

# 网安期末复习

## lect1 为什么学

### 无处不在的通信

基本要素：信源、信道、信宿

一个世纪前电信网 半个世纪计算机网络 单向广播有线电视网——三网合一

### 电信网的发展历程

有线电报、电话；无线电报；寻呼；蜂窝移动；手机

**1830s 电磁电报机**

**morse code**

滴、答、空。1、3、滴答间1字符间3单词中7

送话器：麦克风，电磁感应震动金属片or液体电阻or活性炭震动

铜线传输；1876贝尔专利，双绞线

1878 交换机从一对一到一对多，人工接续

1879 电话号码，1893步进交换 1938纵横交换

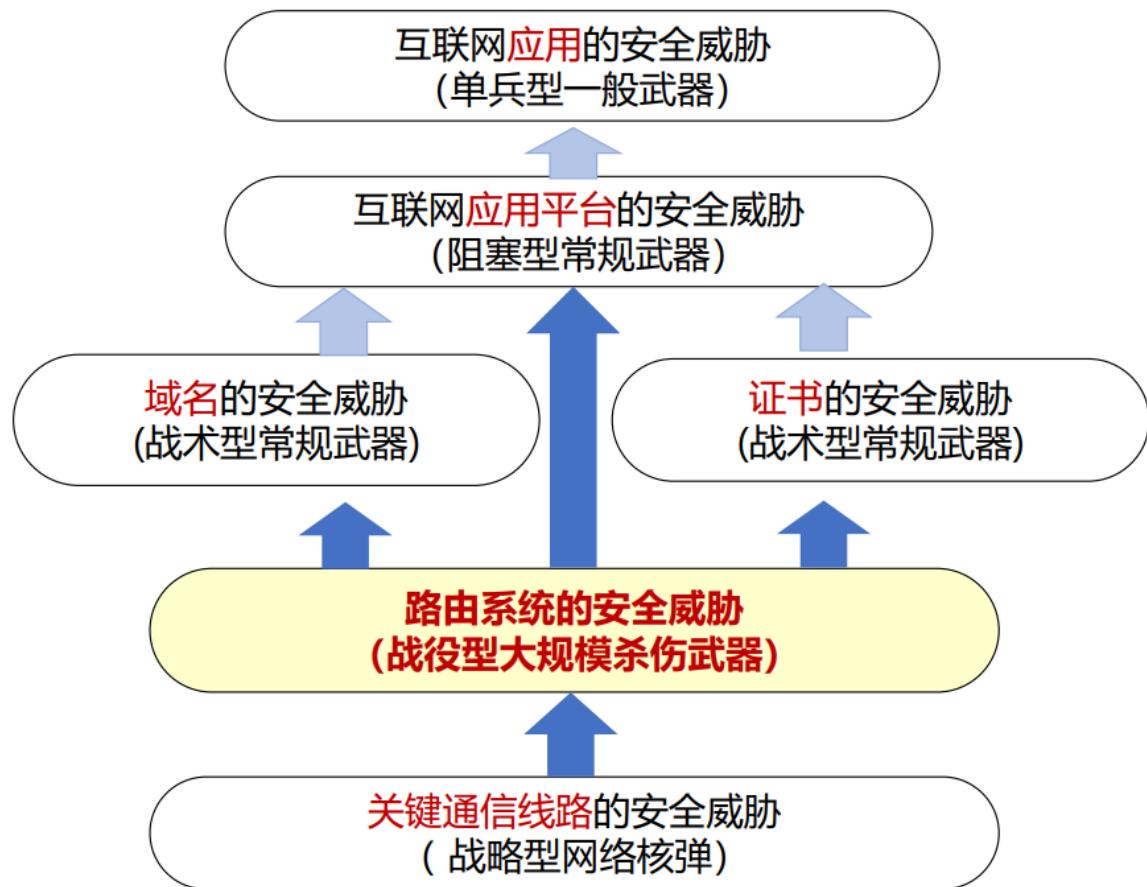
1946 电子计算机 1965程序控制交换机

交换机类型	接续方式	控制方式	交换信息
供电制交换机	人工	环路电流	模拟话音业务
步进制交换机	自动	拨号脉冲	模拟话音业务
纵横制交换机	自动	布线逻辑	模拟话音业务
模拟程控交换机	自动	存储程序	模拟话音业务
数字程控交换机	自动	存储程序	数字话音、数据图文传真等

俄国人波波夫和意大利人马可尼各自独立完成了无线电通信

蜂窝移动通信系统主要是由交换网路子系统（NSS）、无线基站子系统（BSS）和移动台（MS）三大部分组成

### 无处不在的网络安全



## 信息窃取攻击

**蛮力** 常用密码做字典，撞库：用获取的账户密码去登陆更大型的信息系统

**中间人** 在受害者网络链路中数据监听和篡改；链路传输截取，并修改来欺骗两端

**拒绝服务** 使计算机崩溃压垮来组织服务，最容易有效，最多，影响网络可用性。SYN洪泛；UDP洪泛；DNS反射

**病毒、恶意软件** 传播到用户系统上，有时利用漏洞偷偷运行，伪装成正常程序欺骗用户。浏览、email、移动介质、下载

## 防守技术

- 加解密：中间人攻击
- 访问控制、身份认证：阻止非法访问
- 防火墙：阻止大部分外来非法请求
- 防病毒：识别坏东西并阻止
- 补丁：修复漏洞

## 揭开现象看本质

Enigma 一战用于战争，1926装备德国军队

有键盘转轮和显示器，没有空格和标点

制造：1918, Arthur Scherbius

破译：Alan Turing, Bomber

转轮机——防止字频统计

代换密码算法多表代换

每次输入后转轮转动一格，同一输入对应的密文不同

三个转轮后反射器，反射器不转动且字母为绑定的双射。相当于划分26字母到正反两个空间

加解密对应转轮情况一样，即表一样

加密：三次正翻译，取反，三次反翻译

解密：三次正翻译，取反，三次反翻译  
使得译码和编码过程相同

## 使用过程

- 调节三个转轮初始方向
- 明文打键盘到显示器的密文闪亮并记录，电报发出
- 接收方打开同样的Enigma，转轮初始化键入密文出现明文

## 举例

- 初始化转轮。
- 输入A，转轮向前（左，反字典序）转动，七次查表翻译D
- 再次输入A，转三个轮，七次查表得H
- 这个例子只有第一个轮在转动？

## 加强

- 如果得到机器（三个固定表）则秘钥为初始方向，可以蛮力
- 滚轮可拆卸并组合
- 键盘和转轮1之间有“连接板”，错位连接一些字母成为另外的信号。最多相当于两两交换六对字母

# lect2 密码学 I

---

## 基本概念

不同于隐藏（保险柜和密信），不同于访问控制（口令没有加密安全性很弱）

## 加密形式

- 传统（对称、单钥）
  - 代换、置换、二者组合
  - 安全性在于保持算法本身保密
    - 不适于大规模生产以及变动大
    - 用户不了解算法安全
- 公钥（现代、非对称）
  - 算法公开；密钥保密
  - 安全性在于密钥的保密性

## 概念

- 密文cipher 明文message有时候为passage
- 密码算法：Encoder Decoder
- 密钥Key，在ED中作为下标变量
- 编码学与分析学：加密方案和破译
- 编码学的独立特征
  - 转换明文为密文的运算类型
    - 基于置换：重新排列明文元素
    - 代换：映射成另外的元素
    - 不允许丢失信息，运算可逆

- 所用的密钥数
  - 发送接收相同密钥：对称、传统
  - 不同：非对称，公钥
- 处理明文的方法
  - 分组密码、块密码：每次处理一个输入分组对应一个输出分组
  - 流密码，序列密码：连续处理输入元素，每次输出一个元素

简单加密：异或。CPK任何两个异或求出另一个

- 无条件安全（信息论安全）：有多少密文（无限的计算能力）都不能唯一确定由该体制产生密文所对应的明文
  - 每次都换密钥，绝对安全
- 计算安全
  - 破译密码代价大于数据价值
  - 破译时间超过信息声明期

## 发展周期

- 古典密码：艺术，需要算法保密。都是针对字符的代换置换，简单的分析手段
- 近代密码：有计算机；成为科学；有线电报产生现代编码学；无线电报产生现代密码分析学；
  - 加密体系安全性不依赖加密方法本身，而是依赖于所使用的的密钥
  - shannon发表论文成为科学
- 1976 新方向：公钥密码体制，从DES开始

## 古典密码（代换）

### Caesar 凯撒密码

每个字母换成后面第三个字母

key：往后换key个，25种穷举即可

#### 升级

- 密钥次密码：一个密钥次放前面剩下按顺序。

明文	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
密文	T	S	I	N	G	H	U	A	B	C	D	E	F	J	K	L	M	O	P	Q	R	V	W	X	Y	Z

单表代换

评价 明文的语法模式和结构有多少保留在密文，减少这样的保留信息：

- 对多个字母一起加密
- 多表代换

### Playfair密码

多表代换

key: 密钥词构成5\*5字母矩阵，IJ合并，按行填写。

## Playfair密码基于一个由密钥词构成的5×5字母矩阵

- 举例：密钥词是monarchy

构造密钥词的方法是：

- 密钥词从左至右、从上至下填在矩阵格里，再将剩余的字母按照字母表的顺序从左至右、从上至下填在剩余的格子里
- 字母I和J当作一个字母

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

加密：一次两个，相同字母II变成Ix (x为填充字符)

- 同行，向右一个单位
- 同列，向下一个单位
- 四方对角线替换

? 如果是ij密文输出什么?

评价 完整保留了明文语言结构，几百个密文可以分析规律

## Hill密码

多表代换

m个连续的明文字母带换成m个密文，由m线性等式决定，每个字母都是数字0~25。

key: m\*m矩阵，解密用K逆矩阵——并不所有矩阵都可逆。

评价 隐蔽单字母频率，矩阵越大隐藏信息越多。33矩阵隐藏了双字母的频率

## Vigenere密码

一系列凯撒，多表代换

- 相关的单标代换规则集，由密钥决定具体变换

加密过程

- 设明文为：ATTACKATDAWN；密钥为：THUCS

明文	A	T	T	A	C	K	A	T	D	A	W	N
密钥	T	H	U	C	S	T	H	U	C	S	T	H
密文												
明文	A	T	T	A	C	K	A	T	D	A	W	N

密钥	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	U	V	W	X	Y	Z
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q

第一个密钥字母加密明文的第一个字母，第二个密钥字母加密明文的第二个字母，等所有密钥字母使用完后，密钥又再循环使用。

第一步可以先将密钥的长度拓展到（或者缩减到）和明文长度一样。

查表 or 计算 (Mi+Pi即可)

评价 强度在于每个明文对应多个密文，且使用唯一字母作为密钥，来掩盖频率。

- 字母频率隐藏

- 并非所有明文结构信息都被隐藏

## Vernam密码和一次一密

与明文毫无统计关系且一样长，基于位的二进制异或运算得到密文。

改进：与消息一样长且无重复的随机密钥——一次一密OTP，直接相加mod26就好

评价：不可攻破但是没屁用

- 大规模随机密钥很难
- 密钥的保护和分配很难（安全信道&用后销毁）

## 古典密码（置换）

新的排列

简单栅栏技术：对角线写入明文，行读出为密文

写成矩阵块，按列读出并打乱列次序。列次序为密钥。

单步or多步置换

### 简单置换

逐行写入矩阵块（不足行补进随便字母xyz），按列次序读出

key：列次序如4312567，即先读1所在列...

多换几次后，没什么规律了

### 转轮机（多层加密）

包括一组相互独立的旋转轮，电脉冲可以通过；每个圆筒有26个输入引脚和26个输出引脚，并且一一相连

每个圆筒就定义了一个单表代换，多个圆筒就是多表代换

三轮\四轮Enigma

## 古典密码破译举例

穷举、蛮力：不适用一次一密，单纯尝试看明文有无意义

频率分析：猜字，最常出现e，q后面u，还有双码、多码统计特性等等

单表代换破译过程举例：

- 词频统计，先换最高为e, t
- 开始根据三元组（the）、语法规则was been等猜词

## 对称密码算法

加解密密钥相同，密钥用保密信道分配

常用：DES、IDEA、RC245、AES、CAST-128、Blowfish

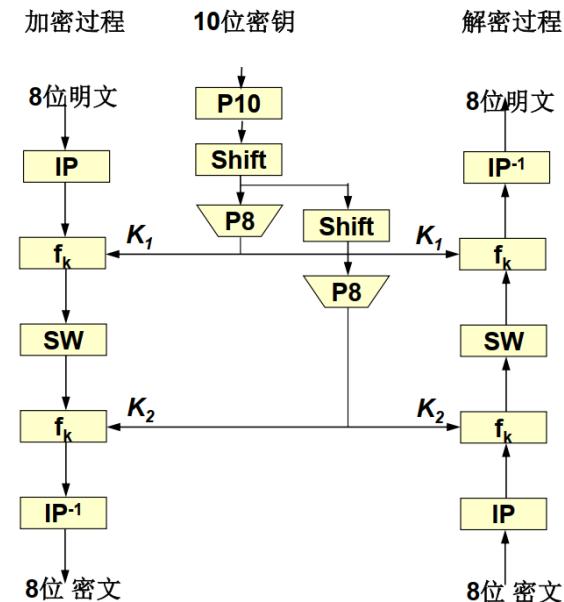
### 对称密钥密码S-DES（简化DES）

输入：10bit key 8bit明文组

输出：8bit密文组，破译过程相反

- 加密算法包括5个函数：
- 初始置换IP
- 复杂函数fk，包含了置换和代换运算，并且依赖于密钥
- 用于转换数据两个部分的简单置换函数SW
- 再一次的复杂函数fk
- 置换函数IP的逆函数IP-1
- 密文=  $IP^{-1}(fk_2(SW(fk_1(IP(\text{明文}))))))$
- 明文=  $IP^{-1}(fk_1(SW(fk_2(IP(\text{密文}))))))$
- $K_1 = P8(\text{Shift}(P10(\text{key})))$
- $K_2 = P8(\text{Shift}(\text{Shift}(P10(\text{key}))))$

## S-DES的整体结构

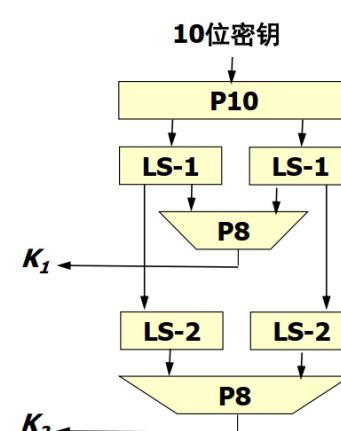


包含函数：

- 初始置换IP
- 复杂函数fk：包含置换代换，依赖key
- 简单置换SW
- 复杂函数fk
- 初始置换地函数IP^-1

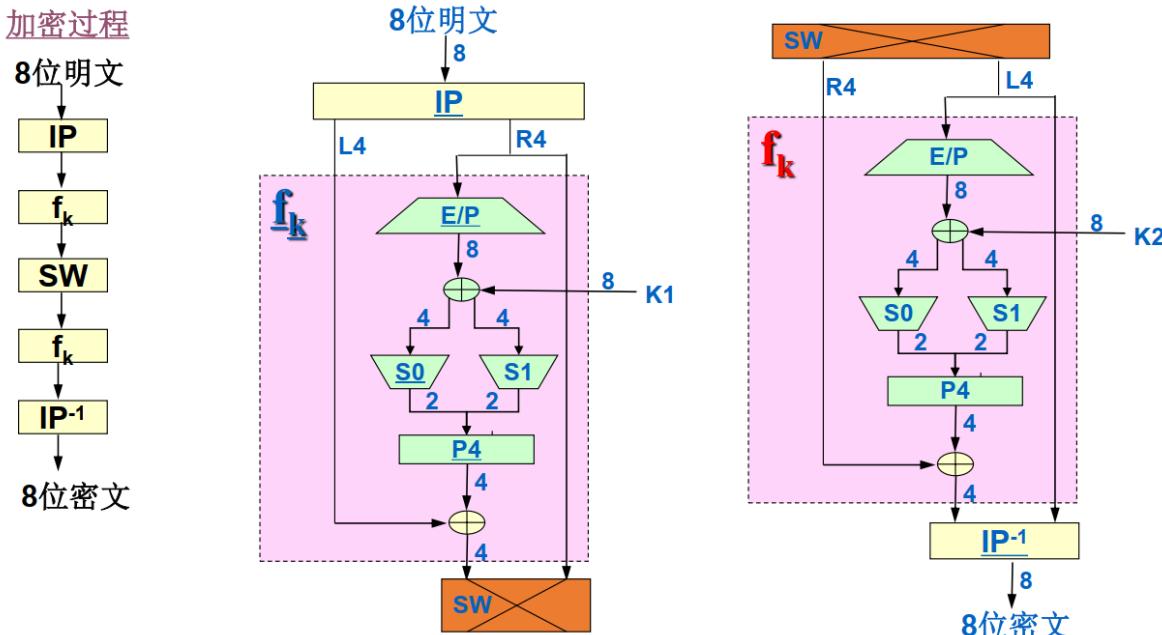
密钥拆分：其实是把10位key拆成两组8bit K1K2用：

- S-DES依赖于收发双方共享的10位密钥，它产生的两个8位子密钥分别用在加密和解密的不同阶段。
- 10位密钥即( $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}$ )。
- 置换P10定义为  
 $P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$
- LS-1表示前5位和后5位分别循环左移1位。
- LS-2表示前5位和后5位分别循环左移2位。
- 置换P8定义为  
 $P8(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_6, k_3, k_7, k_4, k_8, k_5, k_{10}, k_9)$
- 由此可以计算得到子密钥K1和K2。



