https://github.com/DiaosSama/SCUT_IS_Study

引言

Internet发展的主要动力

1. **分散式管理**和**商业化**是Internet快速发展的重要原因
2. **信息安全**是影响Internet应用和发展的一个重要因素

OSI安全框架（开放系统互联）

1. 需要通过系统化的方法来定义需求，定义了：
    - 安全攻击
    - 安全服务
    - 安全机制
2. 安全攻击
    - **攻击类型：中断、侦听、篡改、伪造**
    - 中断：对系统进行攻击，使得信息不能正常使用，破坏信息的**可用性（availability）**原则
    - 侦听：非授权方探听不该探听的信息，破坏了信息的**保密性（confidentiality）**原则
    - 篡改：非授权方中途拦截信息，重整之后发往目的地，破坏了信息的**完整性（integrity）**原则
    - 伪造：非授权方将伪造的客体插入系统中，破坏信息的**真实性（authenticity）**原则
    - 被动攻击与主动攻击：
        - 被动攻击：对传输进行窃听和监测，以获取信息内容或监控网络流量（被动攻击相对难以察觉，重点在于预防，而不是检测）
        - 主动攻击：伪装、重放、修改、拒绝服务（主动攻击难以防止，但容易检测）
3. 安全服务（达到什么目的）
    - 保密：保证数据不被泄露
    - 认证：保证通信实体与其宣称的相同
    - 完整性：保证数据接收时是完整的，数据没有被修改、插入、删除或重放
    - 不可否认：通信行为不可抵赖
    - 存取控制：阻止对资源的非授权访问
    - 数据可用：可按用户需求提供资源的存取和使用
4. 安全机制（用什么手段去实现安全服务）
    - 加密机制
    - 数字签名机制
    - 访问控制机制
    - 数据完整性机制
    - 认证交换
    - 业务流量填充
    - 路由控制
    - 公证

网络安全模型

1. 多层次的立体网络安全防护体系：通道、网关、系统、内核
2. 通道模式：在通路两端架设安全设备如VPN，加密路由器，加密防火墙等。目的是建立一个专用秘密通道，防止非法入侵，保证通路安全
3. 网关模式：在系统入口进行控制。涵盖面非常广，从应用层到链路层，从探测设备到安全网关等出入关控制设备等
4. 系统模式：一般在应用层进行，控制粒度可以到用户级或文件级，较为独立，不受通信协议的影响
5. 内核模式：操作系统中的安全内核是信息系统安全可靠的最基本要素

安全标准和组织

1. 国际标准化组织ISO
2. 国际电信联盟ITU
3. Internet体系结构委员会IAB
4. 互联网工程任务组IETF
5. 互联网工程指导小组IESG

传统加密技术

密码学发展

1. 第一阶段——传统密码（1949年前）
    - 主要特点：数据的安全基于算法的保密
2. 第二阶段（1949-1975）
    - 主要特点：数据的安全基于密钥而不是算法的保密
3. 第三阶段（1976年以后）：Deffie&Hellman提出了不对称密钥密码
    - 主要特点：公钥密码使得发送端和接收端无密钥传输的保密通信成为可能

术语及定义

1. **分类**

- 根据加密操作类型
    - 代换
    - 置换
    - 多重加密
    - 根据所用的密钥的个数
        - 单密钥算法或秘密密钥算法（又称对称加密算法）
        - 双密钥算法或公开密钥算法（又称不对称加密算法）
    - 根据明文被处理的方式
        - 分组密码
        - 流密码（又称序列密码）

2. 对称加密
    - 安全使用对称加密的两个要求
        - 加密算法足够强大：仅知密文很难破译出明文
        - 基于密钥的安全性，而不是基于算法的安全性
    - 前提：
        - 算法开放，易于实现

        - 存在一个安全通道来分发密钥
3. 针对数据加密的攻击手段
    - 密码分析：从数学的角度分析密码算法
        - 惟密文攻击：拥有加密算法、要解密的密文
        - 已知明文攻击：拥有加密算法、要解密的密文、明密文对
        - 选择明文攻击：拥有加密算法、要解密的密文、有目的选择的一些明文以及对应的密文
        - 选择密文攻击：拥有加密算法、要解密的密文、有目的选择的一些密文以及对应的明文
        - 选择文本攻击：拥有加密算法、要解密的密文、有目的选择的一些明文以及对应的密文、有目的选择的一些密文以及对应的明文
    - 强力攻击：逐次试用每个密钥进行解密（爆破）
        - 穷举攻击
4. 攻击复杂性分析：数据复杂性、处理复杂性、存储需求
5. **加密算法的安全性**
    - **无条件安全**
        - 无论提供的密文有多少，一个加密方案产生的密文中包含的信息不足以唯一的决定相对应的明文
        - **除了一次一密，没有无条件安全的算法**
    - **计算上安全**
        - 破译密码的价值超出密文信息的价值
        - 破译密码的时间超出密文信息的有效生命期

代换技术

1. 定义：将明文字母替换为其他字母、数字或符号。如果把明文看成是二进制的序列的话，那么代换就是用密文位串来代替明文位串
2. 分类：
    - 单表代换：用一个代换表决定代换规则
        - 单字母代换：凯撒密码
        - 多字母代换：Hill、Playfair密码
    - 多表代换：用多个代换表决定代换规则
        - 周期：维吉尼亚密码
        - 非周期：一次一密
3. 凯撒密码：简单的替换，改进为单表替换密码
4. Playfair：填入5X5格子中，通过规则变换（以两个字母为单位）（同一行同一列、矩形对角）
5. 一次一密：一个随机密钥，与消息一样长并且不重复，就称为一次一密（问题在于密钥的安全分发比较困难）

