易、无法认证

NAT技术

分类——SNAT 和 DNAT

- **SNAT：**表示外出（源地址转换）——①隐藏内网IP地址。②解决地址匮乏。
- **DNAT：**表示进入（目的地址转换）——①实现SNAT环境下的服务访问。②流量均衡

NAT方式

- **一对一：**一个IP对应转换一个IP（静态NAT）
- **多对多：**多个IP对应转换多个IP（动态NAT）
- **多对一：**多个IP对应一个IP+不同端口（NAT-PT/过载）

虚拟专用网VPN

VPN五大安全技术

- **隧道技术：**隧道协议将其它协议的数据帧或包重新封装在新的包头中发送。新的包头提供了路由信息，从而使封装的原始数据能够通过互联网络传递。
- **加解密技术：**
- **密钥管理技术：**
- **认证技术：**
- **访问控制：**

应用层安全协议

PGP

签名->压缩->加密

- **背景：**zip压缩每次结果都不一样（压缩率和计算速度的平衡）
- **原因：**签名后再压缩是为了方便保存未压缩的报文和签名，为了以后存储方便，如果签名是对压缩后报文进行的，势必要保存一份压缩的文档与之对应，或者增加部分计算。

  如果不保存压缩文档，则接收方会抵赖消息，而用于验证消息的压缩文档没有保存，最后验证签名只能得到压缩文档的hash值，没办法通过hash来复原压缩文档。

SET

背景

为了在Internet上进行在线交易时，保证信用卡支付的安全性而设计的开放规范。

优点

SET提供了**消费者、商家和银行**之间的认证

- 确保了网上交易数据的**保密性**
- 数据的**完整性**以及交易的**不可抵赖性**
- 能保证不将消费者**银行卡号**暴露给**商家**，不将消费者的**购物信息**暴露给**银行**

双重数字签名

- **定义：**发送者寄出**两个相关信息**给接收者，对这两组相关信息，接收者**只能解读其中一组**，另一组只能转送给第三方接收者，不能打开看其内容。这时发送者就需分别加密两组密文，并针对两组明文信息联系起来并进行签名，称其**双重数字签名。**
- **目的：**将订单信息（OI）和交易信息（PI）发送给商家（M），只希望商家（M）只能获取到里面的订单信息（OI）而不能获取交易信息（PI），因为用户的银行账户和密码存放在里面。而由商家将信息转发给银行时，希望银行只能获取交易信息（PI）和部分订单信息，而不能得到用户具体的订单信息（OI）。
- **产生步骤：**

  （1）持卡人（C）通过Hash算法分别生成订购信息（OI）和支付信息（PI）的消息摘要H(OI)和H(PI)。

  （2）把消息摘要H(OI)和H(P1)连接起来得到消息OP。

  （3）通过Hash算法生成OP的消息摘要H(OP)。

（4）持卡人（C）使用自己的私钥签名H(OP)得到双重数字签名Sign(H(OP))。

（5）持卡人（C）将消息(OI，H(PI)，Sign(H(OP)))用商家的公钥加密后发送给商家，将消息(PI，H(OI)，Sign(H(OP)))用银行的公钥加密后发送给银行。

- **验证步骤：**

（1）商家（M）将收到的消息（(OI，H(PI)，Sign(H(OP))）用自己的私钥解密后，将消息OI生成消息摘要H(OI)；同样银行将收到的消息（(PI，H(OI)，Sign(H(OP))）用自己的私钥解密后，将消息PI生成消息摘要H(PI)。

（2）商家（M）将生成的消息摘要H(OI)和接收到的消息摘要H(PI)连接成新的消息OP1；银行将生成的消息摘要H(PI)和接收到的消息摘要H(OI)连接成新的消息OP2。

（3）商家将消息OP1生成消息摘要H(OP1)；银行将消息OP2生成消息摘要H(OP2)。

（4）商家和银行均用持卡人的公共密钥解密收到的双重数字签名Sign(H(OP))得到H(OP)。

（5）商家将H(OP1)和H(OP)进行比较，银行将H(OP2）和H(OP)进行比较，若相同，则证明商家和银行所接收到的消息是完整有效的。

- **讲故事：**

SET 协议工作原理如图 3.1 所示：[9]