- 10bit K 置换P10
- 前后两组循环左移1位，放入P8得K1
- 前后两组循环左移2位，放入P8得K2

加密过程: IP fk SW fk IP<sup>-1</sup>



- 先IP, 初始八位置换26314857
- K1作fk:  $f_k(L, R) = \{(L \text{ xor } F(R, Key)) , R\}$ ,  $F(R, key) = P4(S0S1\{EP(R) \text{ xor key}\})$
- 扩展, 密钥, S盒, 4置换
  - L4R4拆分, R4为输出后4
  - R4作扩展置换EP: 41232341
  - 与K异或
  - L4R4拆分, 分别放入S盒得2\*2位输出, 合并为4bit
    - S盒 4x4矩阵, 取值在0~3, 1、4位决定行, 2、3位决定列, 查表
    - 置换P4: 2431
    - 与刚开始L4异或为输出的前半边, 后半边为R4
- 交换函数SW, 前后4互换
- 用K2重复fk
- 末尾置换IP<sup>-1</sup>: 41357286
  - 置换求逆: 排列回来

加解密过程完全一样且对称

## Feistel密码结构

Vigenere和Vernam为流密码

验证: 改变分组大小不改结果, 不整除分组也不用补字母

分组密码: 64|128bits, 基于网络

Feistel建议:

- 使用**乘积密码**来逼近简单代换密码: 依次使用两个或以上的基本密码, 所得密码强度将强于所有单个密码的强度
- 交替使用代换置换

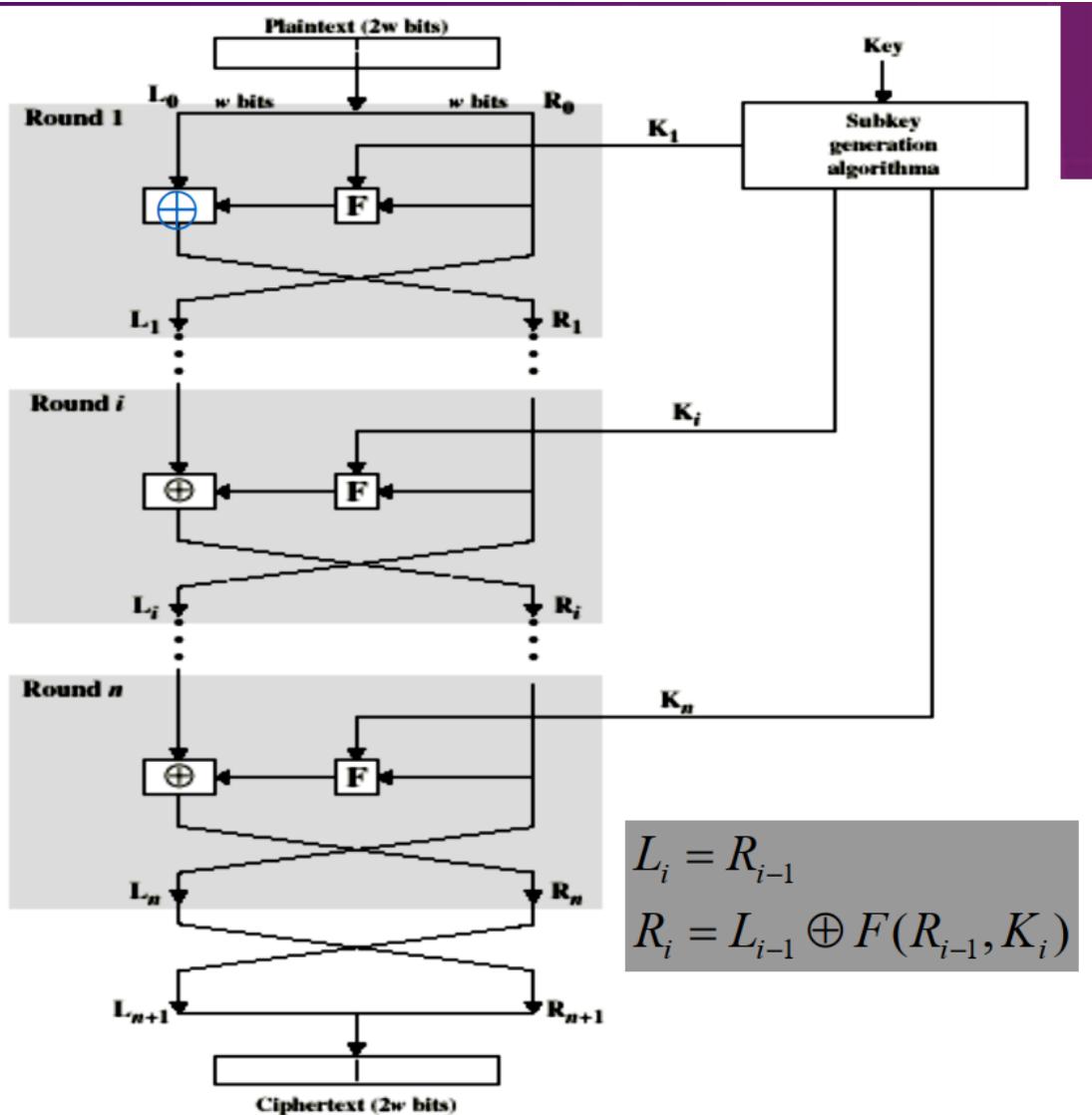
shannon:

- **扩散** 明文统计特征消散在密文中, 每个明文影响多个密文

- 混淆 密文和密钥的统计关系复杂，难以推导密钥

输入： $2w$  bit明文组，密钥 $w$  bit  $K_i$

- 明文分开为 $w$ bit  $L$ 和 $R$ ，经过 $n$ 轮迭代组合成密文
- 每轮迭代的输入都是上层的输出，但结构相同
  - $L_0$ 和 $F$  ( $K_1$  R0) 异或成 $R_1$
  - $R_0$ 直接成为 $L_1$
- 子密钥 $K_i$ 由 $K$ 推出，有 $K_i$ 生成算法



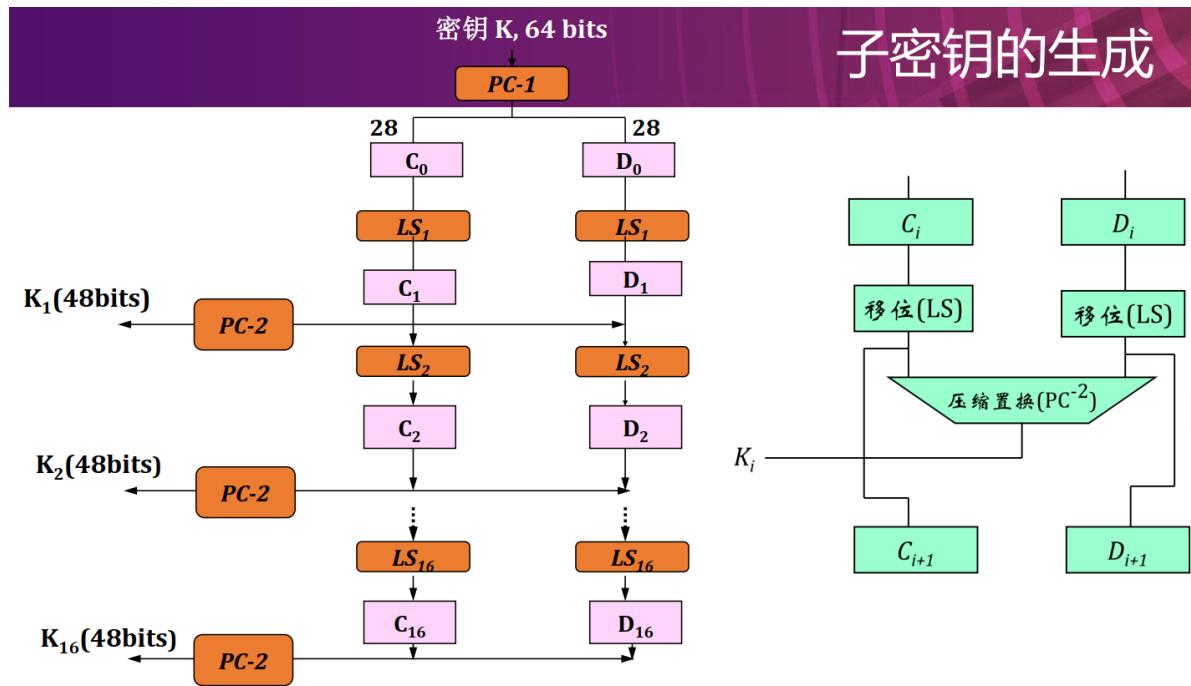
**Figure 2.2 Classical Feistel Network**

评价：

- 分组长度和密钥长度：越长越安全，但是慢
- 迭代轮数：多轮安全
- 子密钥产生越复杂越难以密码分析攻击
- 轮函数复杂，抗攻击强

**DES**

分组加密，对称密钥，56bits Key（奇偶校验后写成64bits），分组64bits，标准算数逻辑运算  
加密过程：



- 密钥生成器：
  - 56+8的密钥，拆分为28+28
  - 两个28根据16轮，不同循环左移1or2位
  - 置换选择PC-2 56选48bits成为Ki
  - 两个28根据16轮，不同循环左移1or2位
  - 置换选择PC-2 56选48bits成为Ki+1
- 明文分64bits块m
- IP初始置换64换64
- T1, 16轮迭代过程和Feistel相同
  - 具体：R0直接成为L1
  - F (R0, K) E扩展，K异或，S选择，P置换
    - 32bR0 E盒扩展至 48b
      - 输入一位影响下一步的两个替换，输出对输入依赖传播快，密文每一位都依赖明文的每一位
    - 和K1异或
    - S盒代换选择，48选到32b
      - 8个6选4的S盒
      - 首尾决定行，中间4b决定列
      - 需要4x8矩阵，取值0~15
    - P盒置换为32
  - 和L0异或成为R1
- T2
- ...
- T16
- IP-1末尾置换64换64

解密：IP T16: 1 IP-1, 完全对称过程

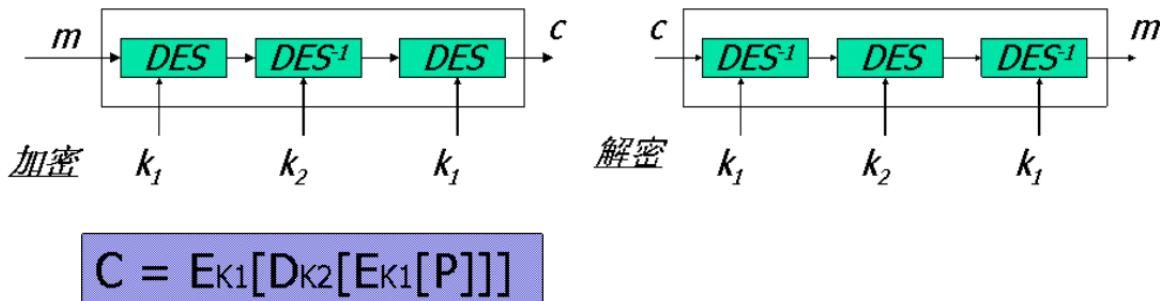
## 其他常用对称密码

3DES, blowfish, RC5, AES (取代DES)

- 高密码强度
- 广泛用于Internet
- 代表DES以来的对称密码

### 3DES

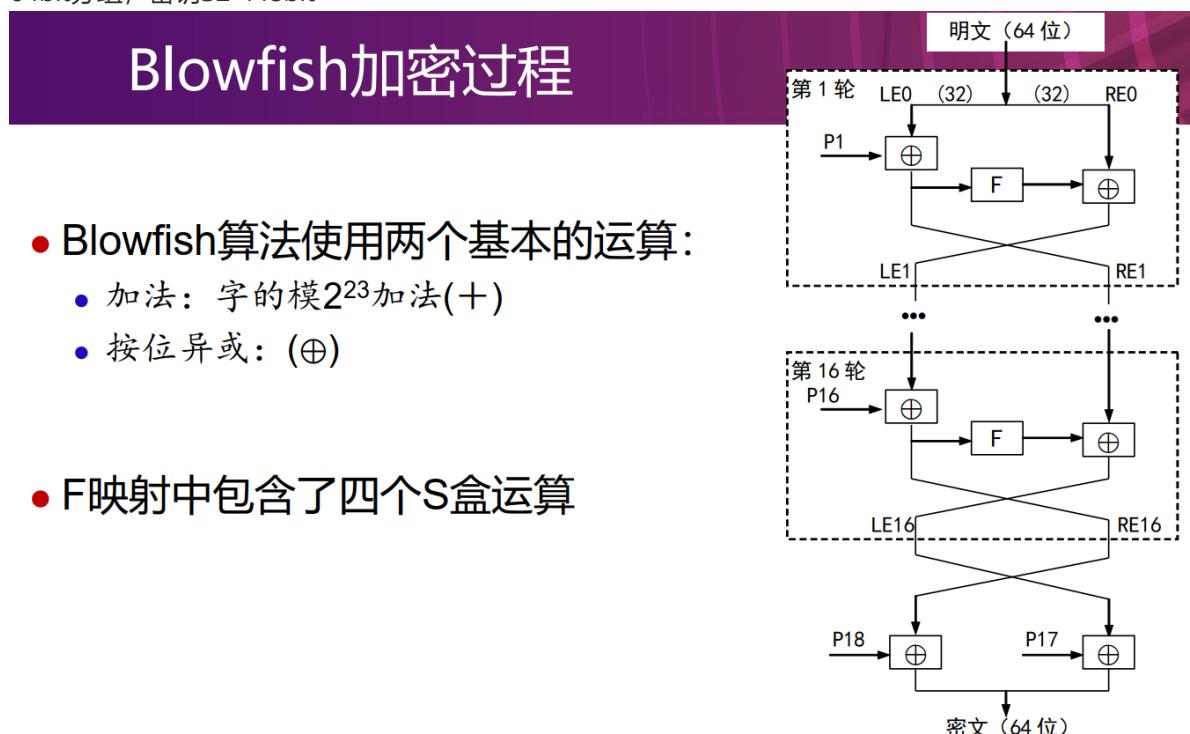
两密钥112bit三重DES，明文攻击代价 $2^{112}$



### blowfish

快速紧凑简单安全可变

64bit分组，密钥32-448bit



- Blowfish算法使用两个基本的运算：

- 加法：字的模 $2^{23}$ 加法(+)
- 按位异或：( $\oplus$ )

- F映射中包含了四个S盒运算

- 子密钥和S盒都是blowfish本身生成，数据不可辨认，密钥分析困难
- 每轮对数据左右同时运算，和古典feistel不同，密码强度增强
- 448bit密码抵抗穷举