置换技术

1. 定义：通过重新安排字母的顺序来隐含信息，没有改变所使用的的实际字母，密文与原文有同样的频率分布
2. 例子：栅栏加密

转子机

1. 有n个圆筒，则有$$ 26^{n} $$个替换密码表

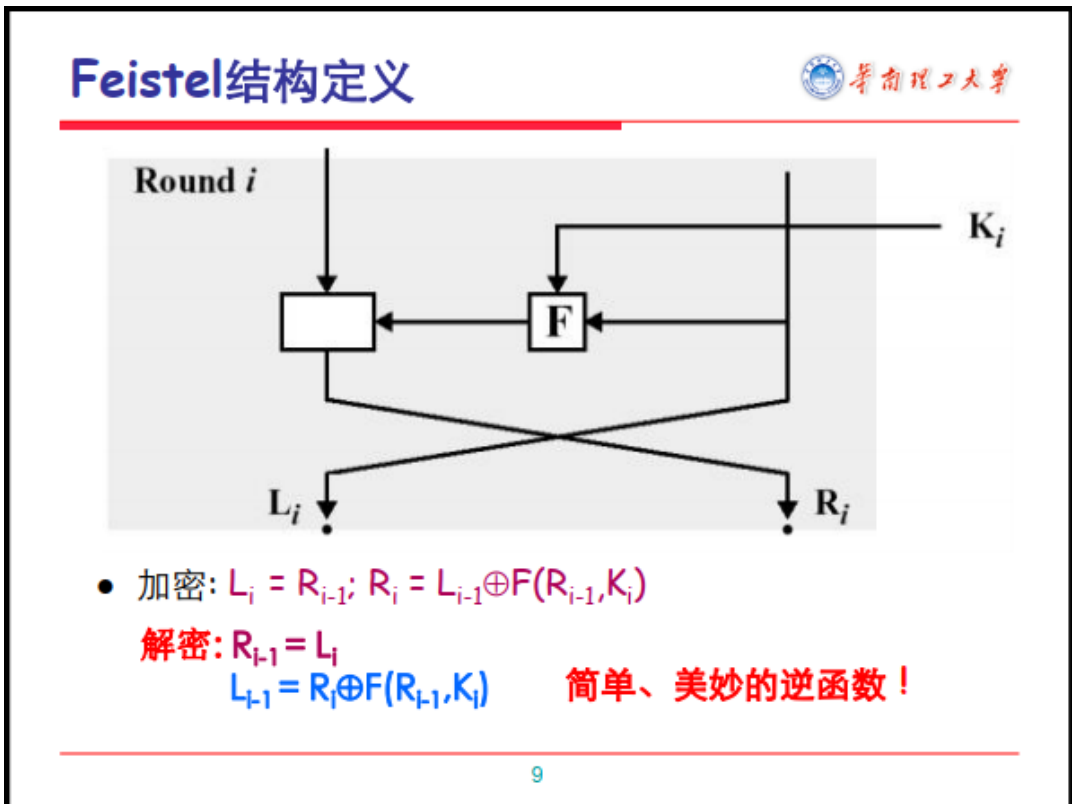
隐写术

需要许多额外的付出来隐藏相对较少的信息

分组密码与DES

分组密码与Feistel密码

1. 分组密码
    - 定义：将明文消息编码表示后的数字序列，划分成长度为n的组，每组分别在密钥的控制下变换成等长的输出数字序列
    - 原理：
        - 看起来像一个特别大的代换
        - 应用了多重加密的概念
        - **对一个n位bit的分组**
            - **不同明文分组的总数？ $$ 2^{n} $$**
            - **不同可逆变换的总数？ $$ 2^{n}! $$**
            - **密钥的长度？ $$ n2^{n} $$**
2. 代换 - 置换加密
    - 1949年香农引入代换 - 置换网络（SPN）的概念
    - 是现代分组加密的基础
    - 基于两种基本的加密操作：代换（S盒）、替换（P盒）
    - 目的是实现消息的**混乱和扩散**（隐藏原始消息的统计性质）
3. **混乱与扩散**
    - 扩散：明文的统计结构被扩散消失到密文的长程统计特性，使得**明文**和**密文**之间的统计关系尽量复杂
    - 混乱：使得**密文**的统计特性与**密钥**的取值之间的关系尽量复杂
4. Feistel Cipher结构
    - 基于可逆多重加密的概念
    - 将输入分组分为两半，经过多轮处理，每个处理包括
        - 对左半数据进行一次替换操作
        - 对右半数据和子密钥做循环函数映射
        - 将两边置换一下
    - 结构定义：

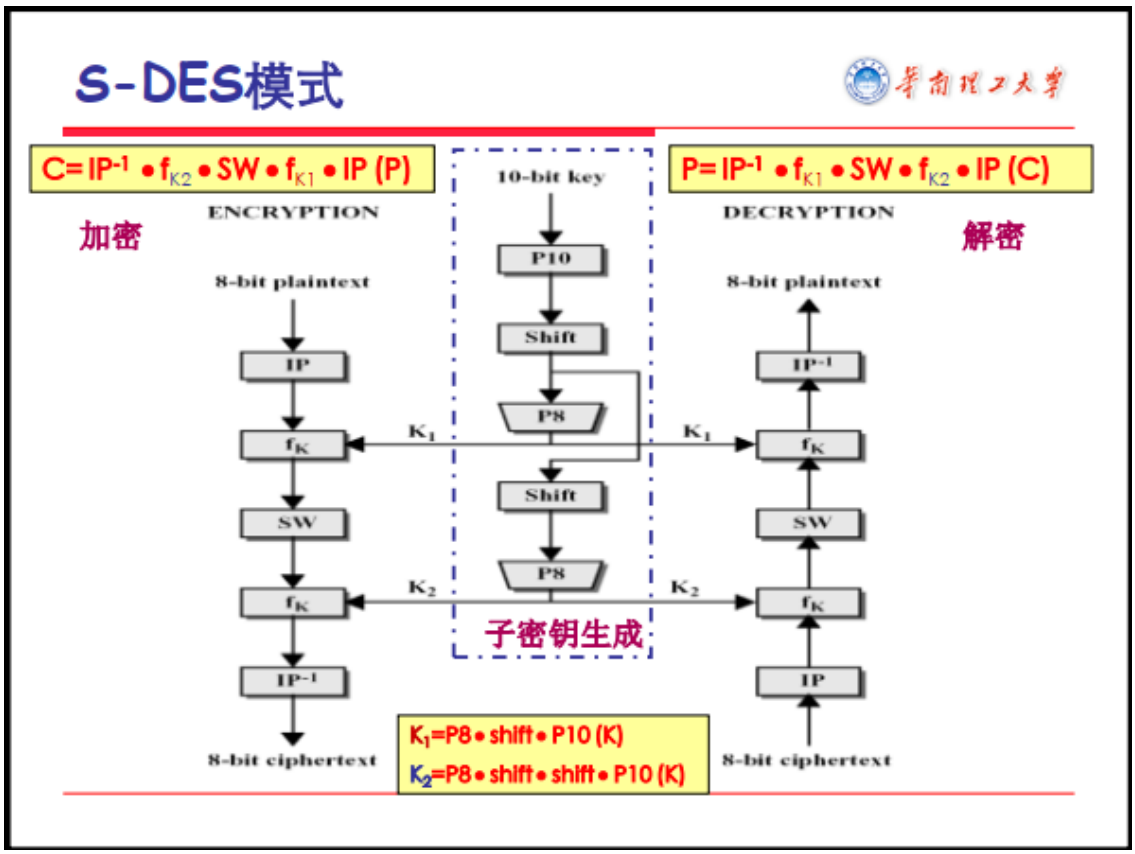


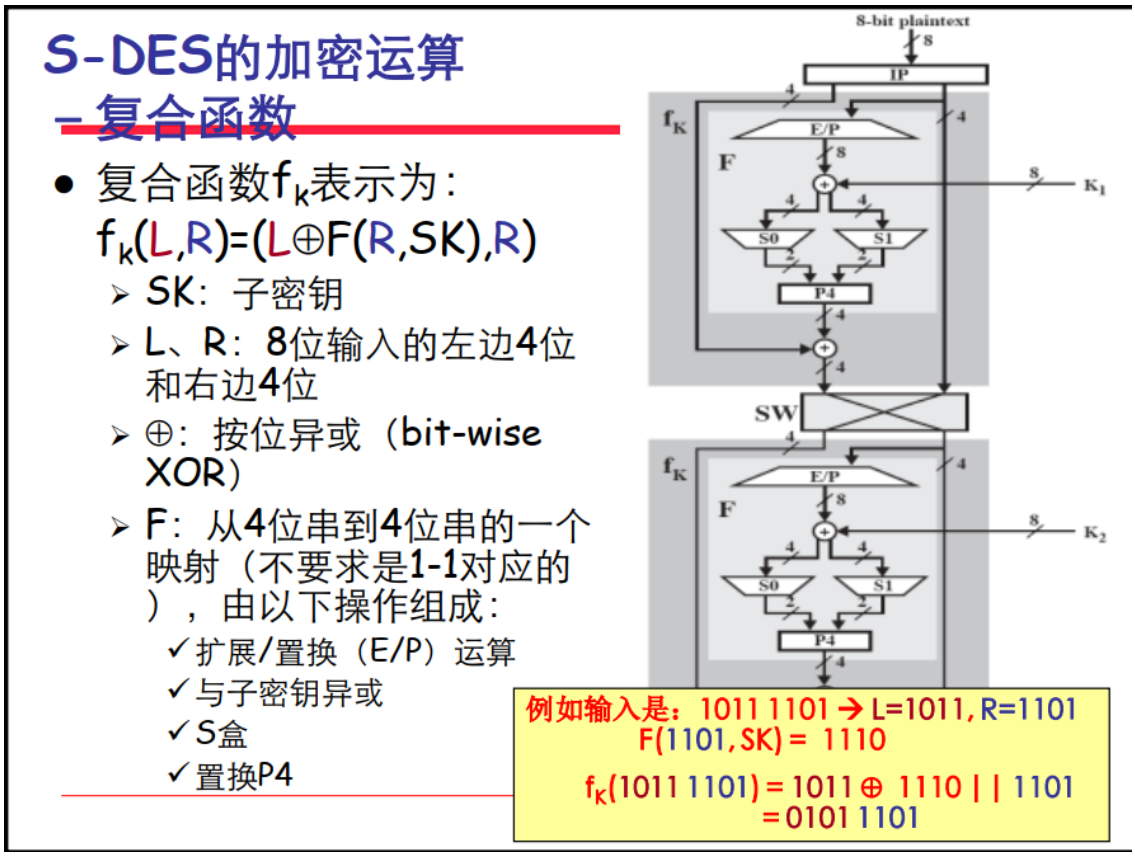
- **解密时，过程与加密过程一致，但是逆序使用子密钥K**

5. Feistel密码设计原理
    - 分组大小：64位（分组越大安全性越强，加解密速度越低）
    - 密钥大小：128位（密钥越大安全性越强，加解密速度越低）
    - 循环次数：16次
    - 子密钥生成算法：较大的复杂性会增大密钥分析的难度
    - 循环函数：较大的复杂性会增大密码分析的难度
    - 加解密的快速软件实现及易于分析