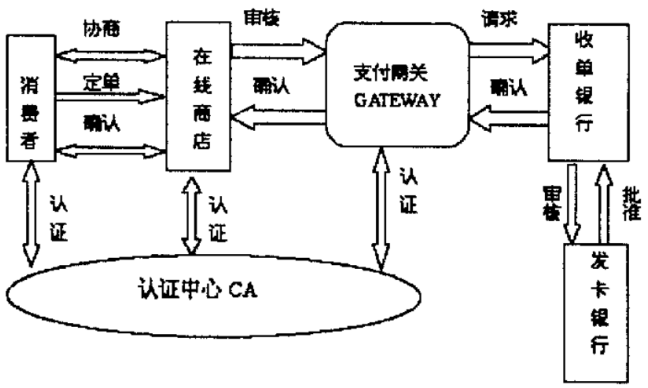


图 3.1SET 协议工作原理

Fig3.1 The Working Principle of SET Protocol

其具体工作流程如下：

①持卡人通过浏览器选择在线商店里自己需要的商品，放入购物篮。

②持卡人填写订单信息，并选择支付方式。

③持卡人将订单信息和支付信息发送给商家，这里订单信息和支付指令由消费者进行数字签名，同时利用双重签名技术保证商家看不到消费者的账号信息及银行看不到消费者的订单信息。

④商家接受订单量信息后，与支付网关进行通信，请求授权认证。

⑤支付网关通过收单银行向持卡人的发卡银行请求进行支付确认。

⑥发卡银行同意支付，将确认信息通过支付网关返回给商家。

⑦商家发送订单确认信息给持卡人，持卡人端软件可记录交易日志，以备将来查询。

⑧商家发送货物或提供服务。

⑨商家向持卡人的发卡银行请求支付，即实现支付获取、完成清算。

在处理过程中，通信协议、请求信息的格式、数据类型的定义等，SET 都有明确的规定。在操作的每一步，消费者、商家、网关都通过 CA 来验证通信主体的身份，以确保通信的对方不是冒名顶替。

WLAN安全

概述

目前最流行的Internet接入网技术之一

优点

- 安装部署灵活
- 即插即用，接入方便
- 用户自由移动方便

安全机制

- **用户认证**：
  - SSID：32位唯一标识
  - MAC地址过滤
- **用户授权**：
- **数据安全**：
  - 基于静态密钥：WEP协议
  - 基于动态密钥：802.1x

习题讲解

4.5 为了给公钥证书提供一种标准格式，X.509 中规范了一种认证协议。X.509 的原始版本包含了一个安全流程。这个协议的实质如下：

$$A \rightarrow B: \quad A\{t_A, r_A, ID_B\}$$

$$B \rightarrow A: \quad B\{t_B, r_B, ID_A, r_A\}$$

$$A \rightarrow B: \quad A\{r_B\}$$

这里，t_A 和 t_B 表示邮戳，r_A 和 r_B 是随机数，而符号 X{Y}表示消息 Y 通过 X 签名、加密和传输。

X.509 的内容指出：在三向认证中，检查时间戳 t_A 和 t_B 是可选的。但是考虑以下的例子：假定 A 和 B 在以前使用上面的协议，然后攻击者 C 截获了前面的三条消息。另外，假定时间戳未被使用而均被设定为 0。最后，假定 C 希望对 B 假扮 A。C 首先向 B 发送第一条被截获的消息：

$$C \rightarrow B: \quad A\{0, r_A, ID_B\}$$

B 做出应答，以为它在与 A 对话，而实际上是与 C 对话：

$$B \rightarrow C: \quad B\{0, r'_B, ID_A, r_A\}$$

C 其间通过某种方式使得 A 发起 C 的认证。结果，A 向 C 发送：

$$A \rightarrow C: \quad A\{0, r'_A, ID_C\}$$

C 使用与 B 提供给 C 相同的随机数应答 A：

$$C \rightarrow A: \quad C\{0, r'_B, ID_A, r'_A\}$$

A 应答：

$$A \rightarrow C: \quad A\{r'_B\}$$

这正是 C 需要使 B 确认它在与 A 进行对话的信息，所以 C 现在将此到来的消息重发给 B：

$$C \rightarrow B: \quad A\{r'_B\}$$

所以 B 将确信它正在与 A 对话，而实际上它是在与 C 对话。提出一种对于这个问题的简单解决方案，在这个方案中不要使用时间戳。

安全原理：

A→B：A发送挑战，要求B对rA签名，只有B能对其签名。

B→A：B响应挑战，并对A发起挑战。

A→B：A响应挑战，证明A的身份。

安全隐患：

攻击者C当作中间人，截获A向B发起通信请求。面对B的挑战，C与A通信，使得A间接响应B的挑战。

解决方法：

关键点在于挑战可以被攻击者C利用，C截获了挑战关键信息rB`，从而造成A间接响应B的挑战。因此破局点在于不让攻击者获取信息，这只需要在响应过程用请求方的公钥加密即可，这样即使C截获了B的挑战，也无法获得随机数rB`，进而无法让A相应B的挑战。而C直接将B的挑战发往A，由于rA不同，A将会拒绝继续通信。

4.6 考虑基于非对称加密技术的单向认证：

A→B：ID_A

B→A：R_1

A→B：$E(PR_a, R_1)$

a. 解释协议。

b. 协议易受什么类型的攻击?