### RC5

快速；迭代和key长可变；简单；高安全

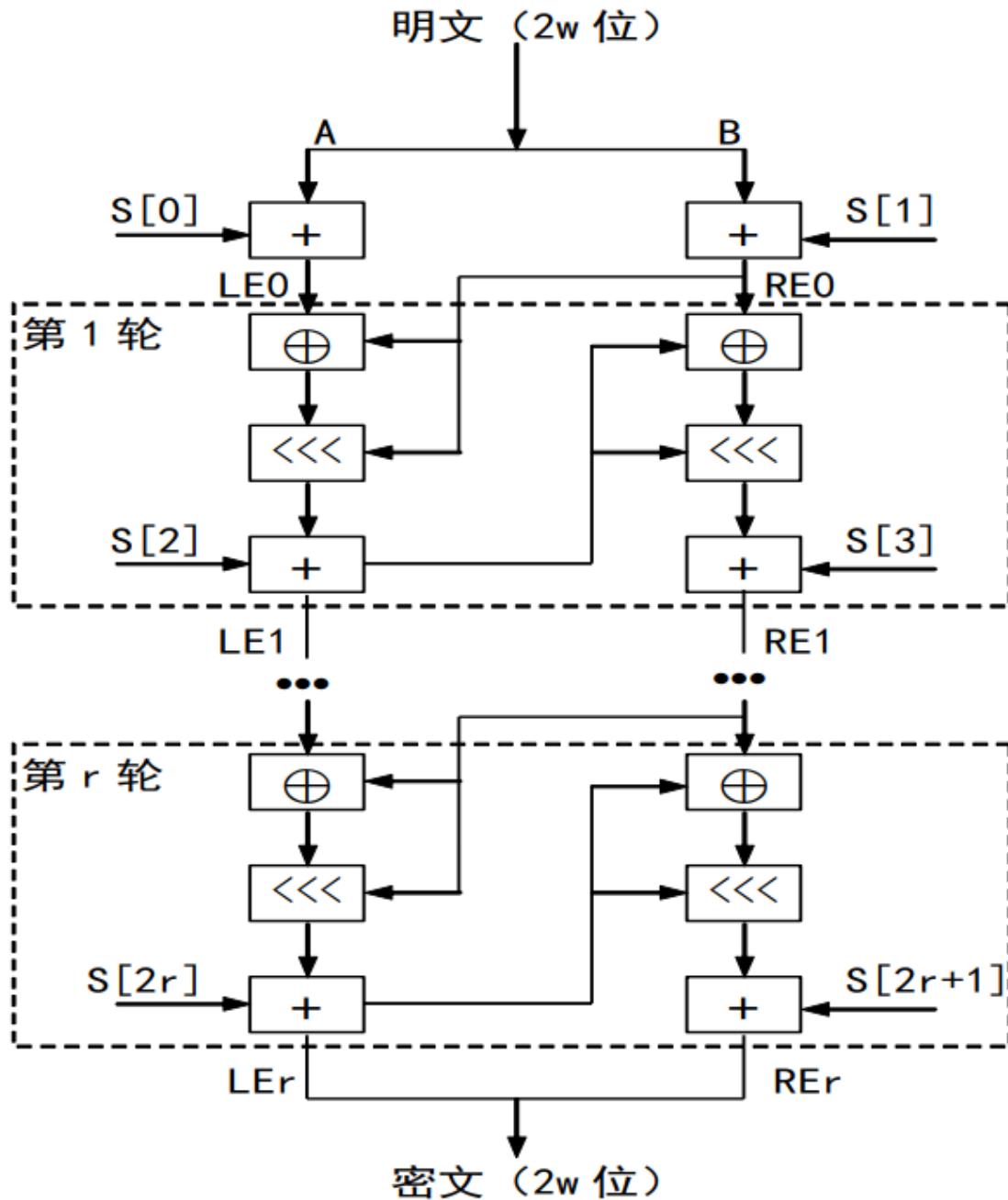
分组32、64、128

密钥0~2040b

由分组长度，迭代次数，密钥的byte数决定的算法族

使用：

- 字的模 $2^w$ 加法
- xor
- 循环左移



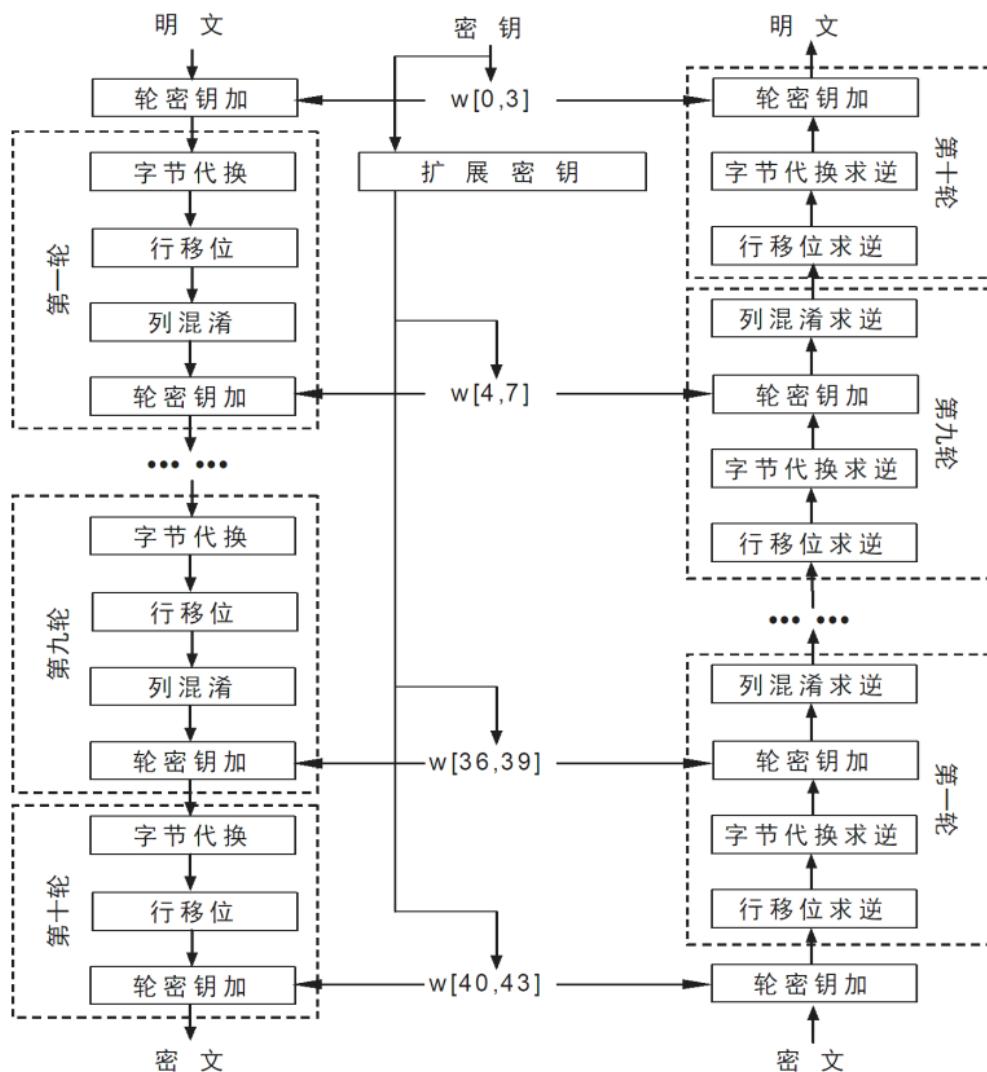
## AES

3des太慢，分组太短——分组128支持128192256密钥的AES

w 128 key128

- 免疫所有已知攻击
- 所有平台执行快代码紧凑
- 密钥被扩展为44个wordlen = uint32的数组，每轮有4个字参与
- 轮密钥加是 vernam 密码
- 设计结构简单，
  - 轮密钥加

- 九轮迭代
  - 字节代换
  - 行移位
  - 列混淆
  - 轮密钥加
- 最后一轮
  - 字节代换
  - 行移位
  - 轮密钥加
- 不是Feistel结构
- 每轮四个阶段，一个置换3个代换，均可逆
  - 字节代换：用一个S盒完成分组中的字节代换
  - 行移位：一个简单的置换
  - 列混淆：算术代换
  - 轮密钥加：利用当前分组和扩展密钥的一部分进行按位异或



算法	密钥长度	迭代次数	数学操作	应用
DES	56	16	XOR, S-Box	Kerberos, SET
3DES	112 or 168	48	XOR, S-Box	PGP, S/MIME
IDEA	128	8	XOR, +, ×	PGP
Blow Fish	最大448	16	XOR, S-Box, +	
RC5	最大2048	<255	+, -, XOR	
CAST-128	40—128	16	+, -, S-Box	PGP

## lect3 密码学 II

### 非对称密码原理

缺陷：

- 密钥需要安全信道分配
- 无法用于数字签名？
- 密钥管理复杂， $O(n^2)$ 的数量

基于数学函数而非代换置换

概念：

- 明文、密文
- 会话密钥Ks：对称密钥
- 加密、解密算法
- 公钥U 公开，用来加密和验证签名
- 私钥R 保密，用于解密和签名

加密：

- 每个实体有一对密钥，接收方的公钥加密信息，私钥解密。
- 每方持有所有公钥和自己的私钥，向谁发就用接收方的公钥加密

签名：

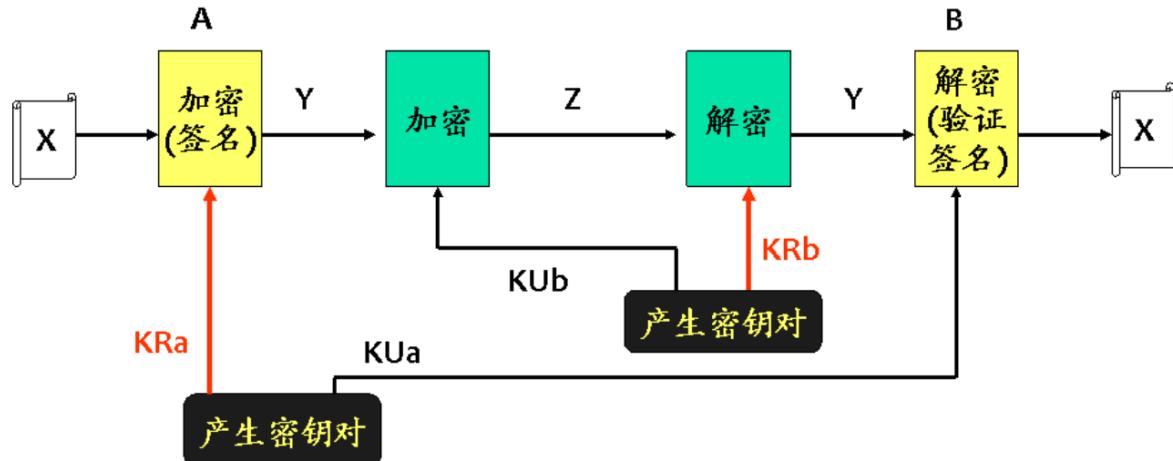
- 发送方用自己的私钥加密签名发出
- 接收方用发送方的公钥解密验证发送方

密钥交换：

- 协商会话密钥来对称加密

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

可同时使用加密和签名——先自己私签再对公加密



$$Y = E_{KRa}(X), \quad Z = E_{KUb}[Y] = E_{KUb}[E_{KRa}(X)]$$

$$Y = D_{KRb}(Z)], \quad X = D_{KUa}[Y] = D_{KUa}[D_{KRb}(Z)]$$

数学原理：陷门单向函数

不知道陷门信息求逆困难，知道后容易实现

- 给定 $x$ , 计算 $y=fx$ 容易
- 给定 $y$ , 计算 $x$ 为 $y=fx$ 困难
- 存在 $\delta$ , 有 $\delta$ 对 $y$ 容易计算 $x$  (陷门性)
- 对于加解密,  $f$ 为公钥 $\delta$ 为私钥

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

密码分析：

- 容易被穷举，利用公钥对所有可能的密钥加密，尝试密文是否一致
  - 应对：长密钥，不能太长
- 函数复杂性不是密钥长度的线性增长，指数或更快
- 仅用于密钥管理和数字签名

非对称vs对称比较

- 公钥并不更安全，安全取决于计算量
- 公钥并不通用，传统并不过时：公钥计算量太大，仅限于密钥管理和签名，不可能取代传统密码
- 公钥密钥分配很简单：也需要协议、中心代理，并不比传统密码更简单

## 数论基础

任意正整数可以唯一表示为一组素数的乘积

模运算，对加法乘法可交换结合分配（需再mod）

## 欧拉函数与欧拉定理

- 欧拉函数 $\phi(n)$ ：n是正整数,  $\phi(n)$  是比n小且与n互素的正整数个数
  - $\phi(3)=|\{1, 2\}|=2$
  - $\phi(4)=|\{1, 3\}|=2$
  - $\phi(5)=|\{1, 2, 3, 4\}|=4$
  - $\phi(6)=|\{1, 5\}|=2$
  - $\phi(7)=|\{1, 2, 3, 4, 5, 6\}|=6$
  - $\phi(10)=|\{1, 3, 7, 9\}|=4$
- 如果p是素数，则  $\phi(p)=(p-1)$
- 如果p,q是素数，则  $\phi(pq)=\phi(p)\phi(q)=(p-1)(q-1)$
- 欧拉定理
  - 若整数m和n互素，则  $m^{\phi(n)} \equiv 1 \pmod{n}$   
等价形式  $m^{\phi(n)+1} \equiv m \pmod{n}$
- 例如：
  - $m=3, n=10; \phi(10)=4; m^{\phi(n)}=3^4=81; 81 \pmod{10}=1$
  - 即： $81 \equiv 1 \pmod{10}; 3^{4+1}=243 \equiv 3 \pmod{10}$
- 推论：
  - 给定两个素数p, q， $p < q$ ，两个整数n, m，使得 $n=pq, 0 < m < n$ ；则对于任意整数k，下列关系成立： $m^{k\phi(n)+1} \equiv m \pmod{n}$

## RSA

分组密码，明文密文0~n-1，常用1024bit或309dec

### 密钥产生

- 密钥产生
  - 取两个大素数p, q，保密；
  - 计算 $n=pq$ ，公开n；
  - 计算欧拉函数 $\phi(n) = (p-1)(q-1)$ ；
  - 任意取一个与 $\phi(n)$ 互素的小整数e，  
即  $\gcd(e, \phi(n))=1; 1 < e < \phi(n)$
  - 寻找d， $d < \phi(n)$ ，使得  
 $de \equiv 1 \pmod{\phi(n)}$ ，即  $de = k\phi(n) + 1$
  - 公开  $(e, n)$
  - 将d保密，丢弃p, q。
- 公开密钥:  $KU=\{e, n\}$
- 秘密密钥:  $KR=\{d, n\}$

- 设： $p=7, q=17$
- 则： $n=119$
- $\Phi(n)=6 \times 16=96$
- 选择 $e=5$
- $5d=k \times 96+1$
- 令  $k=4$ , 得到  $d=77$
- 故知道：
  - $KU = \{5, 119\}$
  - $KR = \{77, 119\}$

公钥：小互素整数e，私钥： $de=k\phi+1$

## ● 加密过程

- 把待加密的内容分成k比特的分组， $k \leq \log_2 n$ ，并写成数字，设为M，则： $C = M^e \bmod n$
- 例如： $C = M^5 \bmod 119$

## ● 解密过程

- $M = C^d \bmod n$
- 例如： $M = C^{77} \bmod 119$

## ● 试证明：解密过程是正确的

## ● 证明：

$$\begin{aligned} M &= C^d \bmod n \\ &= (M^e \bmod n)^d \bmod n \\ &= M^{ed} \bmod n \end{aligned}$$

即  $M^{ed} \equiv M \bmod n$

## ● 根据欧拉定理推论： $M^{k\phi(n)+1} \equiv M \bmod n$ 得到 $ed = k\phi(n) + 1$

### 攻击

蛮力：尝试所有密钥

数学：素数乘积的因子分解

计时：记录计算机解密时长猜私钥，可以攻击很多公钥系统，仅依赖明文

### 安全性

大数素因子分解很难的

### 性能

软件硬件分别比DES慢100、1000倍

## DH密钥交换

两个用户安全的交换密钥，仅此

原理：计算离散对数非常难，**本原根**  $a$  几次方 mod 素数  $p$  可以从  $1 \sim p-1$  排列

对于整数  $b$  ( $b < p$ ) 和素数  $p$  的一个本原根  $a$ ，  
可以找到一个唯一的指数  $i$ ，使得：  
 $b \equiv a^i \pmod{p}$ ，其中  $0 \leq i \leq (p-1)$

$i$  称为  $b$  的以  $a$  为底 模  $p$  的离散对数或指数，  
记为  $\text{ind}_{a,p}(b)$

- $\text{ind}_{a,p}(1) = 0$ , 因为  $a^0 \pmod{p} = 1 \pmod{p} = 1$ ;
- $\text{ind}_{a,p}(a) = 1$ , 因为  $a^1 \pmod{p} = a \pmod{p} = a$ ;

对于  $b = a^x \pmod{p}$

- 已知  $a, x, p$ , 计算  $b$  是容易的
- 已知  $a, b, p$ , 计算  $x$  是非常困难的

密钥交换过程：

0. 其中的  $a, p$  中素数  $p$  最大
1. 选择底数  $a$  和素数  $p$  公开
2. Alice 和 Bob 各生成一个自己的私钥  $X_a, X_b$  保密
3. 计算  $a^{X_a} \pmod{p}$  和  $b^{X_b} \pmod{p}$  公开
4. 用对方的公钥乘以私钥次方得到共同密钥  $a^{X_a} b^{X_b} \pmod{p}$

## 其他非对称密钥算法

### DSA数字签名算法

也基于计算离散对数的难度

- 不能用于加密、密钥分配
- NIST 有后门
- 选择不公开，分析不充分
- 比 RSA 慢 10-40 倍
- 密钥 512 位太短
- 侵犯其他专利

### 椭圆曲线密码

- 有限域、容易构造运算器
- 加密快，小密钥高安全性

# 密码加密

加密选择：链路or端到端

链路：

- 数据链路层，在传输的每个相邻节点都加密再解密，使用不同密钥，
- 分配难度不大；无需额外数据；对用户透明
- 不适用于广播；链路节点中以明文存在；

端到端：

- 中间不解密，灵活的应用和层次选择
- 广播网可用；节点损坏不怕；便宜可靠；容易设计实现维护；没有同步问题；
- 容易被通信量攻击；分配密钥更难；不能掩盖地址

# 密钥分配

## 传统对称密码分配

1. A选择并亲自给B
2. 第三方选并亲自交AB
3. AB最近使用的密钥，加密一个新密钥发给对方
4. C与AB有秘密渠道，分别秘密发给AB

12人工传送适用链路

## 密钥分配中心KDC

AB想连接，都分别有自己和中心知道的密钥（主密钥）

1. A请求会话密钥保护与B的逻辑连接，有AB信息和N1
2. KDC用Ka加密消息响应A：一次性会话Ks；原始请求N1便于A响应；两项用Kb加密的给B内容（Ks和IDa）。
3. A存下Ks把后两项发给B（也可以理解为KDC发给A和B），结束

层次式KDC减少主密钥分配代价，本地KDC攻击破坏更小

## 公钥分配

- 用于分配公钥
- 公钥密码用于传统密码体制的密钥分配

## 公开发布

公钥直接发或者广播，方便但容易伪造

## 公开可访问目录

可信的实体组织维护、分配公钥目录

- 通讯方目录项 <name,ku>
- 通讯方可以注册、更新自己的公钥项；访问通讯录  
目录管理员的私钥破坏很大

## 公钥授权

- 公钥授权是一种更严格的方法，它的工作过程：

- A发送一条带有时间戳的消息给公钥管理员，请求B的当前公钥
- 管理员给A发送一条用其私钥KR加密的消息，A可以用管理员的公钥解密。这条消息中包含：B的公钥、原始请求、原始时间戳
- A保存B公钥，并将A的表示和临时交互号N1发给B
- 与A一样，B使用同样的方法从管理员那里得到A的公钥。
- A与B已经可以通信了，再通过核对临时交互号确认各自的身份后，安全的通信机制就建立了

- 公钥授权的缺点在于公钥管理员就会成为系统的瓶颈

- 只要用户之间通信，就必须向目录管理员申请对方的公钥

## 公钥证书

证书管理员

证书：公钥和其他信息，发给相应私钥的通讯方

通信方传递证书来传递密钥信息，其他方验证这个证书由管理员发出

- 任何通信方可以读取证书确定证书的拥有者的姓名和公钥；可以验证证书出自管理员而非伪造
- 只有管理员可以更新证书
- 各方可验证当前性（过期证书）

## 公钥分配传统密码

公钥太慢

最简单（树洞）

### 例如：A要和B通信，则执行：

- A产生公/私钥对{KUa,KRa}，并将含有KUa和A表示的消息发给B
- B产生秘密钥Ks，用A的公钥对Ks加密后传给A。
- A计算DKRa[EKUa[Ks]]得到秘密钥Ks。
- 这样A和B就可以利用Ks和对称密码进行安全通信。

被主动攻击

## 保密真实

“要真的是你就把我给你的信息再给我发回来~”

3次握手，AN1,N1N2,N2;

1. 互有公钥
2. A用B公加密（A, N1）给B
3. B私钥解密，用A公加密发送（N1, N2）给A
4. A解密看到N1，确认对方是B，B公加密个N2发给B
5. B解密看到N2确认是A
6. A可以选个Ks用Ub加密发过去