数据加密标准DES之S-DES（简化算法）

1. 加密算法涉及五个函数：（8bit明文，10bit密钥）
    - 初始置换IP
    - 复合函数$$ f_{k1} $$ ，它由密钥K1决定，具有置换和替代的运算
    - 转换函数SW
    - 符合函数$$ f{k2} $$ ，与$$ f{k1} $$ 相同，所用的密钥是K2
    - 初始置换的逆置换$$ IP^{-1} $$
2. S-DES模式





S盒运算规则：4个输入比特，1、4比特确定S盒的一个行，2、3比特指定S盒的一个列，输出为2比特

3. S-DES密码分析

- 强力搜索：只有$$ 2^{10} = 1024 $$ 种可能的密钥
- 已知明文攻击：非线性来自S0和S1两个盒子

数据加密标准DES

1. DES的密钥长度为56bits，明文长度为64bits，密文长度为64bits
2. 在F函数中，对32bits的右半部分和48bits子密钥进行变换
3. S盒子：
    - 8个S盒，每个S盒将6比特映射为4比特，结果为32比特
    - 每个S盒是一个16 x 4的盒子：
        - 外比特：即1、6比特组成的二进制数确定行
        - 内比特：即2-5比特组成的二进制数确定列

DES的设计原理及密码分析

1. 设计原理：
    - 基本原理仍然是1970的Feistel密码
    - 循环次数：越多越好
    - 函数f：提供混乱，是非线性的，雪崩效应
    - 密钥产生：复杂的子密钥生成，密钥的雪崩效应
2. 实现原则
    - 安全上的要求：
        - 分组长度足够大
        - 密钥长度足够大
        - 算法足够复杂
    - 软件实现要求：使用子块和简单的运算
    - 硬件实现要求：加解密的相似性、尽量采用标准的组件结构
3. DES雪崩效应
    - 是加密算法期望的一个好性质
    - 明文或密钥的某一位发生变化，会导致将近一半输出密文比特的改变
4. DES攻击方法：
    - 差分分析
    - 线性分析

DES的工作模式

1. 为了应用于实际，DES的三种工作模式：
    - 电子密码本ECB（electronic codebook）
    - 密码分组链接CBC（cipher block chaining）
    - 计数器CTR（counter）
2. 电子密码本ECB
    - 消息被分解为独立的分组来加密，正如密码本一样，每个分组的加密独立与其他分组
    - 优点：
        - 简单和有效
        - 可以并行实现
        - 误差传递：密文块损坏仅对应明文块损坏
    - 局限性：
        - 不能隐藏明文的模式信息：相同明文=》相同密文
        - 同样信息多次出现造成泄漏
        - 对明文块的主动攻击是可能的

        - 信息块可被替换、重排、删除、重放
    - **适合传输短信息，例如用于密钥的安全传输**
3. 密码分组链接CBC
    - 加密算法的输入是当前的明文组和上一个密文组的异或（第一个明文组与初始向量IV异或）
    - 优点：
        - 能隐藏明文的模式信息
        - 对明文的主动攻击是不容易的
        - 信息块不容易被替换、重排、删除、重放
        - 安全性好于ECB
        - 误差传递：密文块损坏 两明文块损坏
    - 局限性：
        - 没有已知的并行算法
        - 需要共同已知初始向量IV
    - **适合传输长度大于64位的报文**
4. 计数器CTR
    - 只加密计数器值而不是任何反馈值，对每一个明文，必须有不同的密钥和计数器值
    - 优点：
        - 效率高：可以并行加密、预处理、适用于高速连接
        - 可随机访问加密的数据块
    - 局限性：
        - 必须保证不重用密钥、计数器值
    - **适用于高速网络加密**

现代对称加密及其传输保密性

现代对称加密——三重DES

1. 为什么不是双重DES？——中间相遇攻击
    - 攻破双重DES的计算量为$$ O(2^{56}) $$ ，跟攻击单DES（$$ 2^{55} $$）差不多
2. 分类：双密钥的3DES和三密钥的3DES
    - 使用两个密钥，运算过程为：加密-解密-加密（交替使用K1和K2）

现代对称加密——Blowfish

1. 特点：快速、紧凑、简单、安全性可变

现代对称加密——RC5

1. 特点：
    - 灵活
        - 适应不同字长的程序
        - 加密轮数可变
        - 密钥长度可变
    - 干净而简洁
    - 安全

现代对称加密——流密码

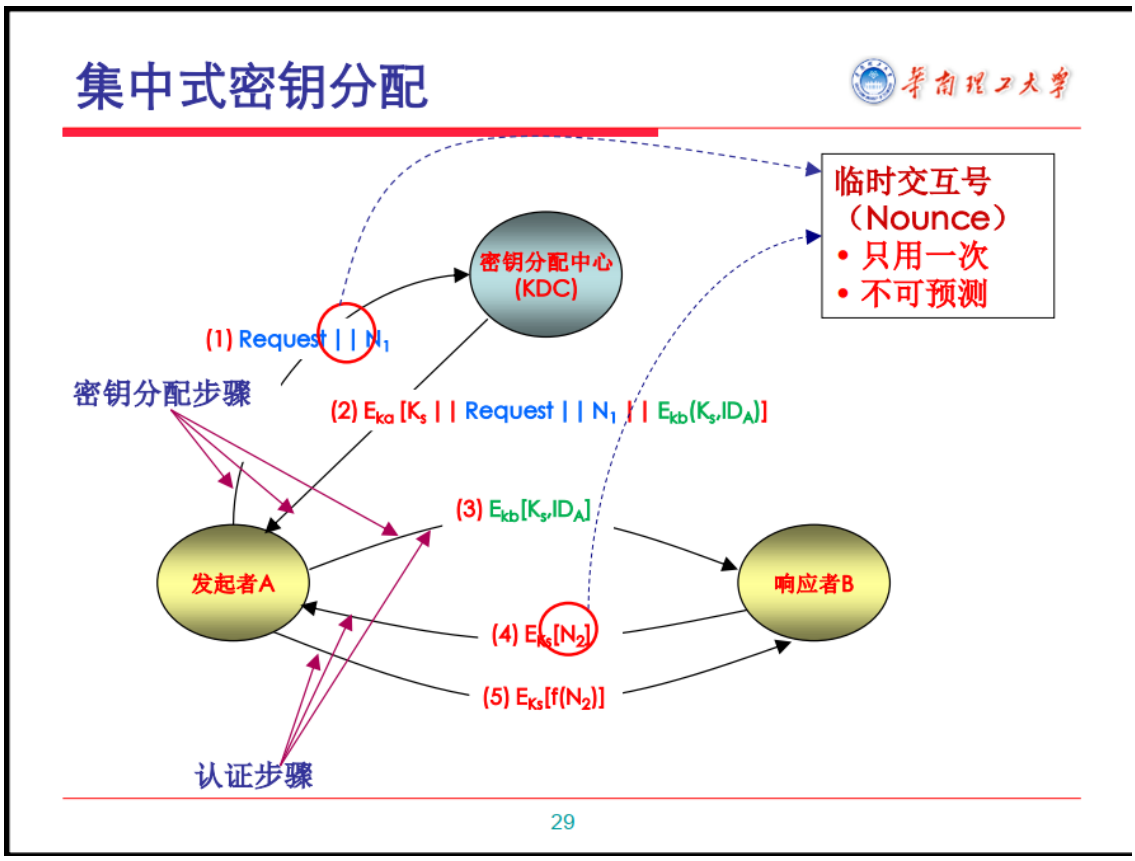
1. 方式：
    - 一位位地处理消息（当做流）
    - 一般有一个伪随机的流密钥，与明文按位异或
2. 优点：
    - 随机性，完全粉碎消息的任何统计信息
3. 缺点：
    - 必须不重用流密钥

传输安全性——加密设置

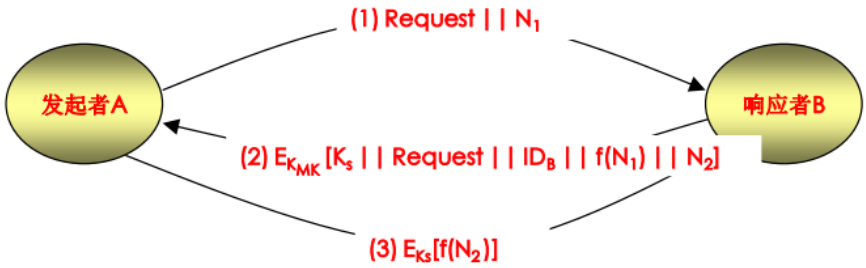
1. 链路加密和端到端加密
    - 链路加密，每两个节点都要协商一个加密规则，报文在中间节点需要解开才能再次转发
    - 端到端加密，只有起点终点需要协商加密规则，只加密报文的应用层，中间节点可以直接转发