a. A将自身主体身份信息，发送给B。B发送随机数给A，发送挑战。A对随机数用私钥签名（私钥加密），发送给B，响应挑战。

b. 容易受到中间人攻击。攻击者让A向其发起通信，获取IDA后冒充A与B通信，并将B发送的R1发送给A，A对其签名的信息被C发送给B，从而使得B认为攻击者C是A。

攻击思路：（伪装某一方）

1.中间人攻击，想办法让其中一方与其发起通信，最后破解消息

2.反射攻击，无法破解消息，但是可以让A误以为和B通信，但实际上在和C通信

3.旧密钥攻击，添加时间戳

第二次小测

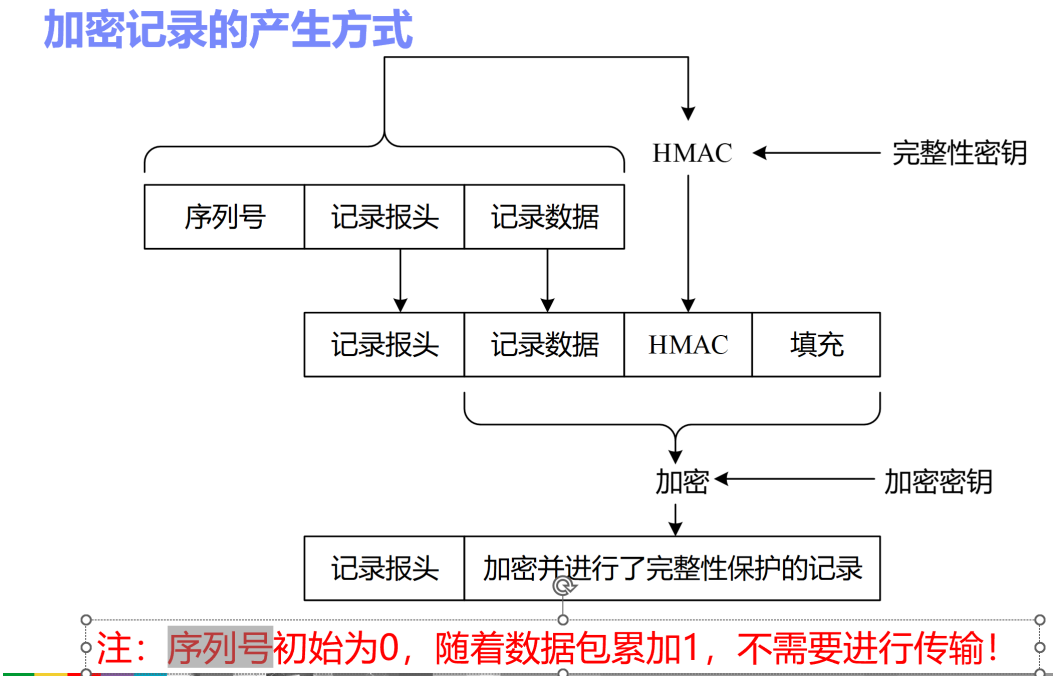
1.简单描述IPSec隧道模式和传输模式的区别?

- **传输模式**：为**IP载荷**提供认证、完整性和机密性。
- **隧道模式**：保护**整个IP分组**，提供认证、完整性和机密性。

2.IPSec如何防御重放攻击？SSL/TLS如何防范数据传输过程中的重放攻击？（重点数据传输，另外还有握手过程）

IPSec维护一个序列号窗口，当数据包序列号在窗口左边时直接丢弃。

SSL/TLS数据传输过程主要靠序列号来抗重放（握手过程中使用随机数）



3.AH计算MAC时，外层IP首部中哪些字段不包含在内？可选的ESP认证的覆盖范围和AH的有什么差别?

字段：可变不可预测的，例如跳数

覆盖范围：ESP不包括IP首部

4.为什么ESP包含一个填充域？除了ESP填充的功能之外，填充在协议设计中还有什么其他作用？

- 32位对齐
- 加密算法要求分组为某字节的整数倍
- 某些字段要求为某字节的整数倍（例如段偏移）
- 隐藏真实长度，抗流量分析

5.SSL/TLS中会话重用和密钥派生的作用和方法？

- 会话重用：
  - 方法：发送端发送之前的SessionID 接收端回复相同的SessionID，则表示会话重用
  - 作用：减少重新协商密钥产生的开销
- 密钥派生：
  - 方法：客户端和服务器产生随机数，双方协商或客户生成**预共享密钥**，利用密钥派生函数（KDF）派生
  - 作用：利用协商出来的预共享密钥，结合协商过程中双方的随机数计算后面的**加密以及完整性保护密钥**

6.IPSec AH/ESP、SSL/TLS中普遍不采用签名来提供完整性和数据源认证，为什么？

因为之前已经协商了共享对称密钥，基于该对称密钥可以使用HMAC提供完整性和数据源认证，相比于使用非对称密码的私钥签名计算效率更高，开销更小（非对称模幂运算与对称密钥计算速率相差两个数量级）