公钥密码分配密钥也需要KDC？通过主密钥实现会话密钥分配

# lect4 认证

凡认证必比较

## 消息认证基本概念

传输分析：双方通信模式

面向连接：频率和持续时间；

通用：消息数量和长度

伪装：欺诈者向网络插入消息

攻击者产生消息并声称来自合法实体

非接收方应答（收到或者未收到）

攻击很多：内容、顺序、计时、否认

泄密与传输分析——消息保密

伪装修改——认证

发送方否认——数字签名

接收方否认——数字签名和专门协议

**消息认证** 验证消息来源且未被修改

也验证顺序和及时，数字签名和协议技术等等

下层：产生认证符的函数

上层：将函数作为原语使接收方验证消息真实性

三类认证函数：

- 消息加密：整个密文
- 认证码MAC：消息和密钥的公开函数，产生定长值为认证符
- Hash函数，映射为hash值的公开函数

## 认证函数1：消息加密

AB共享密钥K，没人知道密钥就有保密性，解密后确认由A发出——并不绝对，容易伪造  
要求明文有易识别结构，不通过加密函数不能重复

加密前附加错误检测码（帧校验序列FCS或校验和）

先FCS后加密，解密后重算FCS：可认证（有效）

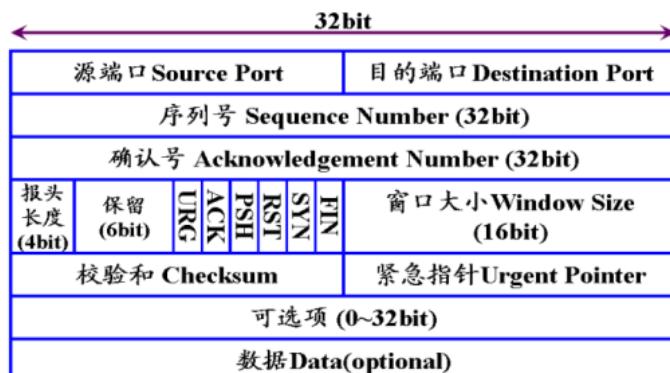
先加密再FCS，可混淆

TCP：

可以对除了IP报头之外的所有数据进行加密

如果攻击者利用一条消息代替加密后的TCP段，解密后也不能恢复出原来的IP报头

这种方法中，头不仅包含了校验和，而且还有序列号，用于防止延时、删除段和改变段的顺序。



## ● 对称加密

- 提供保密性
  - 只有A和B共享密钥K
- 提供认证
  - 只能发自A
  - 传输中未被改变
  - 需要某种数据形式/冗余
- 不能提供数字签名
  - 接收方可以伪造信息
  - 发送方可以否认信息

## ● 公钥加密

- 提供保密性
  - 只有B拥有用于解密的密钥KRb，但是任何人都可用KUb对消息加密并假称是A
- 提供认证和签名
  - 只有A拥有用于加密的密钥KRa
  - 传输中未被改变
  - 需要某种数据组织形式/冗余
  - 任何一方可用KUb来验证签名
- 提供保密性、认证+签名
  - 提供保密性（因为KUb）
  - 提供认证和签名（因为KRa）

## 认证函数2：MAC码

定长MAC = C(sharedK,Message), 不需可逆  
共享密钥，可认证，不能签名

消息+MAC后，整体加密：认证明文为持有K的人发出  
消息加密后，计算MAC附加：认证密文为持有K的人发出，裸奔的MAC？很好篡改MAC

app：

- 组广播，一个接受者验证并告知其他人
- 接受者随机进行消息认证
- 明文的计算机程序，不需要加解密，保护完整性来验证MAC
- 关心认证而非保密性
- 保密和认证分开，层次灵活：应用层认证，传输层保密性

## 认证函数3：Hash

单向hash类似mac，为message digest，具有检错能力

- 附加hash值后对称加密：一定来自A且未篡改，也保密
- 不要求保密时：仅对hash值加密，代价小，也可以认证。
- 用私钥对hash加密：认证和签名
- 私钥加密hash，对称密码加密整个：认证、签名、保密
- 含有共享salt：M加盐后哈希，附加在后面，明文传输：只有有盐的人可以验证通过，也认证
- 加盐的哈希后对称加密：保密+认证

加密的代价：

- 慢
- 硬件成本
- 小数据块更难以优化
- 加密算法专利
- 美国限制

Hash要求：杂凑散列

输入长度不限但输出固定

单向寻找

扛弱碰撞：找一个b与a哈希值相等

抗强碰撞：找xy，哈希相等

# hash一般结构

L个b长分组，最后不足就填充——包含长度信息

每个单元输入是之前的n位哈希（连接变量CV），b长度输入分组(Y)，通过压缩函数输出n为分组  
连接变量的初始值(CV0=IV)又算法开始指定，终值就是哈希

一般分组长度b>n所以是压缩

<https://codimd.s3.shivering-isles.com/demo/uploads/0d2a0805-246a-49b4-9891-af1d5cccefcc.png>

压缩函数抗碰撞——hash抗碰撞

## 主要hash算法

MD组，产生128bit的md

MD2：补成16bit倍数，加16bit校验和算hash

MD4：碰撞几分钟碰撞

MD5：王小云攻破

SHA：Secure Hash Algorithm改MD45，有012方案

RIPEMD-128/160/320

Haval

Gost

## MD5

512bit分组

128bit输出

简单，32位容易

步骤：

- 填充位到mod512=448，少64到整，一个1后面0
- 填充64位为消息长度mod2^64
- 初始化md缓存128bit 0~f f~0
- 以512分组处理，4轮\*16迭代
  - 每个压缩函数分四轮，128的连接变量向下传递，结合512 16字的输入
  - $T[i] = 2^{32} \text{abs}(\sin(i))$
  - 每轮中再进行16迭代：b根据a变化，其他向右挪一位

● MD5中每轮对缓冲区ABCD进行16步迭代，每步迭代为：

$$b \leftarrow a + ((a + g(b,c,d) + X[k] + T[i]) \lll s)$$

- a,b,c,d：缓冲区的四个字，它按照一定的次序随迭代步变化
- g：基本逻辑函数F/G/H/I之一
- $\lll s$ ：32位的变量循环左移s位
- $X[k] = M[q*16+k]$ ：消息第q个512位分组的第k个32位字
- $T[i]$ ：矩阵T中的第i个32位字
- +：模 $2^{32}$ 加法

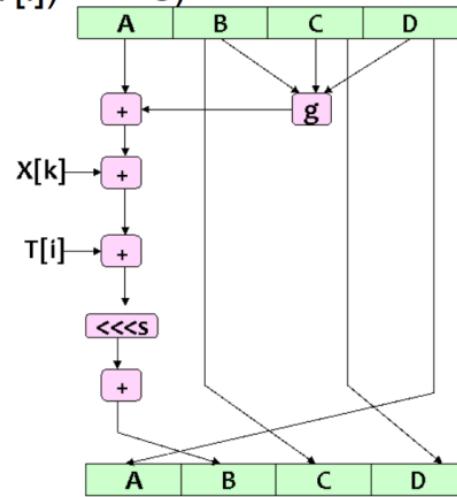
- $b \leftarrow a + ((a + g(b,c,d) + X[k] + T[i]) \ll s)$

- $X[k]$ : 消息第 $k$ 个512位

分组的第 $k$ 个32位字

- $T[i]$  : 矩阵 $T$ 中的第 $i$ 个32位字

循环	原始函数	$g(b,c,d)$
1	F	$(b \wedge c) \vee (\neg b \wedge d)$
2	G	$(b \wedge d) \vee (c \wedge \neg d)$
3	H	$b \text{ xor } c \text{ xor } d$
4	I	$c \text{ xor } (b \wedge \neg d)$



- 最后一轮+CV为下一CV  
s?

- 

**强度** 输入依赖小，强棚64弱128

## SHA

建立在MD4

512bit 分组

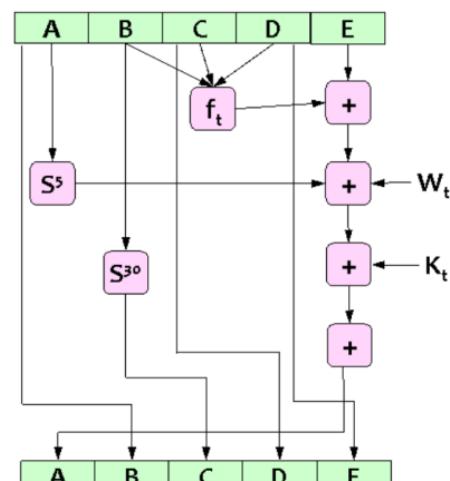
SHA1 message<264 --> 160md

sha1算法

- 补充到mod512=64
- 补充64位报文长度
- 初始化5个32寄存器缓存
- 分组处理，两组\*五轮\*16迭代

- 处理一个512位的分组要执行80步，每步的处理过程是一样的：

- $A, B, C, D, E$ : 缓冲区的5个字
- $t$ : 步骤编号,  $0 \leq t \leq 79$
- $f_t(B, C, D)$ : 第 $t$ 步使用的基本逻辑函数
- $S_k$ : 32位的变量循环左移 $k$ 位
- $W_t$ : 从当前512位输入分组导出的32位字
- $K_t$ : 加法常量。共使用了四个不同的加法常量。
- $+$ : 模 $2^{32}$ 加法



## RIPEMD-160

512分组输出160md

- 填充到mod 512 = 64
- 填充长度
- 初始化5\*32
- 2组\*五轮\*16步

## 比较

	<b>MD5</b>	<b>SHA-1</b>	<b>RIPEMD-160</b>
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	$\infty$	$2^{64}-1$ 位	$2^{64}-1$ 位
基本逻辑函数	4	4	5
使用的加法常量	64	4	9
低端位/高端位结构	低位在前	高位在前	低位在前

- 都不受弱碰撞，md5容易被强碰，另外俩在将来还安全
- 抗分析：MD5最差 RIPEMD最好
- 速度：都是32位加和位逻辑，MD5最快
- 只有sha1高位在前

## 数字签名DSS

防止通信双方自身攻击

- B伪造发自A的
- A否认发过的

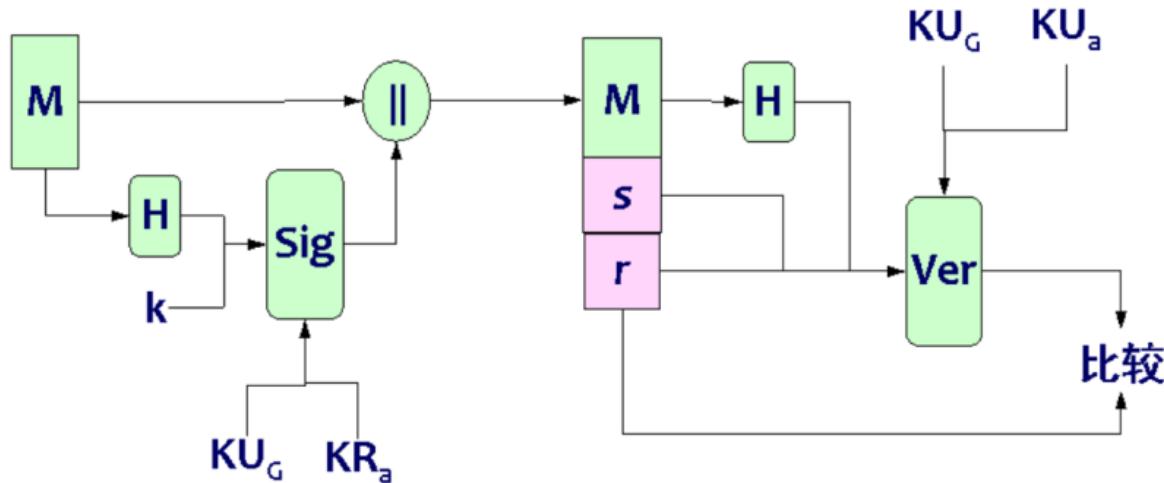
签名需要

- 验证 签名者、日期时间
- 认证被签名的消息内容
- 第三方可仲裁

分为：直接&仲裁

- 直接
  - 用自己私钥对消息orhash加密签名
  - 接收方公钥加密or对称密码得到保密性
  - 依赖发送方的私钥安全性
- 仲裁
  - X到Y的签名消息先给A
  - A检查消息内容和签名来源，然后加上时间戳发给Y，表示仲裁通过

数字签名标准DSS，使用SHA-1，定义DSA  
不能加密和分配，只签名



### ● 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 $k$ , $0 < k < q$

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：

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

如果 $v=r$ ，则确认签名正确

## 身份认证

确定你是你——信息（口令密码）、拥有东西（钥匙，盾）、身体特征（生物特征验证）  
安全网站：认证授权审计+修bug

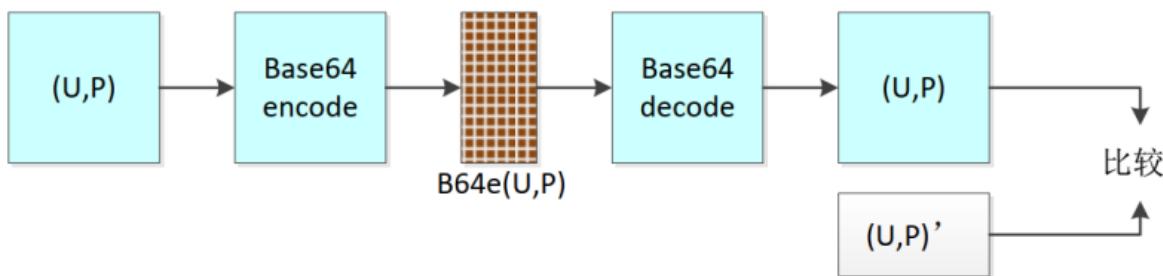
## 网站身份认证

HTTP: 一次连接，无状态

## Basic认证

每次发送携带凭证明文（账号+口令），与服务端用户凭证比较

Base64编解码



每次传递容易被监听窃取；

本地需要保存账号口令，安全隐患；

每个请求都需要验证，低效；

## 改进

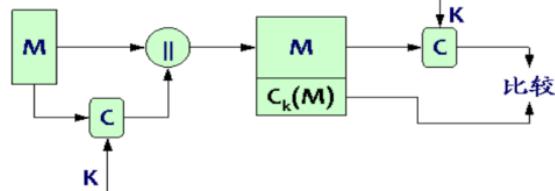
1. 加密传输，消息认证——监听窃取；表单验证+session? ——本地保存和每次检查

2. 对称加密U，口令就是密钥k，传输U和EkU——被重放攻击？

3. MAC认证

- 使用消息认证中的MAC认证技术，

- 客户端和服务端共享密码K（口令）



- 采用挑战/响应(Challenge/Response)机制，需要进行两次HTTP请求

- 第一次请求：服务器向客户端返回一个挑战码M

- 挑战码M由服务器随机生成，可以有效避免重放攻击

- 第二次请求：客户端使用MAC认证发送MAC码，服务器端进行验证

- C可用hash算法，如SHA1

第一次服务器只给你随机挑战码M

第二次用K加密后发回来，服务器加密认证正确

## 表单身份

针对：账号口令本地保存；每次都进行账号口令验证；

Session原理：客户端、服务器和时间段

- 服务器产生唯一的标识符发到客户端
- 客户端把此随机串存在Cookie中or本地文件，作用域和有效时间...
- Session存活时候，每次请求都有标识符，服务器关联起来
- 结束后销毁标识符

表单认证过程：

- 客户端发请求，服务端返回表单页面
- 用户填写表单（账户密码），服务端验证通过启动Session返回
- 客户后面请求携带Session的标识符，服务端验证。一般基于以标识符为key的哈希表

优点：简单；方便；易用

缺点：账号口令裸奔

改进：

- 引入挑战机制，避免明文传输和重放攻击
  - 挑战为字符串+时间戳t, 为U
  - 口令为密钥加密Ek(M || U)，发送U和Ek(M || U)，服务器用存储的k加密M || U来比较
- 传输账号口令时用传输层SSL来传输请求，在传输层加密并验证身份。启动SSL就是Https

不安全：——泄露，重放攻击

- 验证后cookie存放账号口令
- 账号口令加密后Ek (UP) 放Cookie  
可以加上时间戳t，每次服务端验证时间戳

## 增强认证

动态方式！

短信、动态（时间同步or计数器）、USBkey、数字证书

### 动态安全令

服务器和令牌段共享key和sn，对t进行划分取整。计算F(H(t || key || sn))，H为hash，F为压缩

### USB

私钥不可导出，在里面加密签名

## 数字证书

证书授权中心发行，用来数字签名的文件

包含：拥有者公钥，名称和中心的数字签名，对应私钥妥善存储。

# lect5 WLAN

---

## 概述

5W6A的梦想，TDMA，CDMA，GSM，3g，4g

最早的无线计算机通信：ALOHA

## 安全威胁

无线射频电波传输，无法物理隔离，只能广播



- 窃听：开放信道——泄露身份和位置——跟踪
- 假冒：截获身份信息后，假冒其入网
- 篡改：修改窃听信息并发送
- 重放，重路由：复制有效消息再发送or重用；改变路由来捕获
- 错误路由：路由到错误目的地
- 删除：截取删除
- 洪泛：发送大量无关、伪造消息耗尽资源

## 加密认证技术

### 无加密认证

AP 无线网络创建者，相当于无线路由器

STA 连接到无线网络的终端设备站点

SSID Service Set Identifier 便于用户识别的AP标志名，WIFI名。使用者提出正确SSID，AP就接受登入。SSID会被广播，禁用提高安全性