传输安全性——密钥分配

1. 定义：在交换双方之间共享密钥
2. **分类：**
    - **层次型密钥分配**
    - **集中式密钥分配**
    - **分散式密钥分配**
3. 层次型密钥分配
    - 通过KDC的帮助，发送者和接收者建立一个会话密钥$$ K_{AB} $$ 用以在一个会话中加密数据
    - 主密钥：
        - 有双方共享的长期密钥
        - 用来创建临时的会话密钥
    - 会话密钥：
        - 在线创建
        - 只在一个逻辑会话中加密数据
        - 经常改变
4. 集中式密钥分配
    - 先决条件：
        - 一个第三方：密钥分配中心（KDC）
        - 每个用户X与KDC共享一个长期密钥$$ K_{X} $$
    - 适用于小型系统



5. 分散式密钥分配
    - 先决条件：Alice和Bob共享一个主密钥$$ K_{MK} $$
    - Alice和Bob建立一个会话密钥$$ K_{S} $$



传输安全性——随机数的产生

1. 用途
    - 认证过程中避免重放攻击
    - 会话密钥
    - RSA公钥算法
2. 特点：随机性、不可预测性

公钥密码学与RSA

公钥密码体系

1. 私钥加密存在的问题：
    - 无法保护发送方，如果接收方伪造一个消息并宣称是发送方发送的
    - 密钥的分配
2. 为什么需要公钥加密？

    解决上述的两个问题：

    - 密钥分配
    - 数字签名
3. 为何先签名、后加密?
    - 签名对最原始的信息进行操作
    - 防止别人伪造签名
4. 公钥应用:
    - 加密解密
    - 数字签名
    - 密钥交换
5. 对公钥密钥的要求
    - 从已知算法和加密密钥推出解密密钥在计算上是不可行的
    - 当相关的加密/解密密钥已知时，加密/解密消息的计算比较容易
    - 两个相关的密钥中的任何一个可用于加密，另一个用于解密

数学原理

1. Fermat定理（费马定理）

   $$ a^{p-1} \mod p = 1 $$

   （这里p是素数，并且gcd(a, p) = 1）
2. Euler函数（欧拉函数）

   表示小于n且与n互素的正整数的个数，若n为素数，$$ \phi(n) = n-1 $$
3. 欧拉定理

   有$$ a^{\phi(n)} \mod n = 1 $$，如果gcd(a, n) = 1
4. 欧拉定理推论

   $$ a^{\phi(n)+1} \mod n = a $$, k是整数

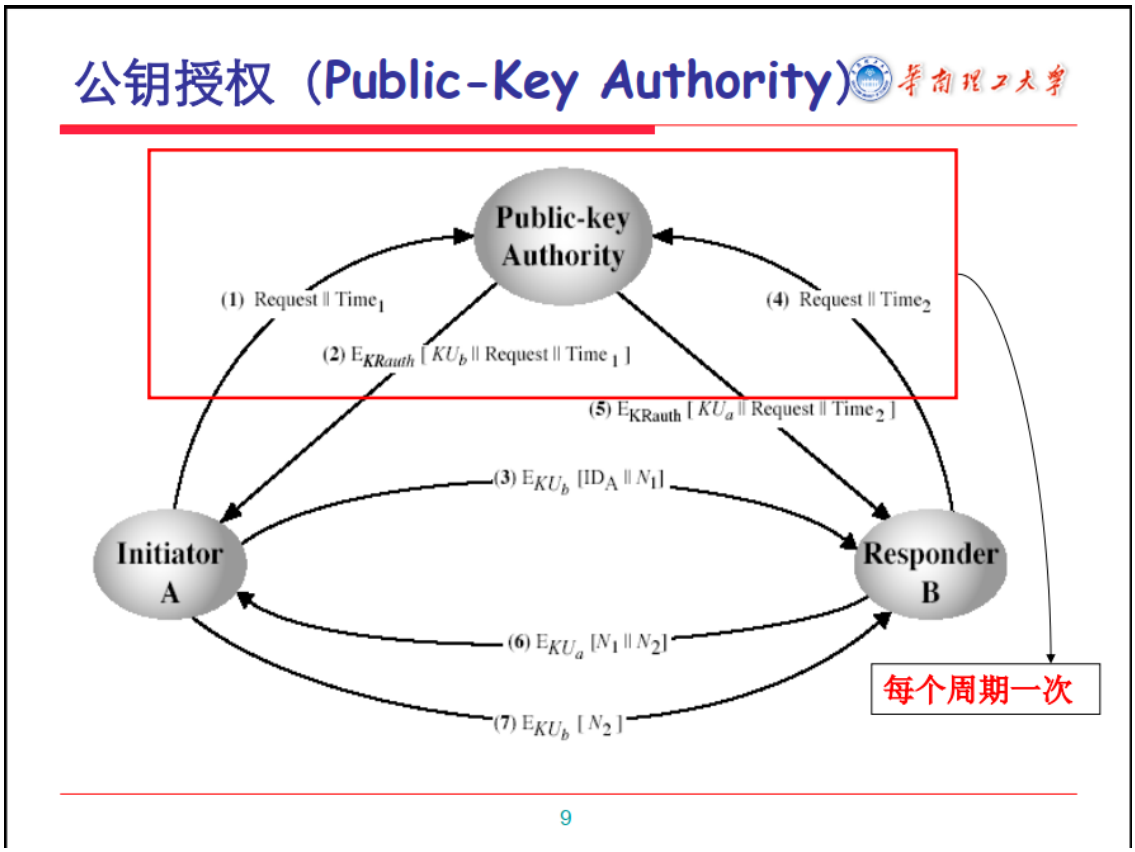
RSA算法

1. 步骤:
    - 随机选取两个素数p, q
    - 计算N=pq
    - $$ \phi(n)=(p-1)(q-1) $$
    - 随机选择一个加密密钥e，$$ gcd(e,\phi(n))=1 $$
    - 由$$ ed\mod \phi(n)=1 $$, 求得e
    - 发布公钥 KU={e, N}
    - 保存私钥 KR={d, N}
2. RSA的安全性
    - 强力密钥搜索：考虑到密钥空间大小，不可行
    - 数学攻击：等价于分解模N
    - 时间攻击：通过解密的运算
3. 实现要求
    - p与q必须为足够大的素数（大约100位的十进制素数）
    - 模n的长度至少要求是512比特
    - 为防止很容易分解n，pq还有如下要求
        - p和q的长度应仅相差几位
        - （p-1）和（q-1）都应该有一个大的素因子
        - gcd(p-1, q-1) 应该较小

密钥管理

基于公钥的密钥分配

1. **分配方法**
   - **公开发布**
   - **公开可访问目录**
   - **公钥授权**
   - **公钥认证**
2. 公开发布
   - 用户分发公钥给接受者，或者在一定范围内广播
   - 优点：简单
   - 最大的缺点：伪造（任何人都可以创建一个公钥，然后宣称是别人的）
   - 应用：PGP（pretty-good-privacy email system）使用这种方法
3. 公开可访问的目录
   - 向公共目录注册密钥获得更高的安全性
   - 一个可信的组织维护一个动态的目录
   - 缺点：目录是攻击的目标。仍可能被篡改或伪造。
4. 公钥授权