### 有线对等保密协议WEP

802.11b RC4流加密，访问控制+保护隐私。有挑战机制，四次握手

开放系统认证——默认认证方式，对请求认证的人提供明文认证

RC4：流密码，40/112bit的key，用CRC32循环冗余校验

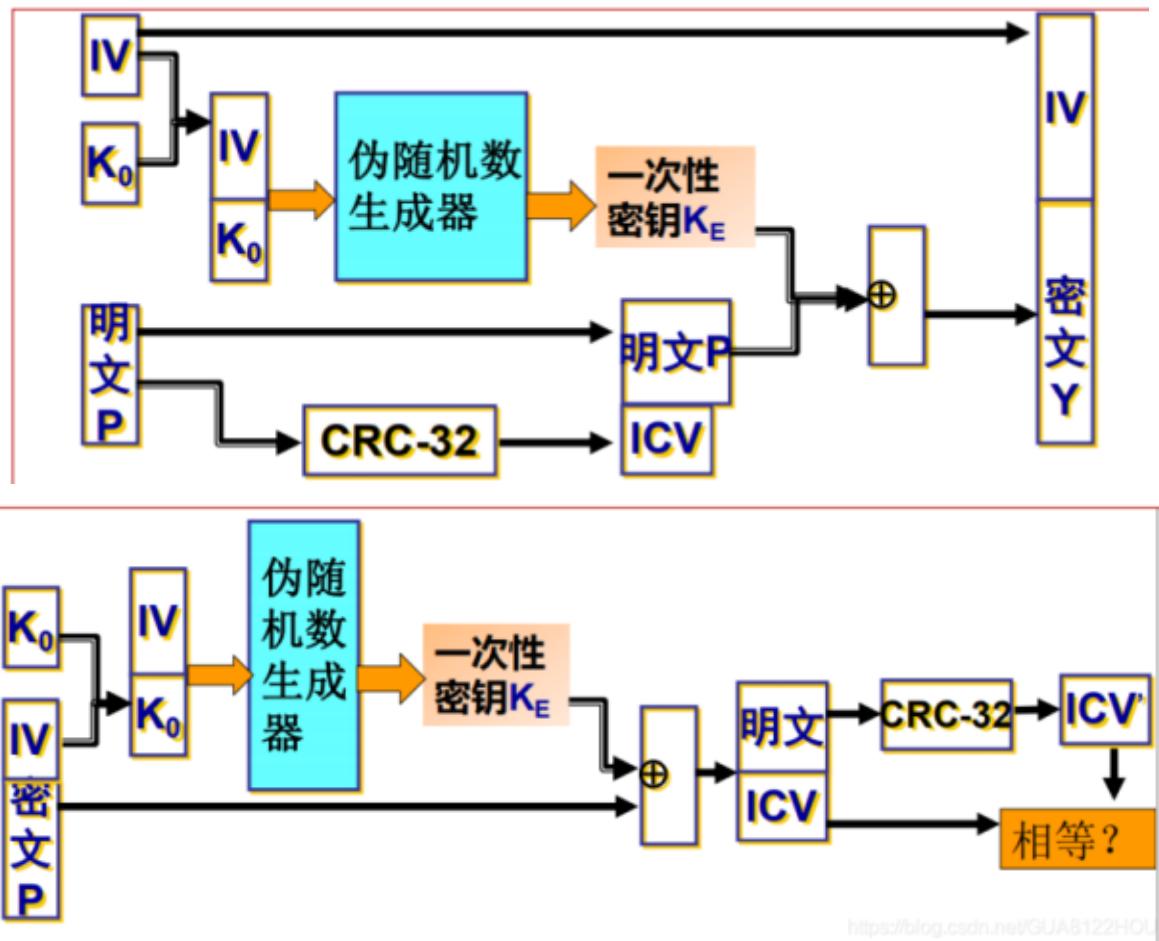
设备和接入点共享默认密钥，每个设备和其他设备要有密钥对关系，分发困难。

### WEP加密

- CRC32算校验和ICV，串接在明文P后。
- 24位初始向量IV和40位key连接得到64位数据，输入到虚拟随机数生成器产生一次性密钥KE
- KE和第一步异或得Y
- IV || Y传输

### WEP解密

- 24位初始向量IV和40位key连接得到64位数据，输入到虚拟随机数生成器产生一次性密钥KE
- 把密文和KE做异或得到明文P和校验和ICV
- 计算明文的CRC32，比较；接受或丢弃



安全性？

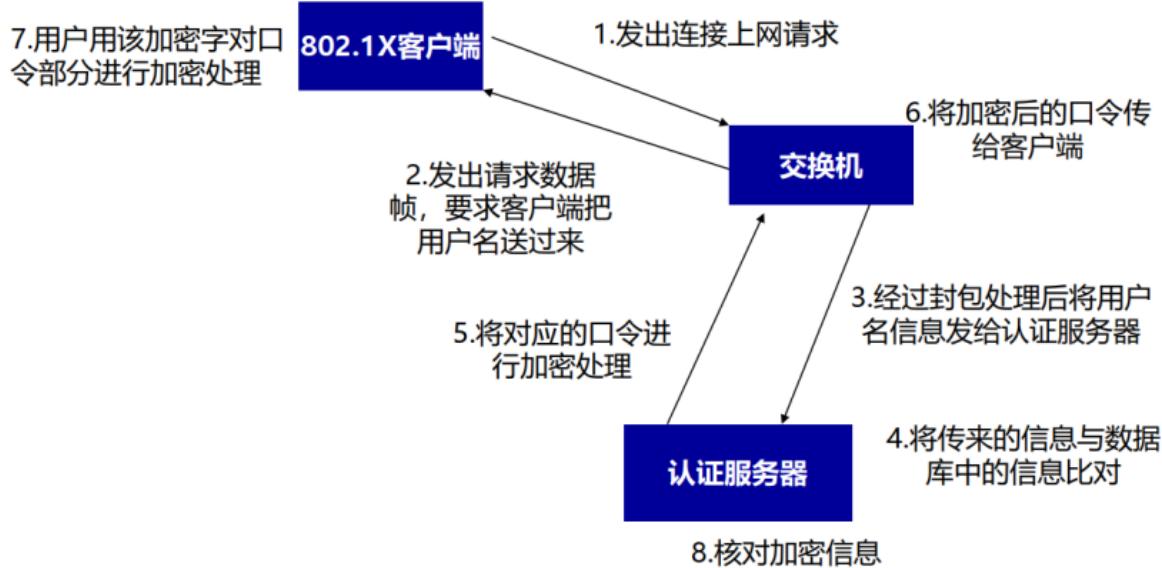
- SSID下所有STA和AP共享密钥，容易泄露
- RC4密码流的IV明文发送，且24位IV太短容易重复
- CRC线性非加密，不是安全的杂凑函数（hash）不能认证
- RC4位序列密码加密，太容易被攻破
- 不含序列号，没有帧顺序，重放攻击

## WPA1 过渡性，核心是1x和TKIP

TKIP认证，WEP包装

- 企业：802.1x认证
- 个人：Pre-shared key模式
- RC4流密码 128bitkey
- 动态变化每个数据包的密钥，混合生成不能轻易破译
- 每个包有独特的48位序列号防范重放
- Michael消息认证码MIC，包含帧计数抵抗重放

802.1x CS模式，在终端和AP建立连接之前验证用户身份，需要上层EAP配合认证和分发密钥



## WPA2

- 更安全的CCMP消息认证替代Michael
- 更安全的AES对称加密替代RC4
- 支持11g以上

加密技术	全称	加密算法	协议
WEP	Wired Equivalent Privacy (有线对等保密)	RC4	IEEE 802.11b
WPA	Wi-Fi Protected Access (WiFi安全存取)	RC4, 使用TKIP	IEEE 802.11i draft 3
WPA2	Wi-Fi Protected Access 2 (WiFi安全存取 第二版)	支持AES, 使用CCMP 需要新硬件支持	IEEE 802.11i

- WPA使公共场所和学术环境安全地部署无线网络成为可能
  - WEP使用一个静态的密钥来加密所有的通信，这意味着为了更新密钥，技术人员必须亲自访问每台机器，而这在学术环境和公共场所是不可能的
  - WPA采用有效的密钥分发机制，不断转换密钥

## lect6 CIA, VPN

### 网安体系结构



Confidentiality 保密、机密

信息不泄露；数据拓扑流量都需要保密；防止被动攻击；

Integrity 完整性

不被篡改or可以检测篡改、插入、重放；防止主动攻击；

Availability 可用性

授权用户得到应得资源服务，防止拒绝服务攻击；

对信息系统可用性的攻击；

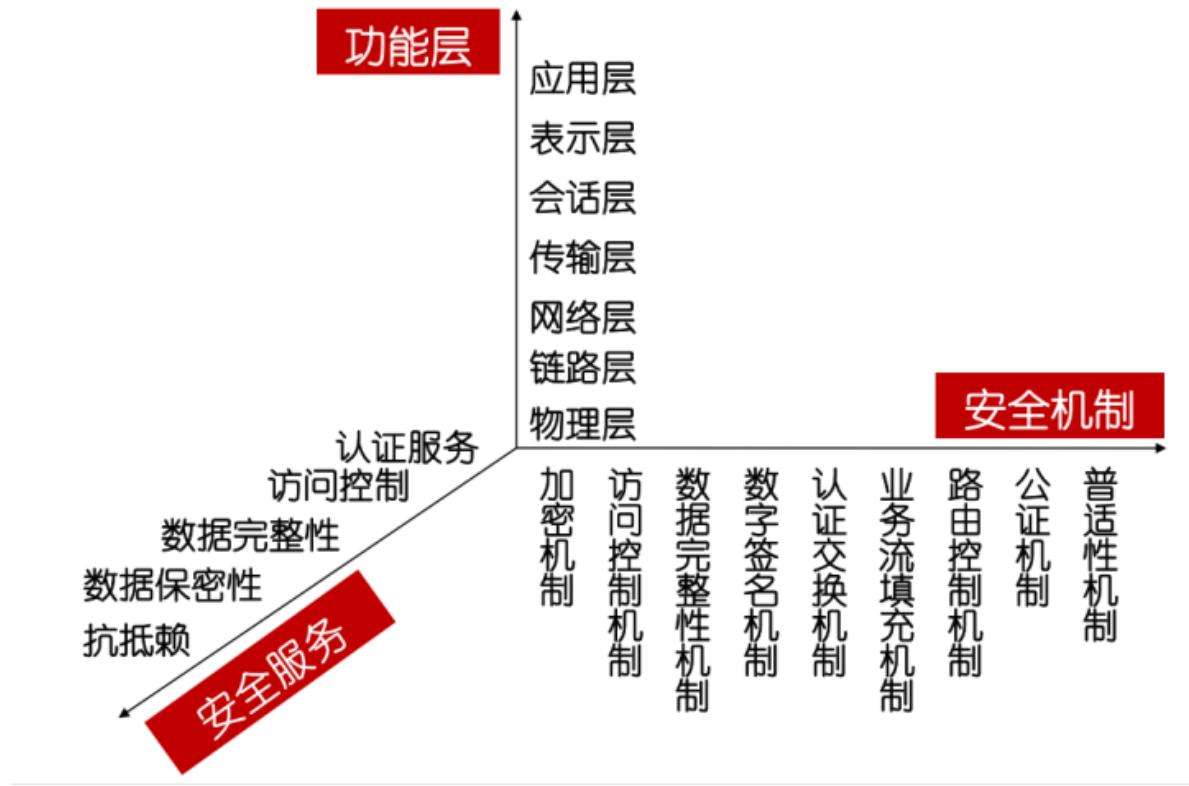
对路由设备处理能力，buf和链路带宽攻击

## X.800：OSI安全框架，定义系统方法提供给网络管理员

安全服务：由系统提供的对系统资源特殊处理或通信，通过安全机制实现安全策略

安全机制：免收监听；组织攻击；恢复系统

安全攻击：主被动



## 服务

服务	机说明
认证	单条：保证发送方真实；通信：保证双方真实+不受干扰和伪装
对等实体认证	保证实体身份；禁止伪装和非授权重放；面向连接
数据源认证	保证来源是来源；保证来源身份合法；不保护数据复制修改；面向无连接
保密	连接or无连接；力度：流、消息、选择字段 防止流量分析 (src dst freq len等特征)
完整性保护	连接：收发一致；无连接：不被篡改； <b>主动攻击检测而非阻止攻击？</b>
访问控制	合法才可访问
抗抵赖	接收方证明消息只能由发送方发出；发送方证明消息确实被接收到
可用性	根据系统性能说明，按照授权系统实体要求使用系统和其资源

## 机制

普通&特定

普通：不属于任何协议层or安全服务的机制

- 可信功能(trusted functions)
  - 根据某些标准被认为是正确的
- 安全标签(security Labels)
  - 资源的标志，指明该资源的安全属性
- 事件检测 (Event Detection)
  - 检测与安全相关的事件
- 审计跟踪 (security audit Trail)
  - 收集用于安全审计的数据，对系统记录和行为的独立回顾和检查
- 安全恢复 (security recovery)
  - 处理来自安全机制的请求，如事件处理、管理功能和采取恢复行为

特定：特定协议层实现的

- ① 加密机制
- ② 数字签名机制
- ③ 访问控制机制
- ④ 数据完整性机制
- ⑤ 认证机制
- ⑥ 业务流填充机制
- ⑦ 路由控制机制
- ⑧ 公证机制

## VPN Virtual Private Network 虚拟专用网

BG TCPIP安全缺陷

- 容易监听，篡改，重放，修改IP
- 源发可以指定中间路由——源路由攻击
- 序列号猜测，缓冲区溢出

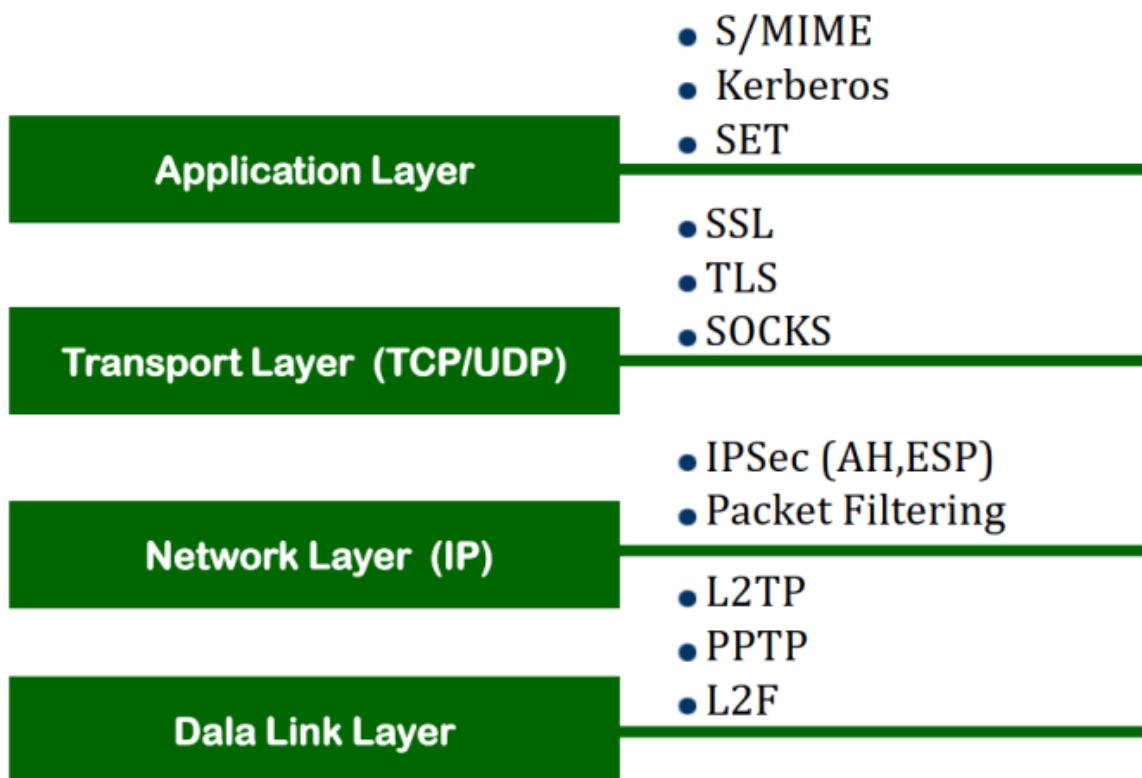
概念：逻辑等价于一条物理的专用长途数据线路，定制化。在公共网络上仿真一条点对点的私有链接。

安全功能

- 数据机密性
- 数据完整性
- 数据源身份认证
- 重放保护

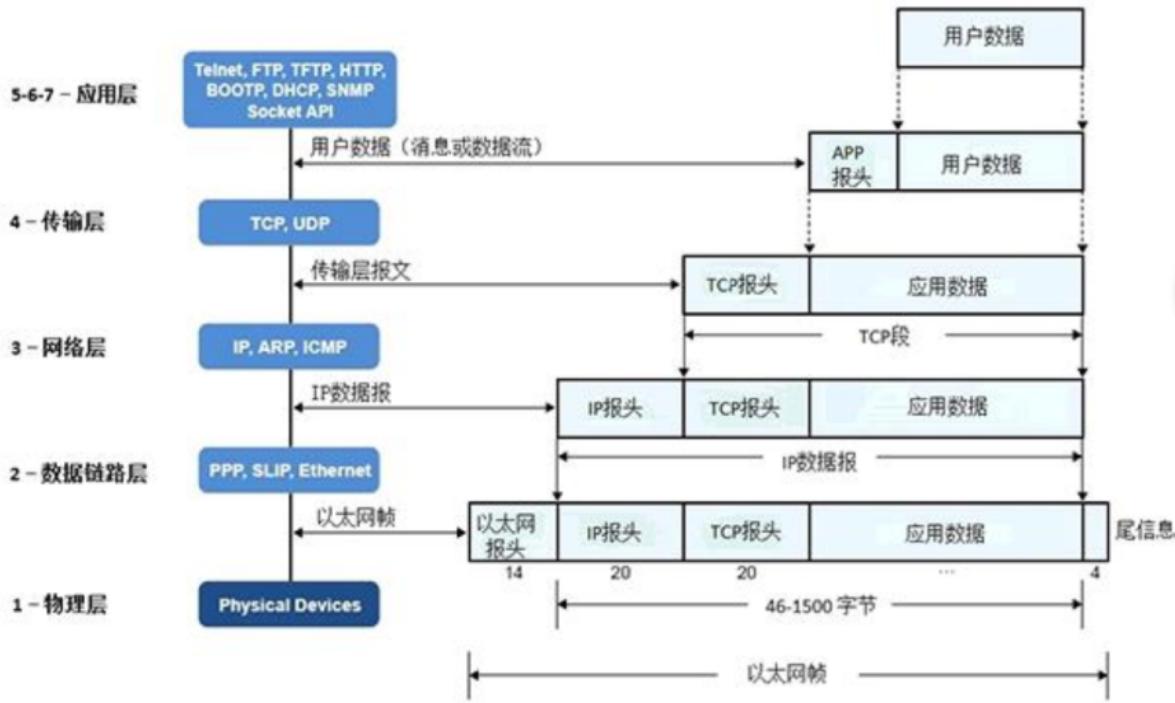
解决方案（安全协议）

- 链路层：L2TP PPTP L2F 很少使用
  - 认证终端实体而非流入报文，无法抵抗插入和地址欺骗
  - 没有完整性校验，可能拒绝服务攻击
  - PPP报文加密但是不支持自动产生、刷新密钥，会被破译
- 网络层：IPsec IKE
- 传输层：SSL
  - 零客户端，任何B/S结构 (browser和Server)
  - SSL和IPsec结合，网关也常被集成
  - SSL低成本，但是视频会议非BS无法用ssl vpn
- 应用层（无法VPN）
  - https : http+ssl
  - S/MIME 安全邮件
  - SET 安全交易



## lect7 IPsec和IKE

网络层：IPsec安全，IKE管理密钥



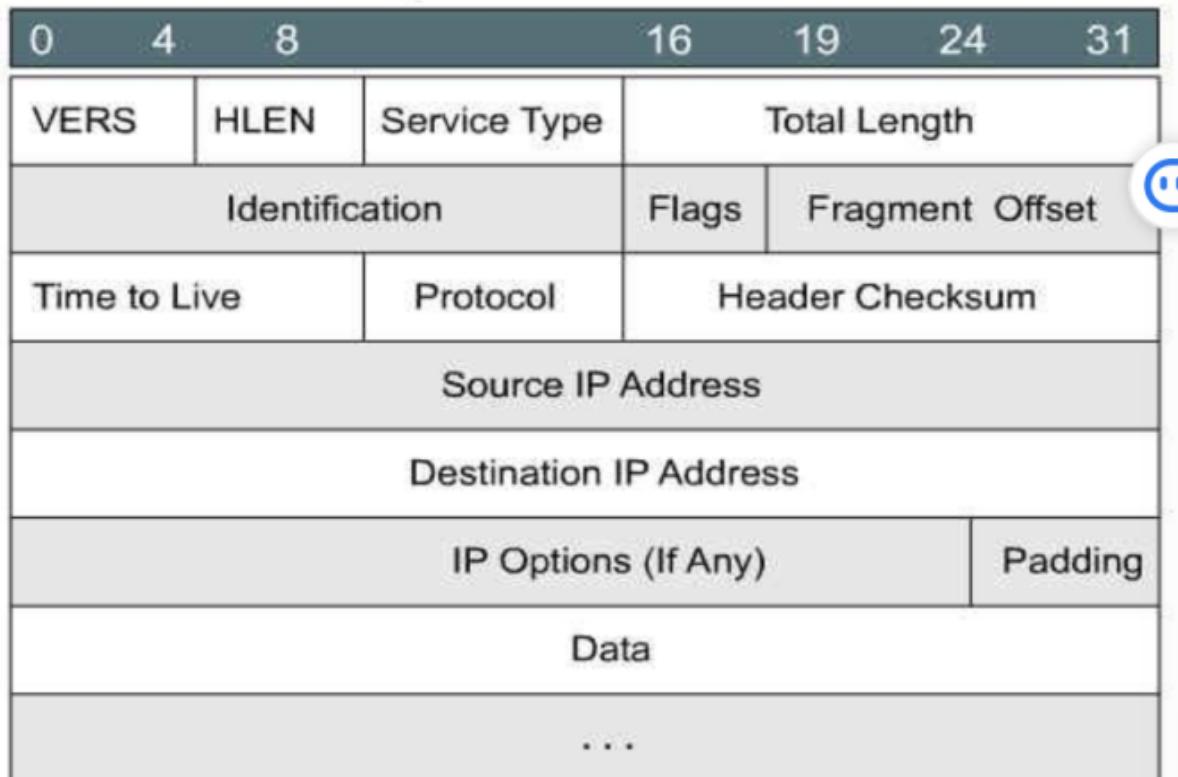
IPsec保证IP级别的安全性

- 认证：确保从源发出且未被篡改
- 保密：加密防止窃听
- 管理密钥

原来IP：无连接，无顺序（重复、丢失）设备简单无状态

不认证、不完整、不保密，通过IP地址有一些访问控制

威胁：窃听、伪造地址、篡改、重发



实现：

- 主机实现（OS）：端到端安全；针对每个会话提供安全保障；对应用和用户透明

- 防火墙上实现：不改OS，为所有应用提供安全服务。在网络层以下
- 路由器：

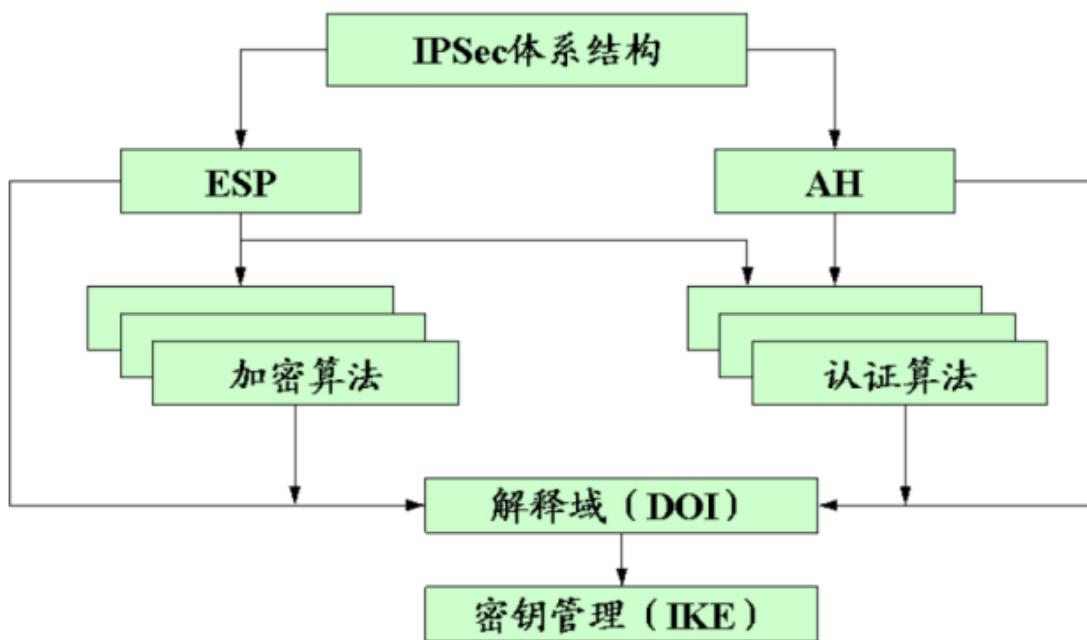
## IPSec体系结构

### 文档

认证和加密——通过主IP包头使用扩展包头实现安全特性。

认证扩展头：认证头AH

加密头扩展：封装安全载荷ESP(也认证也加密)



### 服务

为IP层提供安全服务，系统选择协议和算法，并提供密钥

	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		✓	✓

## 安全关联SA Security Association

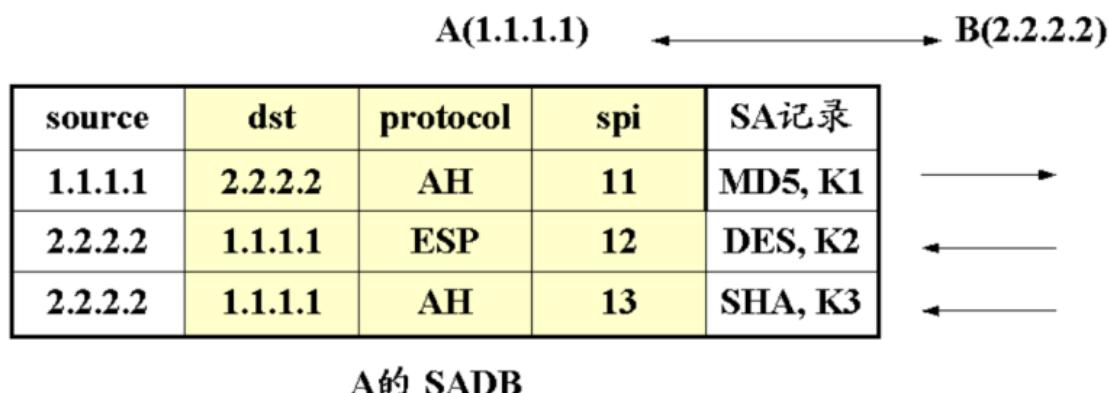
通信双方对要素的协商，一些安全信息参数集合比如  
协议；操作模式；密码算法；认证算法；密钥及有效期；  
是单向关系 发到收的，如果交换需要建立两个SA。

安全服务可由AH或ESP提供但不能二者都提供？

SA由三个参数唯一确定：

- 安全参数索引SPI parameters index
  - 和SA相关仅在本地有意义
  - AH和ESP携带，接收方选择合适的SA来处理
- IP目的地址IPDA
  - 单一地址表示SA目的地址
  - 可以是用户端、防火墙or路由
- 安全协议标识符：是AH or ESP的SA

任何IPsec实现端，都得有SADB来定义每个SA相关的参数。用上述三元组来索引。



### SA参数列表

- 序列号计数器 (必须实现)
  - 一个32位整数，刚开始通常为0，用于生成AH或ESP头中的序列号域
  - 每次用SA保护一个包时增1；在溢出之后，SA会重新进行协商
- 序列号溢出标志 (必须实现)
  - 表明序列号计数器是否溢出，
  - 序列号计数器的溢出时，该值为1时，产生审查事件并阻止该SA继续下发数据包
- 反重放窗口 (必须实现)
  - 用于决定输入AH或ESP报文是否是重放的32位计数器
- AH信息组 (AH必须实现)
  - 认证算法，密钥，密钥生存期和AH的相关参数

- **ESP信息组** (ESP必须实现)
  - 加密和认证算法，密钥，初始值，密钥生存期和ESP的相关参数
- **SA的生存期**
  - 一个时间间隔或字节计数
  - 生存期结束时，SA必须终止，或用一个新的SA替换
- **IPSec协议模式** (必须实现)
  - 隧道模式或者传输模式
- **Path MTU** (必须实现)
  - 任何遵从的最大传送单位路径和迟滞变量

收发IP包时的操作：

- 在**SPDB**中通过比较对应域的值 (dstsrc的地址和端口 用户标识 数据敏感级别 传输层协议 IPsec协议 IPv6报类、流标签、服务类型) 寻找匹配的entry, 0或多个。
- 根据策略来操作：丢弃； bypass IPsec； apply IPsec等
- 如果存在SA则选定SA和对应的SPI， AH或ESP的选择。。
- 收到数据包，解析出三元素SPI， DSTaddr和AH | ESP
- 查找本地的**SADB**
  - 找到SA条目，把参数和包头中的域比较。
  - 没有查到，输入包丢弃，输出包创建并存入SADB

## SA的两个库

存在OS kernel

通信前必须建立SA，先查SPDB找信息流和联系的SA。有的丢弃有的不加密有的加密；再关联或找SADB选择参数。

多种方式实现IP通信的sec服务；

对于需不需要保护的流量**大粒度**区分；

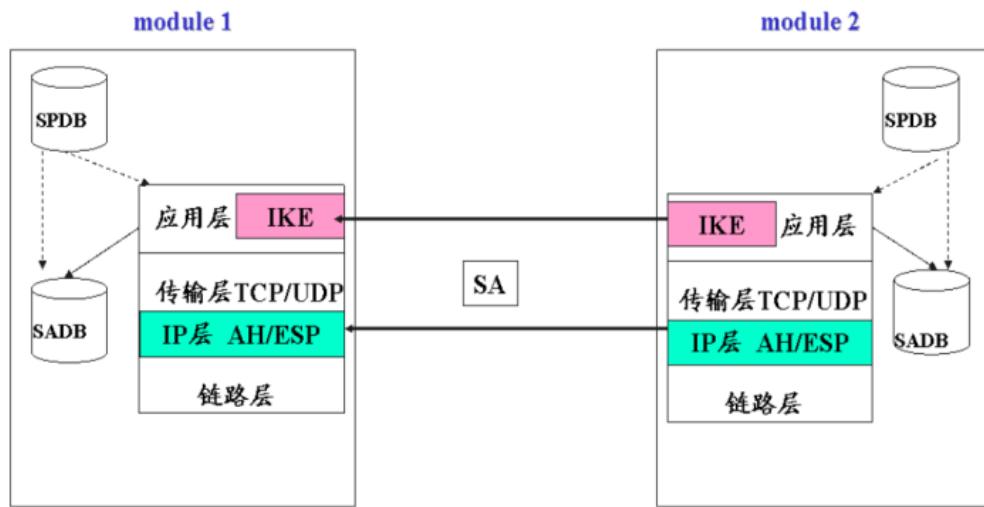
IP流量和SA相关是通过安全策略数据库**SPDB**定义的

- 定义IP流量子集的entry
- 指向流量对应SA的指针
- 多个entry和同一个SA相连
- 多个SA和同一个SPDB entry 相连

每一条SPDB entry由IP集合和上层协议定义，成为选择子SA selector

- 过滤输出流量，映射到某个SA

- 安全关联数据库(SADB): 定义SA
- 安全策略数据库(SPBD): 使用SA



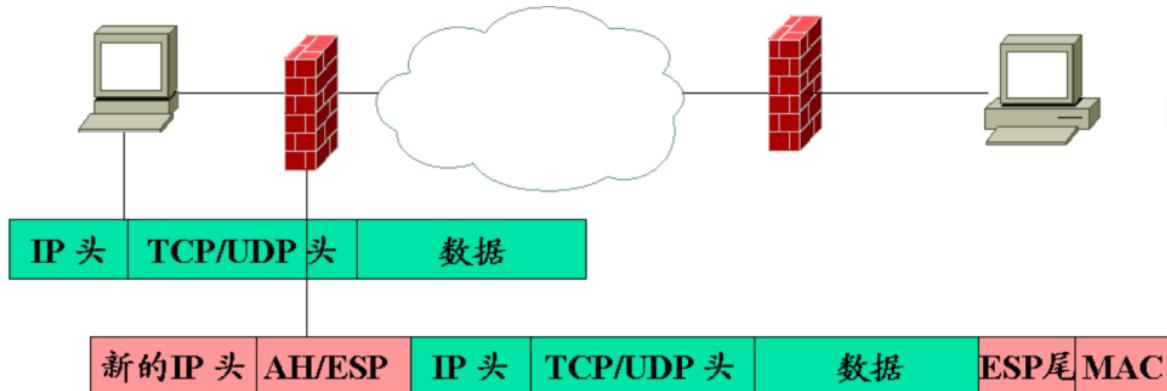
## IPsec 传输模式

提供上层协议保护，增加IP包载荷保护。用于两台主机的端到端通信，放到IP头后面和TCP/UDP头前面

- ESP加密和认证IP载荷，不包括报头
- AH认证IP载荷和部分报头

## IPsec 隧道模式

原来的包被看成数据直接抱起来成新的大IP包



保护整个IP包，整个数据包被当做新的IP载荷拥有新IP报头，利用隧道传播，中途路由器不能查内部IP报头

- ESP加密和认证整个内部IP包
- AH认证整个内部IP包和外部IP报头的部分

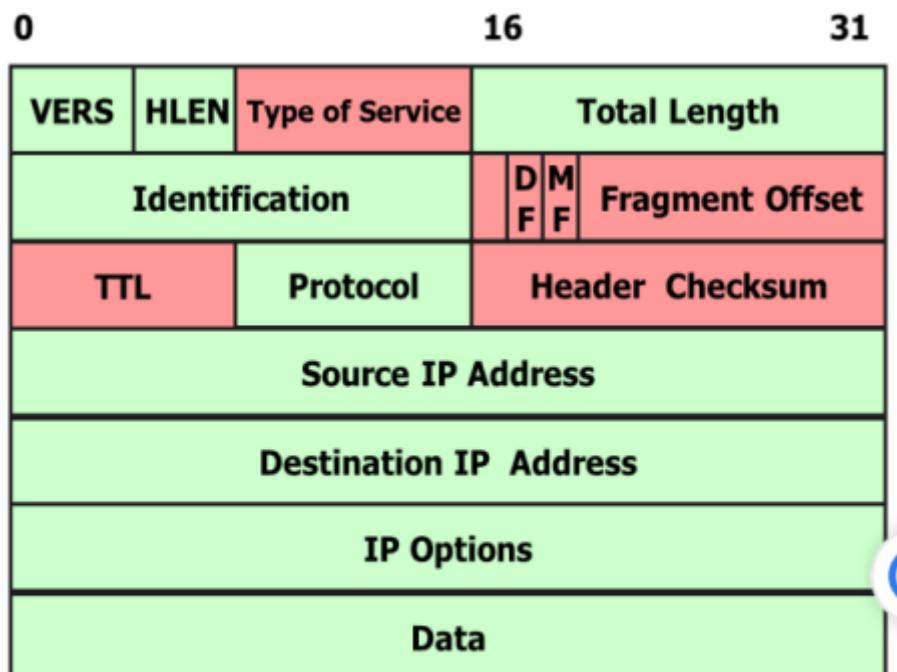
## 认证头AH

基于消息认证码MAC，双方共享公钥

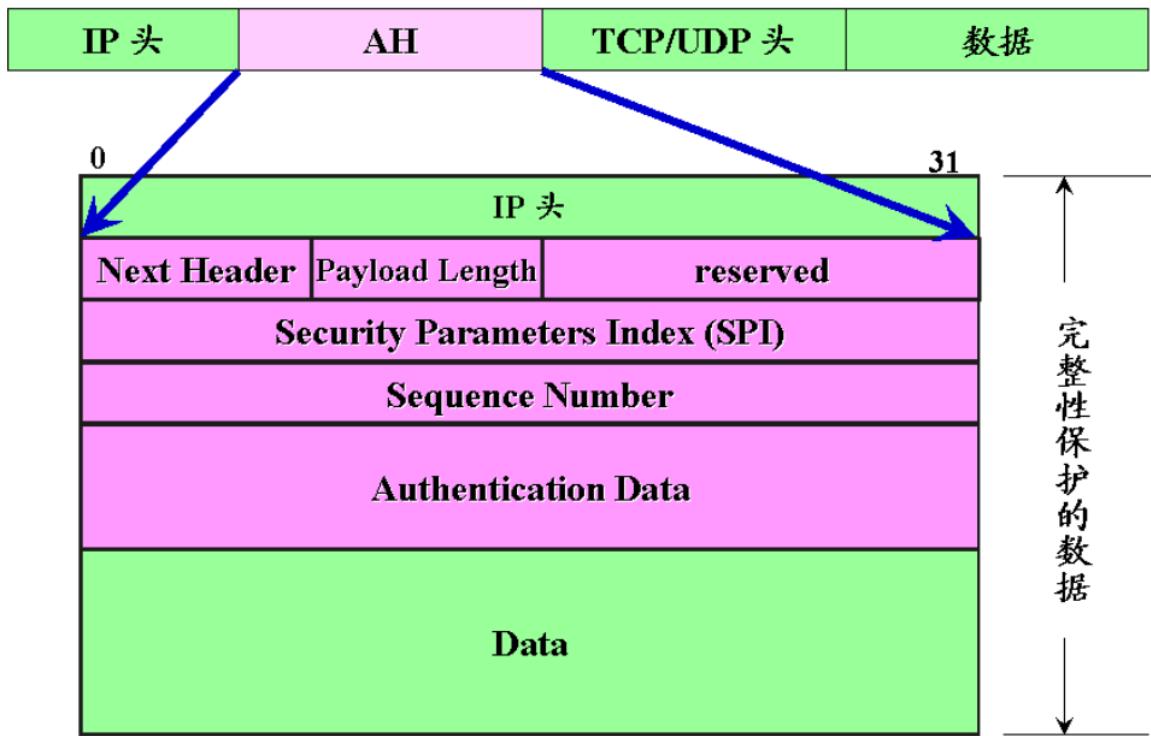
- SN不能允许循环计数，到头就中止SA再次协商，可以反重放。要求按照这个序号建立窗口W (64)
- 窗口的右边界代表最大的序号N，记录目前收到的合法包的最大序列号，任何序列号在N-W+1到N之间的包均可以被正确接收，并标记窗口的正确位置
- 包到达之后：
  - 如果接收包在窗口中且是新包，则验证MAC；验证通过，则在窗口中标记位置
  - 如果接收的包超过窗口右边界而且是新包，则验证MAC；验证通过，则以这个序列号为窗口的右边界并在窗口中标记相应的位置
  - 如果接收包超过窗口的左边界或未通过验证，则丢弃此包，并生成审核事件