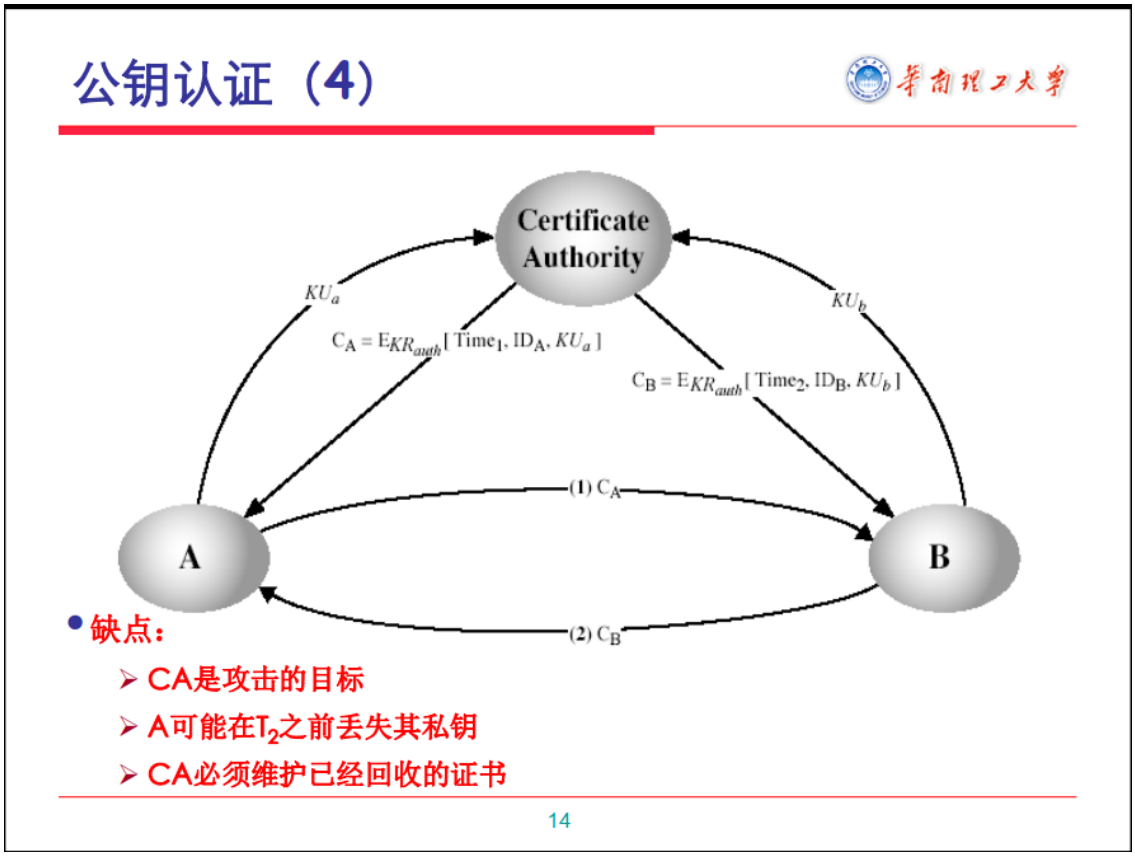


   - 通过严格控制目录中的公钥分配，提高安全性
   - 所有用户必须知道管理者的公钥
   - 缺点：
     - 公钥授权是一个瓶颈，所有用户都要与之通信以获取其他人的公钥
     - 公钥授权仍是攻击的目标
5. 公钥认证
   - 无需实时通讯的密钥交换
   - 一个证书包含标识和公钥（还包含有效期，使用权限等）

- 所有内容由可信的认证机构签发
- 每个通信方要求CA为其公钥签发一个证书

  $$ C{A}=E{KRauth}[T,ID{A},KU{A}] $$

  - T=(T1, T2) 是证书的有效期，说明证书$$ C_{A} $$ 在时间T1和T2内有效



基于公钥的密钥分配——简单模式

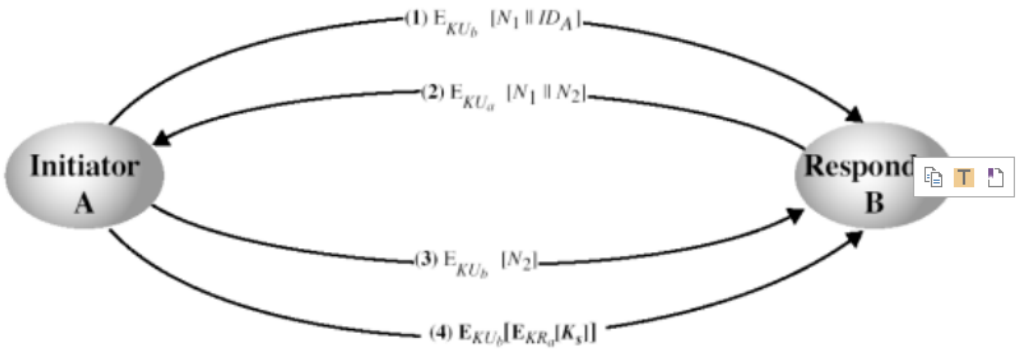
1. 步骤：
   - A将公钥及其标识符发给B
   - B生成会话密钥K，并用A提供的公钥加密后，发给A
   - A解密会话密钥，然后用会话密钥通信
2. 问题：如果存在主动攻击，攻击者可以拦截并模仿双方的通信

基于公钥的密钥分配——保密模式

1. 具有保密性和真实性的模式：假设已经交换了公钥

基于公钥的密钥分配：保密模式

- 具有保密性和真实性的模式：假设已经交换了公钥



- 问题：过程中临时交互号N_1和N_2的作用？

基于公钥的密钥分配——混合模式

1. 定义：公钥方法用来在KDC和用户之间分配主密钥，然后通过该主密钥的加密来实现秘密的会话密钥的分配
2. 步骤：
    1. KDC和用户A、B均拥有公钥对，并已经相互交换了公钥（公钥证书的交换）
    2. 在KDC和A、KDC和B之间生成主密钥Ka，Kb（基于公钥的保密模式密钥分配）
    3. 通过KDC在A和B之间生成会话密钥Ks（基于KDC的集中式密钥分配）
3. 性能：适用于需要频繁交换会话密钥的应用

Diffie-Hellman密钥交换

1. 特点：
    - 一种公钥分配模式，不能用于交换任何信息，用于创建一个只为两个通信方共享的密钥
    - 两个用户通过非安全通道创建秘密的会话密钥
    - 密钥值依赖于通信方的私钥和密钥信息
2. 步骤：
    - 双方共同选择大素数q（例如512bits长）以及q的一个本原根α
    - 用户A选择一个随机数 $$ X{A}<q $$，*计算公钥* $$ Y{A}=\alpha^{X_{A}}\mod q $$
    - 用户B选择一个随机数 $$ X{B}<q $$，*计算公钥* $$ Y{B}=\alpha^{X_{B}}\mod q $$
    - 每一方保密$$X$$值，而将$$Y$$值交换给对方
    - 用户A计算出$$ K{AB}=Y{B}^{X_{A}}\mod q $$
    - 用户B计算出$$ K{AB}=Y{A}^{X_{B}}\mod q $$
    - 双方获得一个共享密钥$$K_{AB}$$
3. 安全性：基于离散对数问题的困难性

单向陷门函数

1. 定义：满足下列条件的函数$$f$$ 称为单向陷门函数：
    - 给定x，计算 y=f(x) 是简单的
    - 给定y，计算 x=f(y) 是困难的
    - 存在z，已知z时，对给定的任何y，若相应的x存在，则计算x使 y=f(x) 是容易的
2. 满足条件1、2的称为单向函数，第3条称为陷门性

消息认证与Hash算法

消息认证

1. 信息安全的需求：保密性、完整性、可用性、认证、不可否认性
2. 通信系统典型攻击：
    - 泄密
    - 传输分析
    - 伪装：从一个假冒消息源向网络中插入消息
    - 消息篡改：内容、顺序、时间
    - 发送方否认：发送方否认发送过消息
    - 接收方否认：接受者否认收到消息
3. 消息认证可以用于解决：伪装、消息篡改
4. **认证：一个短的字符串V追加到消息M之后，用以认证该消息**
5. **一个安全的认证系统，需满足：**
    - 意向的接受者能够检验和证实消息的**合法性**、**真实性**和**完整性**
    - 消息的发送者和接受者不能抵赖
    - 除了合法的消息发送者，其他人不能伪造合法的消息
6. 首先要选好**恰当的认证函数**，该函数产生一个认证标识，然后在此基础上，给出**合理的认证协议**，使接受者完成消息的认证
7. 可用来做认证的函数分为三类：
    - 消息加密函数：用完整信息的密文作为对信息的认证
    - 消息认证码MAC：是密钥和消息的公开函数，产生一个固定长度的值作为认证标识
    - 散列函数：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，作为认证值

消息加密

1. 特点：
    - 消息的自身加密可以作为一个认证的度量
    - 包括对称加密和公钥加密，两者有所不同
2. 对称加密（保密性与认证）：
    - 提供保密性：只有A和B共享K
    - 提供认证：
        - 仅来自A
        - 传输中没有被更改
        - 需要某种结构或冗余
    - 不提供签名
        - 接收方可以伪造消息
        - 发送方可以否认消息
3. 公钥加密（保密性）：

    - 提供保密：只有B拥有解密的密钥$$KR_{b}$$
    - 不提供认证：任何一方都可以用$$KU_{b}$$对消息加密并假称是A
4. 私钥加密（认证与签名）：
    - 提供认证和签名
5. 公钥加密，先签名，后加密（保密性、认证与签名）

消息认证码MAC

1. MAC：使用一个密钥生成一个固定大小的短数据块，并加入到消息中
    - $$MAC(K, M) = C_{k}(M)$$
2. 前提：A和B共享一个密钥
3. 基本用法：
    - 消息认证：使用密钥生成MAC，附在消息之后
        - 仅提供认证
    - 消息认证与保密性（1）：使用密钥生成MAC，附在消息之后，再使用密钥加密
        - 提供认证和保密性
    - 消息认证与保密性（2）：使用密钥K2加密消息，再用密钥K2为密文生成MAC，附在密文之后
        - K1提供认证，K2提供保密性
4. **基于DES的消息认证码**
    - 数据认证算法DAA
        - **使用CBC方式方式，初始向量为0**
        - 将数据按64位分组，必要时最后一个数据块用0向右填充
5. 问题：
    - 保密性与真实性是两个不同的概念
    - 根本上，信息加密提供的是保密性而非真实性
    - 加密代价大
    - 某些信息只需要真实性，不需要保密性

散列函数

1. 定义：
    - 输入为一个任意长度的消息M
    - 输出为一个固定长度m的散列值，称为**消息摘要**
2. 特点：
    - 这个散列值是消息M的所有位的函数并提供错误检测能力
3. 散列函数需求：
    - 容易计算：比密钥加密块
    - 单向性
    - 抗碰撞性：
        - 弱抗冲突：任给x，不容易找到 y≠x，使得 H(x)=H(y)
        - 强抗冲突：不容易找到 x 和 y 使得 H(x)=H(y)
4. Hash与MAC的区别
    - MAC需要对全部数据进行加密
    - MAC速度慢
    - Hash是一种直接产生验证码的方法
5. **Hash函数通用结构：**

- 把原始消息M分成一些固定长度的块$$Y_{i}$$
- 最后一块padding并使其包含消息M的长度
- 设定初始值$$CV_{0}$$
- 压缩函数$$f$$, $$CV*{i}=f(CV*{i-1},Y_{i-1})$$
- 最后一个$$CV_{i}$$为hash值

MD5

1. MD5把数据分成512bits
2. MD5的hash值是128bits

数字签名和认证协议

数字签名

1. 数字签名是笔迹签名的模拟：
   - 必须能够验证作者及其签名的日期
   - 必须能够认证签名时刻的内容
   - 签名必须能够由第三方验证

   因此，数字签名功能包含了认证的功能
2. 设计需求：
   - 签名必须使用某些对于发送者是唯一的信息，防止双方的伪造与否认
   - 生成、识别、验证签名必须比较容易
   - 伪造该数字签名在计算上不可行
   - 保存一个数字签名副本是可行的
3. **分类**
   - **直接数字签名**
   - **仲裁数字签名**
4. 直接数字签名（DDS）：A→B
   - $$E_{KRa}[M]$$：提供了认证与签名
   - $$E*{KUb}[E*{KRa}[H(M)]]$$：提供保密性（KUb）、认证和数字签名（KRa）
   - $$M||E_{KRa}[H(M)]$$：提供认证及数字签名
   - $$E*{K}[M||E*{KRa}[H(M)]]$$：提供保密性、认证和数字签名

   缺点：
   - 方法的有效性依赖于发送方私钥的安全性
5. 仲裁数字签名
   - 引入仲裁者，仲裁者在这一类签名模式中扮演敏感和关键的角色

认证和交换协议

1. 认证与密钥交换协议的核心问题：
   - 保密性：防止伪装和暴露会话密钥
   - 时效性：防止消息重放攻击
2. 重放攻击
   - 常见形式：
     - 简单重放：简单复制一条消息然后重新发送
     - 可检测的重放：在一个合法有效的时间窗内重放一个带时间戳的消息
     - 不可检测的重放：原始信息被拦截，只有重放的信息到达目的地
     - 不做修改的反向重放：向消息发送者重放

- 对策：
  - 使用非重复值：序列号、时间戳、随机值/响应值

认证应用

1. 认证应用分类：
   - Kerberos：私钥认证服务
   - X.509目录认证服务
2. Kerberos：私钥认证服务
   - 环境：
     - 开放的分布式环境，服务器在网络上分布
     - 用户希望获得服务器上的服务
     - 服务器可限制用户访问
     - 服务器可对服务请求进行认证
   - 目标：
     - 安全
     - 可靠
     - 透明：用户没有感觉到认证的发生，只需要使用密码进入系统
     - 可扩展
3. Keberos4：提供认证服务和保密性
   - 认证服务器AS
     - 用户通过与AS协商表面自己的身份
     - AS提供不可修改的认证标识TGT（ticket granting ticket）
   - 票据授权服务器TGS
     - 基于TGT，用户随后向TGS请求对一些服务的访问
   - 票据Ticket
     - 由安全服务器（AS或TGS）颁发给用户C的加密消息$$Ticket_{V}$$，使得用户C可以用T进入服务器V
     - $$Ticket*{V}$$ *使用服务器V与安全服务器的共享密钥* $$K{V}$$加密，因此用户并不知道内容
     - $$Ticket*{V}=E*{Kv}(ID*{C},AD{C},ID*{V},Lifetime,K{CV})$$
   - **认证过程**
     - AS
       - 检查要求登录系统的用户A的密码的有效性
       - 颁发一个TGS的票据授权票据（TGT）$$Ticket_{tgs}$$给A，使得A可以进入TGS
     - TGS
       - 检查用户发送的$$Ticket_{tgs}$$的有效性
       - 处理用户A的服务请求
       - 颁发一个服务授权票据$$Ticket_{V}$$给A，使得A可以进入应用服务器V
     - 应用服务器V
       - 检查用户A发送的$$Ticket_{V}$$的有效性
       - 对用户A的服务请求授权或拒绝
   - 应用
     - **Kerberos域（realm）**
4. X.509目录认证服务
   - 是X.500目录服务的一部分
   - 基于公钥加密和数字签名

- 为用户提供公钥证书
- 基于公钥证书定义认证协议

5. X.509部分定义
    - CA（Certificate Authority）
        - 一个可信任的颁发证书的服务器
        - CA X 有一个签名密钥$$KR*{X}$$和与之对应的一个大家都知道的验证密钥$$KV*{X}$$
        - 存在许多CA，通常以树结构组织
    - 证书（Certificate）
        - 证书由CA X颁发
        - 证书由CA或用户放在目录中，使得所有人均可查询数据
        - 用户A的证书包括：
            - 颁发者的名字
            - A的公钥$$KU_{A}$$
            - CA X 的签名$$Sig(KR*{X},ID*{A},KU_{A})$$
            - 过期时间
        - $$Cert*{A,X}=[ID*{A},KU*{A},Sig(KR*{X},ID*{A},KU*{A})]$$
    - X.509层次结构
        - 每个CA入口包含两种证书：前向证书、后向证书
    - 认证过程：
        - 单向认证
        - 双向认证
        - 三向认证

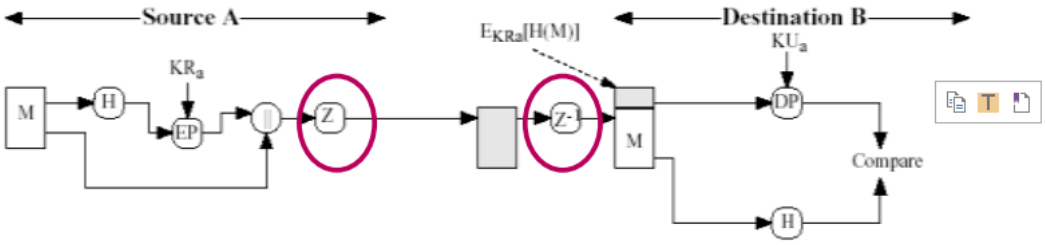
电子邮件安全

PGP

1. 概述
    - 数字签名：DSS/SHA或RSA/SHA
    - 消息加密：CAST-128 或 IDEA 或 3DES+Diffie-Hellman 或 RSA
    - 数据压缩：ZIP压缩
    - 邮件兼容：Base64转换
    - 数据分段

PGP操作：认证

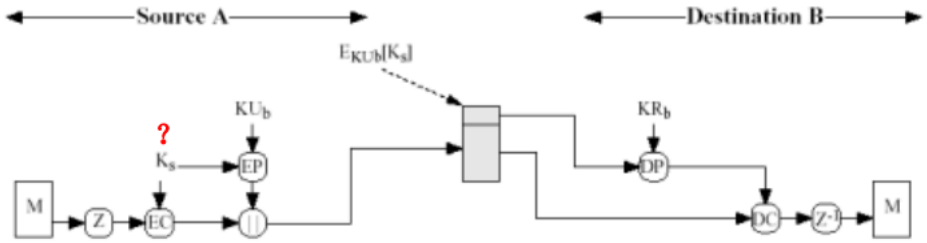
PGP 操作：认证（Authentication）



1. 发送者创建消息 M
2. 用 SHA-1 创建消息的 160bits 散列码 H
3. 用发送者的私钥对散列码 H 做 RSA 加密，结果追加到消息 M
4. 接收者用发送者的公钥进行RSA解密并恢复出散列码H
5. 接收者对消息M生成新的散列码，并与H比较，如果匹配，则消息被认证