AD变长域，是4B整数倍。包含完整性校验值ICV或者包的MAC，可以用HMAC-MD5 | SHA1-96 取前96位

IP头的不变部分，AH中终点可预测部分参与计算MAC。其他全0



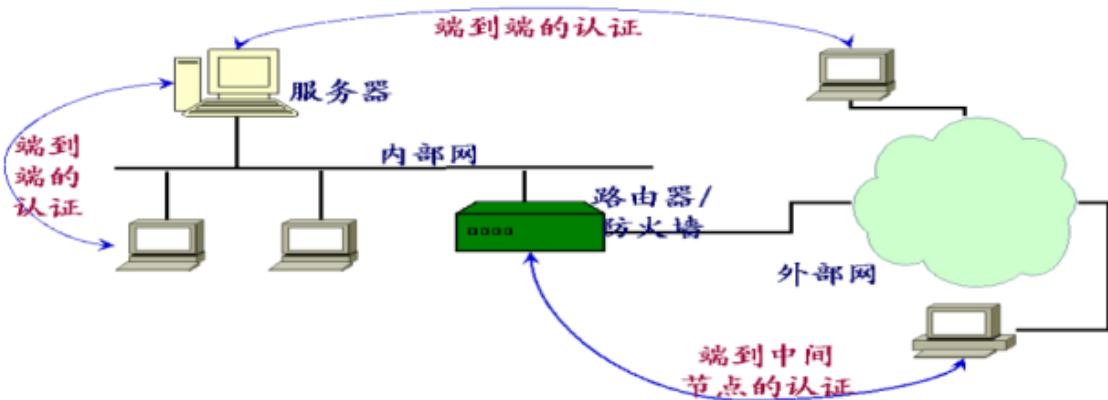
IPv4 Header 中的可变部分



途径1 直接在服务器和客户站之间使用传输模式认证。只要共享同一个key就好，这个key用在哪了？

好像在MAC中也需要独特的公钥Key

途径2 服务器不支持认证，工作站需要向防火墙证明自己身份并访问整个内部网络，用隧道模式SA



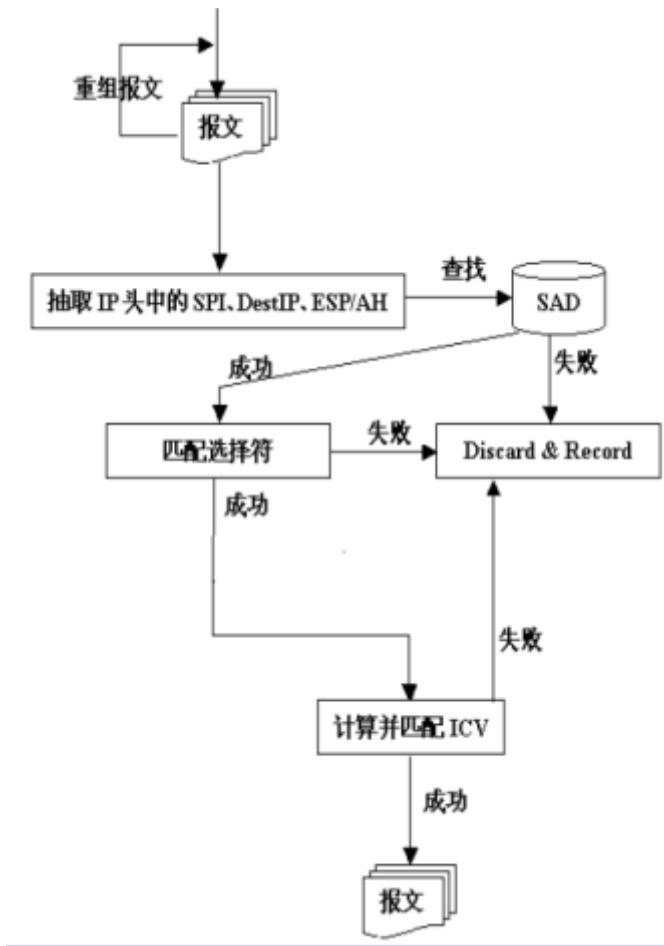
隧道模式AH 通常在防火墙|路由器上实现

前面包裹上新的IP头和AH，没有尾巴

### 对接受包的处理

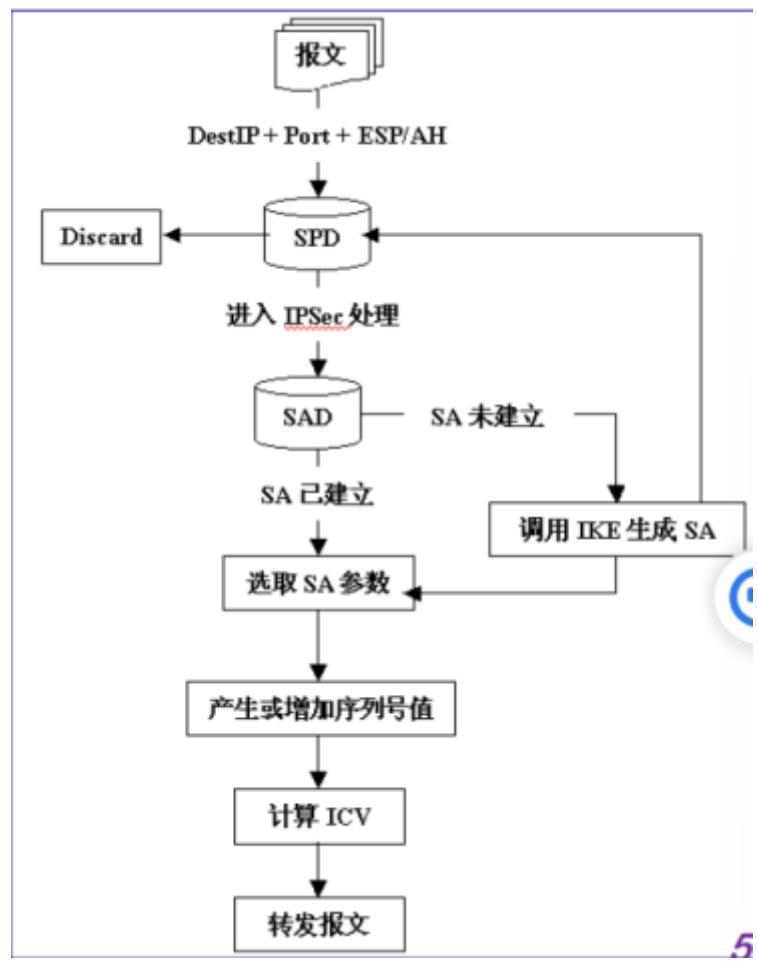
- 解析SA的三元组，在SADB里**匹配SA参数**，不一致就扔了，找不到也扔了
- (opt) 滑动窗口的检查序列号
- 计算**ICV比较**，不相等扔了，相等交由IP协议栈处理，继续路由...

随时不相符扔了+记录审计



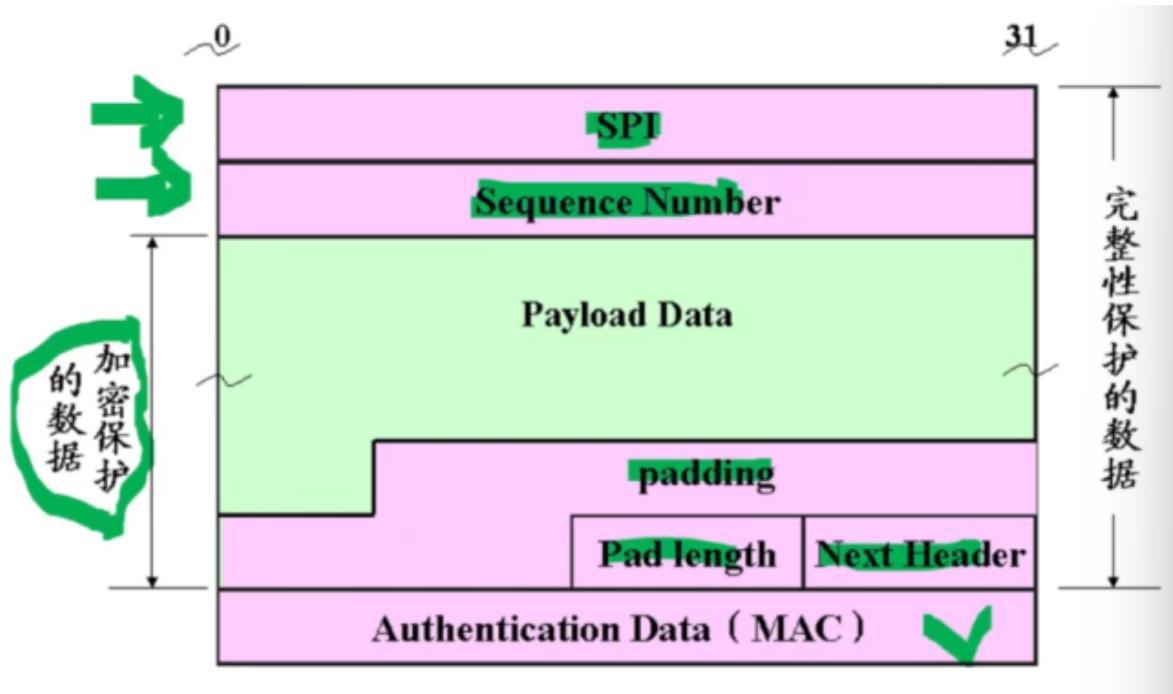
## 对出去outbound的包处理

- 从IP协议栈中收到需要转发的包
- 使用选择子查找SPDB，获取安全策略，也可能直接扔了
- 需要IPsec，则查找SADB
  - 找到了，选取参数计算ICV
  - 没找到，使用IKE协商建立SA存上，并选取参数计算ICV转发
- 不需要直接发走



5

## 加密和认证头ESP



加密算法: 3DES RC5 IDEA CAST Blowfish 对称加密

padding: 填充到明文符合加密算法的要求, 对齐后面的两个填充长度和nexthead到右对齐。隐藏载荷实际长度, 提供流量保护

## 传输模式

IP ESP头 密文 秘密ESP尾巴 可选MAC尾巴 (opt)

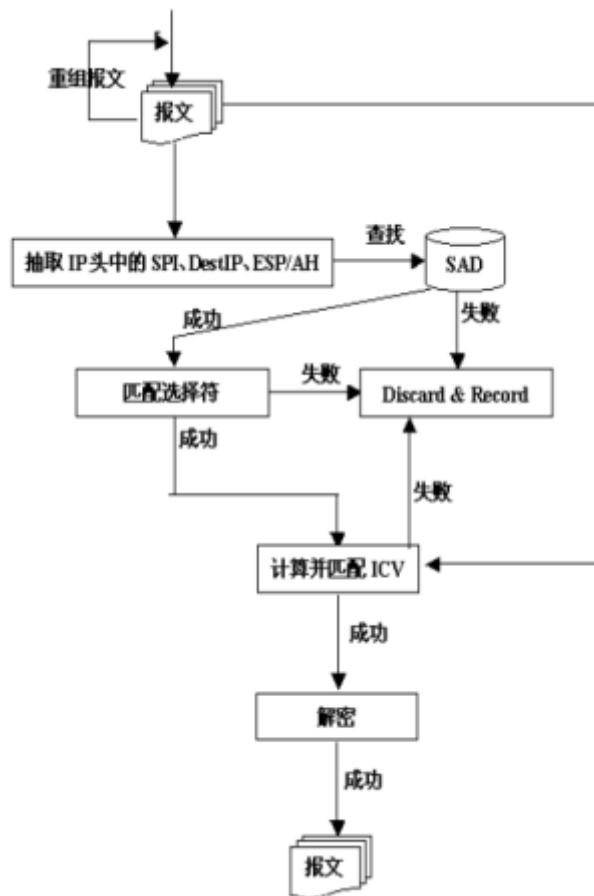
中间路由只检查前面的IP头

目的节点处理三元组，查SADB找到SA的参数，来恢复数据

## 隧道模式 保护原来的IP头吗

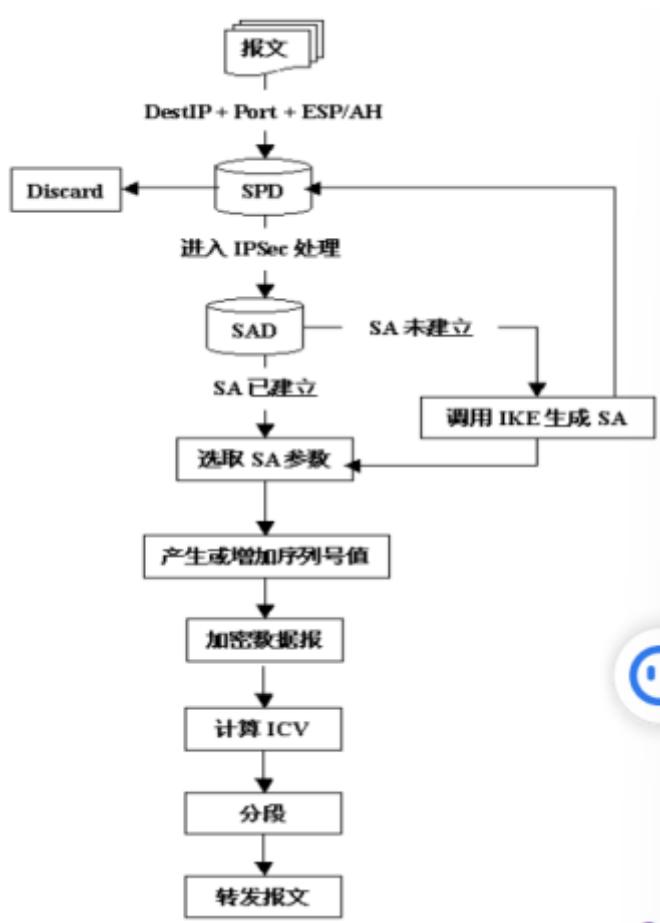
新的IP头 ESP头 加密的 (IP头 数据 ESP尾巴) 可选MAC尾巴

### 对Inbound 最后多一个解密



一开始也要查SPDB

### 对outbound 中间多一个加密——先加密再ICV



- 查SPDB
- 查SADB——用IKE协商建立储存or查到
- 产生计算序列号值
- 加密
- 计算ICV
- IP——分段转发 (根据MTU分片)

路由在中间 安全在两边

## 安全关联组合

单SA实现二者之一，需求：主机间IPSec在安全网关最相同流量分离。即在防火墙拆一层然后在主机可能再拆一层。。

有个SA序列组合成束，每层SPI|SA都不同

- 传输邻接：IP包多个协议，只一级。即AHESP
- 隧道迭代：可以一直包装。。。多层次嵌套

甚至可以先用传输，然后再隧道包装

## IKE管理密钥Internet Key Exchange

- 管理员手工给系统分配各类密钥
- 自动，协商。。。

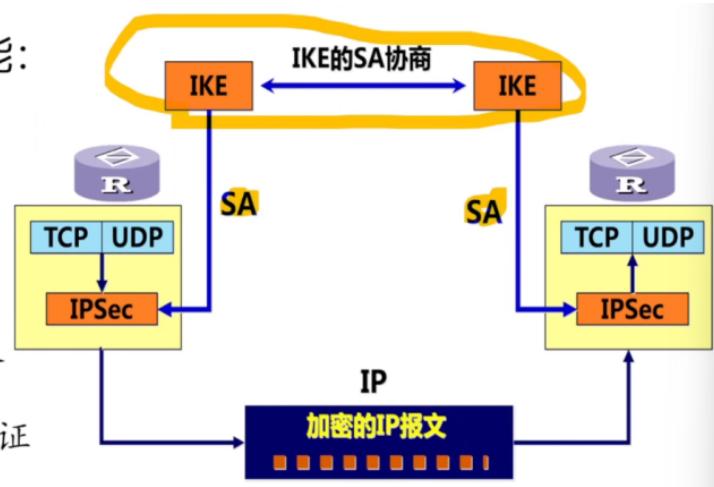
IKE：自动协商交换密钥 + 建立安全关联SA（维护SADB）

服务很多SNMPv3 RIPv2 OSPFv2

精髓：不安全网络上不能直接传，都是算出共享密钥。（核心为DH交换）

IKE为IPsec提供了如下功能：

- 降低手工配置复杂度
- 安全关联SA定时更新
- 密钥定时更新
- 允许IPSec提供反重放服务
- 允许在端与端之间动态认证

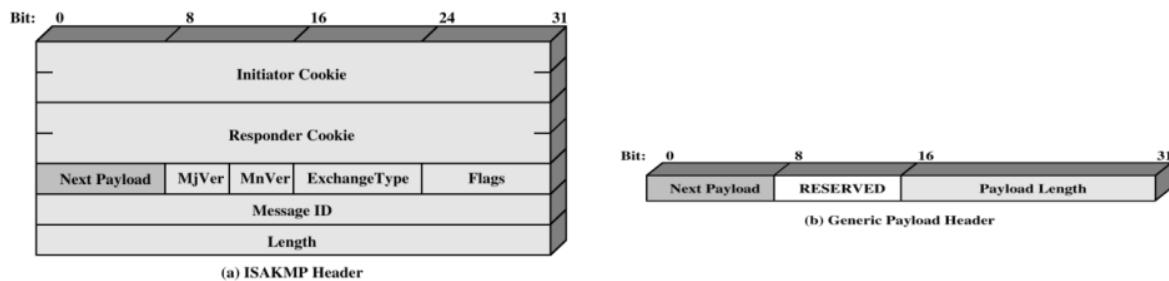


混合了三种协议

## 报文格式

from ISAKMP，可以在任何传输orIP层实现，用UDP port500

- ISAKMP双方交换的信息以“报文头+载荷”的形式传输，每个ISAKMP报文由一个定长的报文头和不定数量的载荷组成



载荷里面是协商的参数，需要协商的参数都是payload。

- 下一个payload的类型
- 两个cookie用来唯一标识密钥交换会话
  - 使用地址、随机数、日期时间等等MD5出来
- FFlag只用到了低三位
  - 载荷加密
  - 同步密钥交换，如果1则需要A最后给B发建立成功
  - 载荷有认证or没加密只认
- 报文总长度（头+载荷）