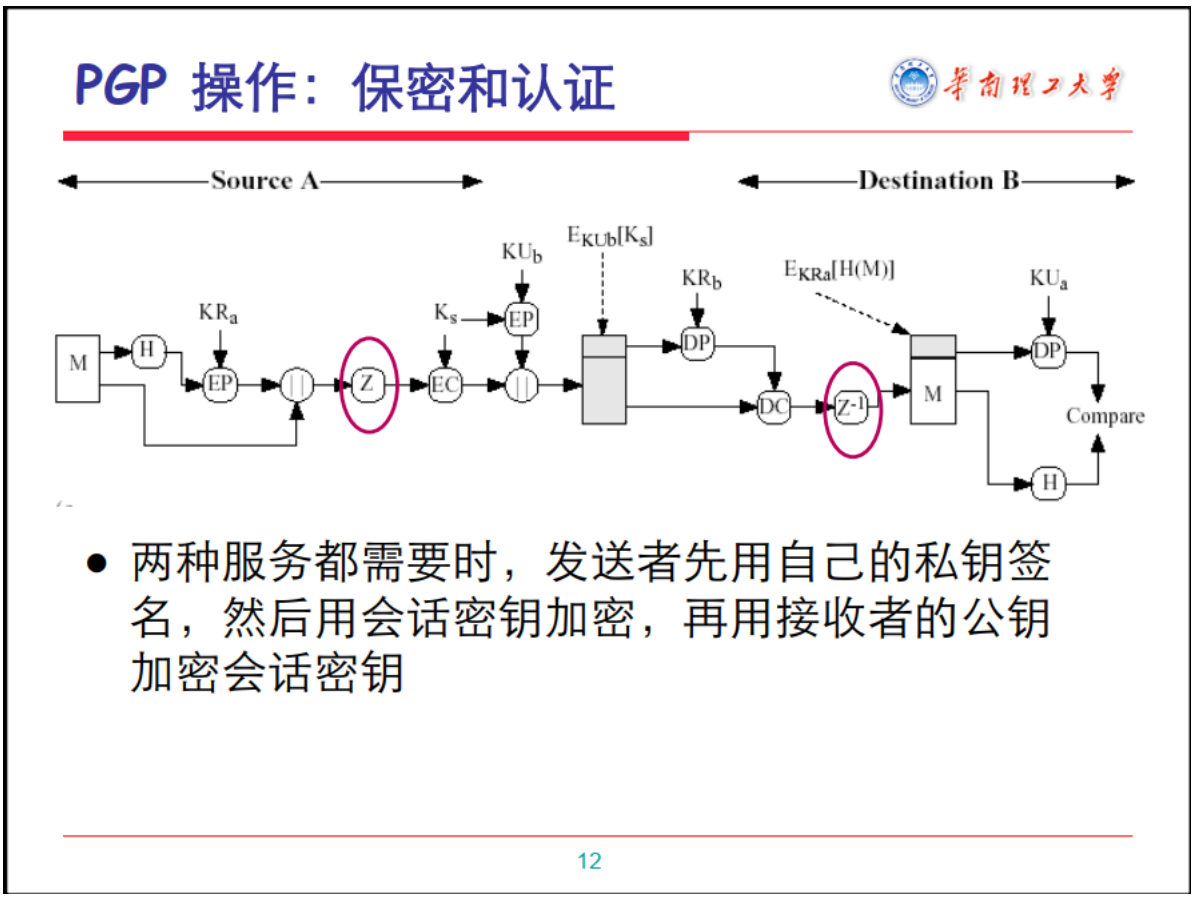
PGP操作：保密

PGP 操作： 保密（Confidentiality）



1. 发送者生成消息 M 及 128bits 的随机数做为消息的会话密钥
2. 用该会话密钥加密 M
3. 用接收者的公钥加密会话密钥，追加到消息的末尾
4. 接收者用私钥解密，恢复出会话密钥
5. 用会话密钥解密和恢复消息M

PGP操作：保密和认证



1. PGP操作：压缩
    - 使用ZIP压缩算法
    - 压缩的位置：发生在签名后，加密前
    - 压缩之前生成签名：
        - 验证时无需压缩
        - 压缩算法的多样性
    - 在加密之前压缩：压缩的报文更难分析
    - 对邮件传输或存储都有节省空间的好处

S/MIME

1. S/MIME是对电子邮件格式的安全扩展
2. 与PKI的结合，使用X.509证书，以及PKCS标准

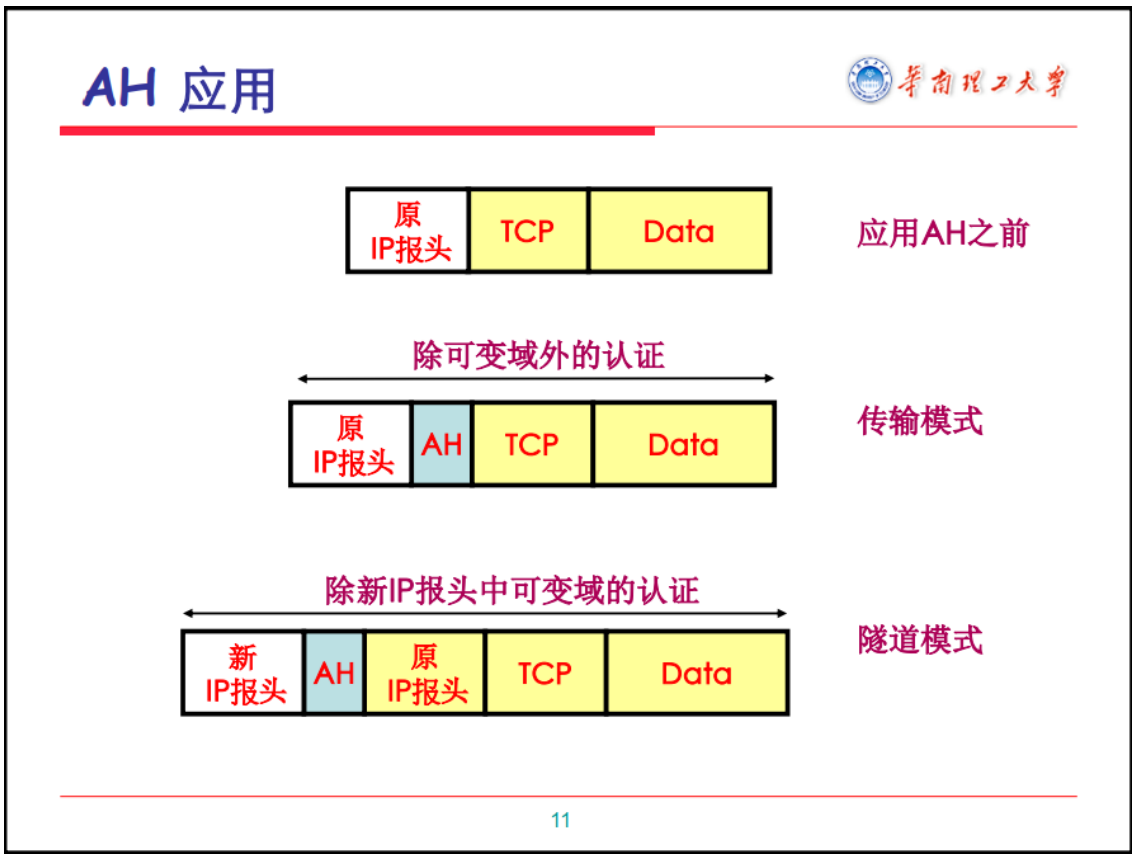
安全电子邮件配置

IP安全性

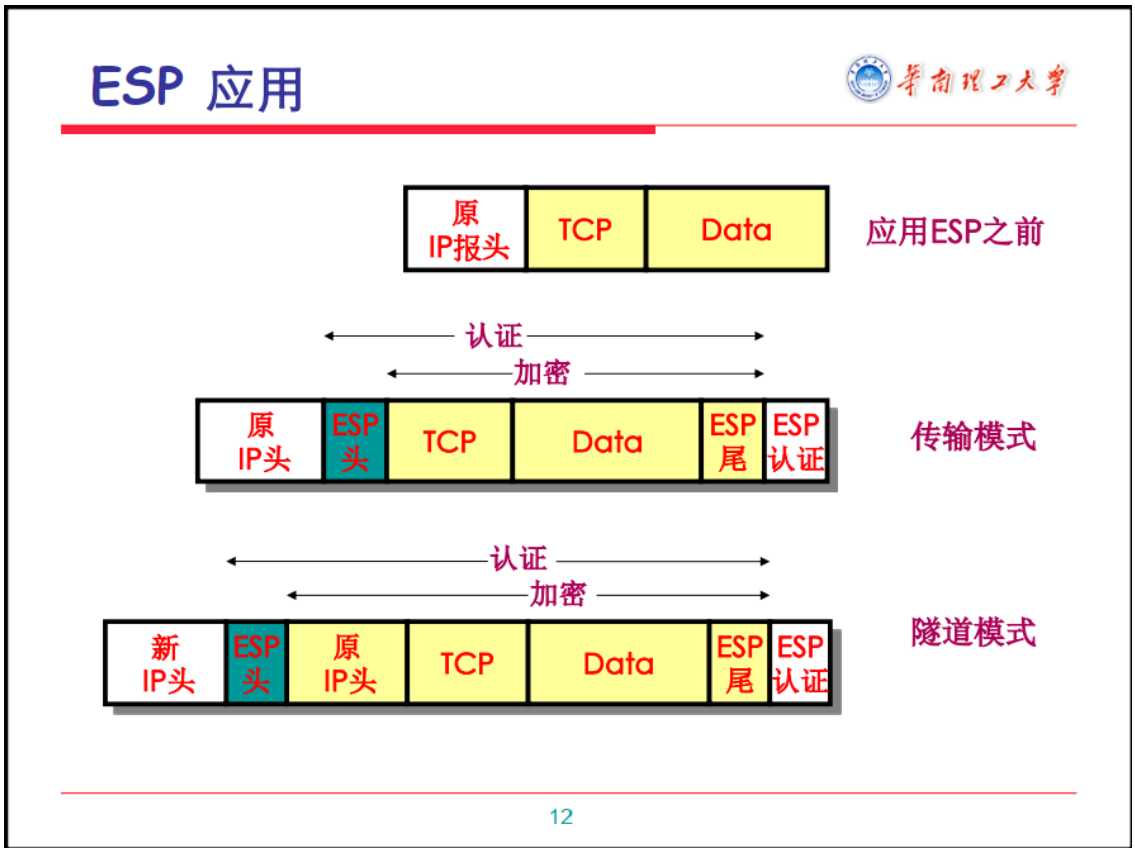
1. IPSec
    - IP层的安全包括了3个功能域：鉴别、机密性和密钥管理
    - 重要概念：
        - 鉴别报头（AH），封装安全有效负载（ESP）
        - 传输模式，隧道模式
        - 安全关联（SA），安全关联组（SA Bundle）
        - ISAKMP

2. IPSec概述
    - IPSec提供了在局域网、专用和公用的广域网（WAN）和Internet上安全通信的能力
    - 基于IP网络
    - 由IETF正式定制的开放性IP安全标准
    - 虚拟专网（VPN）的基础
3. IPSec应用
    - Internet上的安全分支办公室连接
    - Internet上安全的远程访问
    - 与合作者之间建立企业内部和外部的连接
    - 增强电子商务的安全性
4. IPSec的优点
    - IPSec在传输层之下，对于应用程序来说是透明的
    - IPSec对中断用户来说是透明的，因此不必对用户进行安全机制的培训
5. IPSec体系结构

6. AH（鉴别报头）：认证



7. ESP（封装有效载荷）：认证和保密性（保密必选，认证可选）



8. 传输模式的加密
    - 传输模式用以加密和可选地认证IP数据
        - 数据被保护，但头信息不加密
        - 可对传输包进行流量分析
        - 适用于ESP主机到主机的通信
9. 隧道模式的加密
    - 隧道模式加密整个IP包

        - 对于下一跳加一个新头
        - 适用与VPN，网关到网关
10. 安全关联SA
    - 在发送者和接受者之间的一种单向关系
    - 由三个参数来标识
        - 安全参数索引（SPI）
        - IP目的地址
        - 安全协议标识符

Web安全

SSL/TLS

1. SSL/TLS被设计用来使用TCP提供一个可靠的端到端安全服务，为两个通讯个体之间提供保密性和完整性（身份认证）
2. SSL体系结构：SSL协议栈

    SSL协议栈
3. 两个主要协议：
    - SSL记录协议
        - 建立在可靠的传输协议（TCP）之上
        - 用来封装高层的协议
        - 用来提供连接安全性：
            - 保密性：用握手协议定义的共享的保密密钥对SSL有效载荷加密
            - 消息完整性：用握手协议定义的共享的保密密钥形成MAC
    - SSL握手协议
        - 客户和服务器之间相互认证
        - 协商加密算法和密钥
        - 提供连接安全性
4. SSL连接与会话的区别
    - SSL连接
        - 一个连接是一个提供一种合适服务类型的传输
        - SSL的连接是点对点的关系
        - 连接是暂时的，每一个连接和一个会话关联
    - SSL会话
        - 一个SSL会话是在客户与服务器之间的一个关联
        - 会话由握手协议创建，定义了一组可供多个连接共享的密码安全参数
        - 会话用以避免为每一个连接提供新的安全参数所需的昂贵协商代价
5. SSL记录协议的操作
    - 分段
    - 压缩
    - MAC计算
    - 加密
    - 封装：加上SSL头
6. SSL记录格式：
    - 内容类型
    - 主版本、从版本
    - 压缩长度

    - 加密数据分片
7. SSL握手协议
    - 允许服务器和客户机
        - 互相认证
        - 协商加密和MAC算法
        - 协商使用的加密密钥
    - 由四个阶段组成：
        - 建立安全能力
        - 服务器认证和密钥交换
        - 客户端认证和密钥交换
        - 结束
8. **SSL握手协议流程**
    - **交换hello信息**
    - **交换必要的密码参数，以便双方得到统一的pre-master secret**
    - **交换证书和相应的密码信息，以便进行身份认证**
    - **产生master secret**
    - **把安全参数提供给SSL记录层**
    - **检验双方是否已经获得同样的安全参数**

    目的：实现身份认证和密钥交换
9. SSL的安全性
    - 在对付重放攻击上，SSL协议为每一次安全连接产生了一个128位长的随机数作为链接序号
    - 安全的SSL至少需要128位对称密钥和1024位非对称密钥长度

安全电子交易SET

1. 提供的服务：
    - 为交易各方提供安全的通信
    - 通过使用X.509-v3数字证书提供新人
    - 由于信息只在需要的时间和地方提供，因而要确保私密性
2. 主要特性
    - 信息保密性
    - 数据完整性
    - 持卡人账号认证
    - 商家认证
    - 使用X.509 v3数字证书，提供信任
3. **双向签名（DS，Dual Signature）**
    - 出现原因：发给商家的**订购信息（OI）**和发给银行的**支付信息（PI）**需要相匹配
    - 定义：对OI和PI的hash的串接签名
    - 构造：
        - H：hash函数，如SHA-1
        - 数据：
            - OI：发给商家的定购消息
            - PI：发给银行的支付消息
            - PIMD = H(PI)：PI的支付摘要
            - OIMD = H(OI)：OI的定购摘要
            - POMD = H(PIMD || OIMD)
        - 最后顾客对POMD签名获得双向签名

        - $$DS=E_{KRc}[POMD]$$
    - 验证：
        - 商家：
            - 获取PIMD，OI，DS
            - 验证$$E_{KUc}(DS)=H(PIMD||H(OI))$$
        - 银行
            - 获取OIMD，PI，DS
            - 验证$$E_{KUc}(DS)=H(H(PI)||OIMD)$$
4. 购买申请
    - 顾客发送初始请求，请求商家和支付网关的证书
    - 商家生成初始应答消息，包括交易标识，并用私钥签名，消息还包含商家的签名证书和支付网关的密钥交换证书
    - 持卡人通过CA验证商家和网关的证书，并生成OI和PI，并生成一次性的对称加密密钥Ks
    - 持卡人发送购买请求，包括
        - **购买相关信息**
            - **PI、DS、OIMD，并用Ks加密**
            - **数字信封：包括用支付网关公开交换密钥加密的Ks**
        - **定购相关信息**
            - **OI、DS、PIMD**
        - **持卡人证书**