13种载荷：

1. SA，协商安全属性，指出解释DOI和状态
2. proposal 包含在1
3. TTransform 包含在2
4. Key Exchange
5. ...

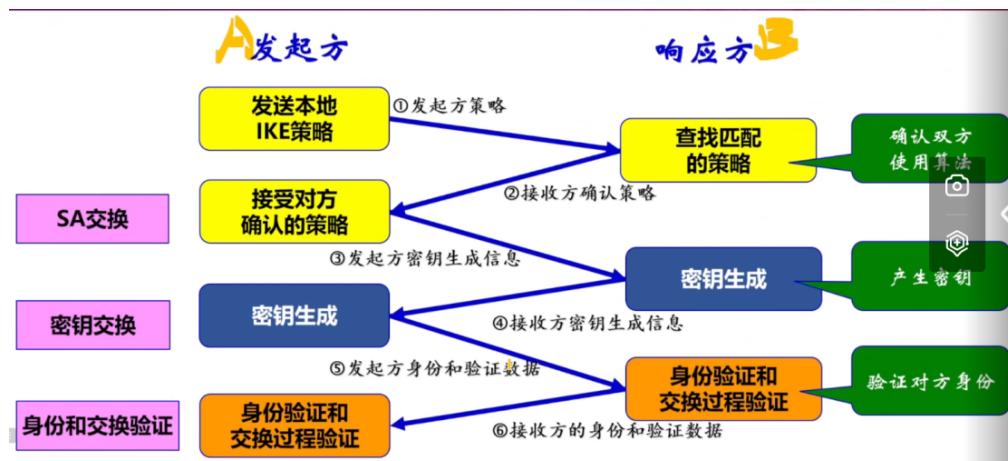
# IKE体系结构——两个阶段

- 一阶段：两个IKE实体间建立IKE SA，创建通信信道并验证。提供机密，消息完整和消息源验证服务

需要协商：加密、哈希、认证、DH信息等等

- main主：6消息，有身份保护

- 12：SA协商策略，商量SA之类的协议信息。大部分是使用的算法和限制等。1载荷很长2剩下3个回去
- 34：密钥生成信息，带着key exchange和nonce载荷，用DH
- 56：加密的身份和验证数据，ident身份认证（标识信息主机名等等）和消息认证Hash（hash三组密钥信息）
- 56的散列载荷hash相同，一阶段成功
- 



- 积极（快速）：3消息，无需身份，一方地址动态。可以用name验证而无需IP，但是两个都变就不行

- 1：发起策略+密钥生成，SA KEY Nonce ID都给过去
- 2：密钥生成+身份和验证数据，响应1并加入HASH
- 3：身份验证数据，加入HASH。看hash成功就直接完成
- 



- 二阶段：使用IKESA建立IPsecSA（各种参数）】

- 快速：3消息。可以协商多个SA，双向双方通信需要8个SA，两边各四个（出入\*AHorESP）  
协商加密和哈希的算法，验证方法，DHkey，周期和密钥长度等

- 1：Hash SA ProposalTrans KeyEx Nonce ID...
- 2：Hash SA ProposalTrans KeyEx Nonce ID...
- 3：Hash
- 很像上面的3步快速

## IKE的三个工作模式

### 传输

端到端节点，均实现IPsec，那么传输模式

### 隧道

网关间用隧道，端不需要IPsec

### 嵌套

端到网关：隧道套着传输模式

## IKE工作过程

守护进程在后台，被内核或对方IKE唤醒

内核可以指示IKE删除某个SA，此时IKE也会通知对面IKE删除or忽略

### 总结

IKE 层次相当高，位于应用层，用udp500走

不足：

- 太复杂
- 只用于Ipsec的建立SA
- 往返太多，消耗资源
- 容易被攻击：拒绝、中间人、重放

## lect8 SSL

---

相比Ipsec好处，Ipsec在网络层对所有传输都会使用。但是SSL可以选择，有差别的为应用层提供服务

### 介绍

SSL在TCP之上，保护任何TCP上的应用。

IPsec无法处理同端系统中不同应用的安全需求，为两个应用之间提供保密和安全。

SSH强制认证+数据加密也在此，还有SP4和TLSP

SSL 安全套阶层安全机制，保护基于WEB

用于CS的身份认证（证书与第三方），消息认证和数据保密

在应用层协议传输之前，SSL协议已经完成了客户端和服务器的身份认证、加密算法和密钥的协商；在此之后，双方建立起了一条安全可信的通信信道，应用层协议所传送的所有数据都会被加密，从而保证了安全

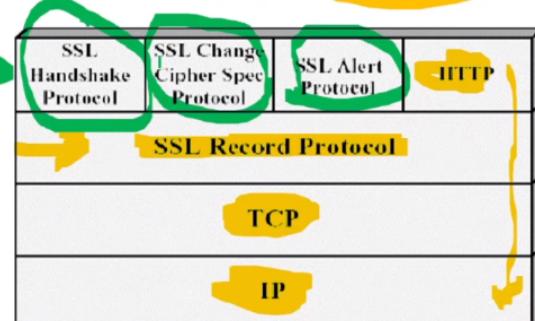
## 使用案例

接受时候相反

- 用户：浏览器里输入 **https://www.sslserver.com**
- HTTP层：将用户需求翻译成**HTTP请求**
- SSL层：借助下层协议的**信道安全的协商出一份加密密钥**，并用此密钥来**加密HTTP请求**
- TCP层：与web server的**443端口**建立连接，传递**SSL处理后的数据**；接收端与此过程相反
- SSL在TCP之上建立了一个**加密通道**，通过这一层的数据经过了加密，因此达到**保密的效果**

## SSL体系结构

- 两个实体：**客户机+服务器**
  - 基于证书，SSL在客户机和服务器之间完成双方**身份认证**
- 两个概念：**会话+连接**
  - 会话是虚拟连接，**连接是特定通道**，一个会话协商可以由多个**连接共享**
- 两层协议：**握手协议+记录协议**
  - 握手协议：**认证+协商**
    - 算法、密钥、secrets、初始向量等
  - 记录协议服务：**数据封装+安全通信**



会话Session，通过握手关联CS，虚拟的连接关系。握手商量xxx

连接Connection：特定信道映射到一个TCP连接，共享一个会话中协商好的信息。连接参数：

服务器和客户机的随机数	<b>Server and client random</b>	每个连接中，服务器和客户机选择的字节序列
服务器写 MAC 密码	<b>Server write MAC secret</b>	用于对服务器发送数据进行 MAC 操作的密钥
客户机写 MAC 密码	<b>Client write MAC secret</b>	用于对客户机发送数据进行 MAC 操作的密钥
服务器写密钥	<b>Server write key</b>	用于服务器对数据加密和客户对数据解密的对称加密密钥
客户机写密钥	<b>Client write key</b>	用于客户对数据加密和服务器对数据解密的对称加密密钥
初始化向量	<b>Initialization vectors</b>	当使用 CBC 模式的分组密文时，为每个密钥维护的初始化向量。该字段首先被 SSL 握手协议初始化，然后每个记录最终的密文块被保留下作为下一个记录的 IV
序列号	<b>Sequence number</b>	每一方为每个连接的传输和接收报文维持着单独的序号。当一方发送和接收修改密文规约报文时，相应的序号被设置为 0。最大 64 比特。

## SSL记录协议

主要：保密+完整性

过程：分片 (2^14)——压缩后追加HashMAC——对称加密——加上SSL记录协议头——TCP——解密——解压缩（拼接）

SSL记录协议头：(内容类型，主次版本，压缩长度)

## SSL握手协议

主要：身份认证（先认证S，C可以不认证），协商算法，协商会话密钥

报文结构 1B类型 3B长度 1+B参数内容

### 握手协议消息类型

消息	参数	描述
<b>hello_request</b>	<b>Null</b>	服务器发出此信息给客户端启动握手协议
<b>client_hello</b>	版本，随机数，会话 <b>id</b> ，密码参数，压缩方法	客户端发出 <b>client_hello</b> 启动 <b>SSL会话</b> ，该信息标识密码和压缩方法列表，服务器响应
<b>server_hello</b>		
<b>certificate</b>	X.509 v3证书链	服务器发出的向客户端验证自己的消息
<b>server_key_exchange</b>	参数，签名	密钥交换
<b>certificate_request</b>	类型， <b>CAs</b>	服务器要求客户端认证
<b>server_done</b>	<b>Null</b>	指示服务器的 <b>Hello</b> 消息发送完毕
<b>certificate_verify</b>	签名	对客户证书进行验证
<b>client_key_exchange</b>	参数，签名	密钥交换
<b>finished</b>	<b>Hash值</b>	验证密钥交换和鉴别过程是成功的

分四个阶段完成握手

- 建立安全能力

- Chello
  - ver,random,sessionid, cipher\_suite(KEmethod, hash&encoder), compression
- Shello
  - same as top
- 服务端身份认证和密钥交换 (可以失败)
  - certificate X.509v3 匿名DH可能不需要)
  - SKE
    - (DH RSA需要交换密钥)
  - certifica\_req
    - 需要客户的certi
  - S\_hello\_done
- 客户端身份认证和密钥交换
  - certificate
    - 可选, 也可能没有
  - CKE
    - DH RSA需要交换密钥
  - certifica\_verify 签名此消息
    - 用于配合Cert让服务器验证客户端的数字证书所有权
- 完成阶段
  - change\_cipher\_spec
    - 把pre转化成主密钥, 派生出所有密钥
  - finished
  - change\_cipher\_spec
    - 你算完我也算完了,
  - finished

**master secret** 主密钥, 用于生成一堆密钥

CS双方的MAC密钥, 加密密钥和MAC初值向量IV

## SSL告警

两个字节

- 1告警 2致命错误: 终止连接但不终止会话, 不再会话中建立新连接
- 包含告警信息的代码

## SSL 修改密码协议

一个字节“1”, 握手结束后发送, 以后记录用刚才的算法和密钥来加密认证。

接收方把挂起会话恢复到当前状态

## 安全分析

- 一个安全协议除了所采用的加密算法安全性以外，更为关键的是其逻辑严密性、完整性、正确性，SSL比较好地解决了这一问题
  - SSL协议可以很好地防范“重放攻击”
    - SSL协议为每次安全连接产生一个128位长的随机数“连接序号”，攻击者无法提前预测此连接序号，因此不能对服务器请求做出正确应答
  - 几乎所有Web服务器以及各类操作系统上的web浏览器支持SSL协议，因此使用SSL协议开发成本小
- 保密问题：SSL协议的数据安全性其实就是建立在RSA等算法的安全性上，从本质上讲，攻破RSA等算法就等同于攻破此协议
- 认证问题：SSL对应用层不透明，只能提供交易中客户与服务器间的双方认证，在涉及多方的电子交易中，SSL协议并不能协调各方面的安全传输和信任关系

## 应用层

### WEB和安全威胁

网页服务器与浏览编辑器WWW——HTML、HTTP、URL

特点：

- 双向互联网
  - 服务器容易受到攻击
  - 金钱和信息丰富且重要
- 
- 服务器存储安全（用户认证，访问控制，日志）
  - 客户端安全（服务器认证，访问控制和签名）
  - 信息传输安全（IPsec+SSL/TLS+MIMEPGPSET）

IP级、TCP级和应用级安全

### HTTP攻击举例

HTTP：主要是规定浏览器和WWW服务器通信规则，裸奔html，当时是信息没有财富

明文传输，不检查完整性 + 无状态，不验证身份——监听明文，篡改劫持，伪造服务器钓鱼

中间人加入并分别建立和通讯双方的连接，可以被随便改——ARP欺骗

### ARP协议

数据链路层，维护IPMAC映射表

- A知道B的IP地址，在映射表中查不到B的MAC地址，就广播需求：“谁知道IP地址为B的MAC地址，请告知”
- B发现这个请求，先将A的“IP-MAC对”存入自己的映射表，再回复A：“IP地址为B的MAC地址是\*\*\*\*\*”
- A接到B的回复后，将B的“IP-MAC对”存入自己的映射表

可以伪造ARP frame改变网内任何主机的映射表，来切断通讯并且把自己的物理MAC加入。即发对方以为就是发攻击者

## DNS

域名——IP对应

欺骗：所有主机名都对应攻击者的IP

## HTTPS = HTTP + SSL

明文——加密（记录协议加密）

80端口到443端口

无状态连接——用SSL+HTTP协议加密和身份认证的连接

- **用户：** 浏览器里输入 <https://www.sslserver.com>
- **HTTP层：** 将用户需求翻译成HTTP请求
- **SSL层：** 借助下层协议的信道安全的协商出一份加密密钥，并用此密钥来加密HTTP请求
- **TCP层：** 与web server的443端口建立连接，传递SSL处理后的数据。接收端与此过程相反
- **SSL在TCP之上建立了一个加密通道，通过这一层的数据经过了加密，因此达到保密的效果**

## SSL能否保证安全

ARP和嗅探都可行，但是看不懂的密文

**篡改：** 密文无法篡改，也看不懂

https有小锁，显示发放的证书来认证服务器身份。如果**钓鱼或者恶意代理**，没有受信任的证书（钓鱼wifi），有危险

攻击者用自己**证书替换**服务器证书：没办法，开摆，明文了

**会话劫持** 偷不到用户密码，但是登陆之后跳转到HTTP页面可以偷取Cookie来冒充用户

## SET安全电子交易

定制给电子商务安全

涉及到多方：客户、商家、银行、相关管理认证部门

## 威胁

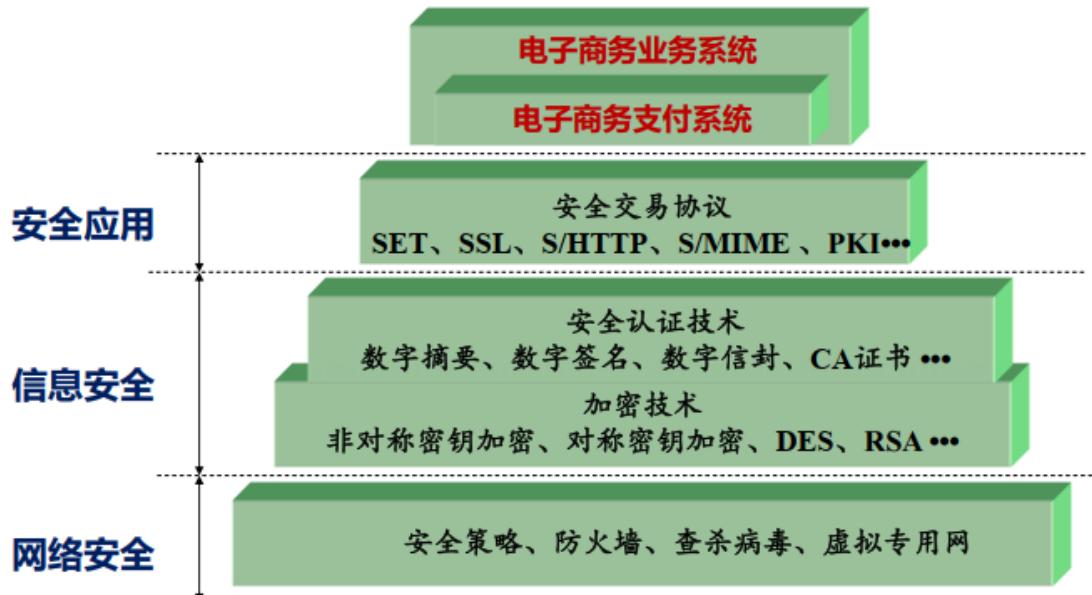
- 支付账号密码的窃取
- 金额更改
- 商家和支付方的互相确认
- 双方的抵赖，否认
- 故意延迟

## 需求

- 所有数据的保密性和完整性。银行不应该知道我买的东西，京东不应该知道银行的余额。我买的是什么，多少钱就是多少
- 结算双方身份的认定，唯一确定的身份
- 不抵赖和否认

- 可靠快捷

体系结构：社会——管理——技术——应用



### ● 网络系统安全：针对物理技术系统的安全问题

- 保证网络设施的正常运行
- 避免受到外界的恶意攻击

可靠安装、维护、管理  
设置防火墙、防止病毒

### ● 网络信息安全：针对商务逻辑系统的安全问题

- 信息保密、信息完整
- 身分认证、不可抵赖
- 信息有效

加密技术  
认证技术

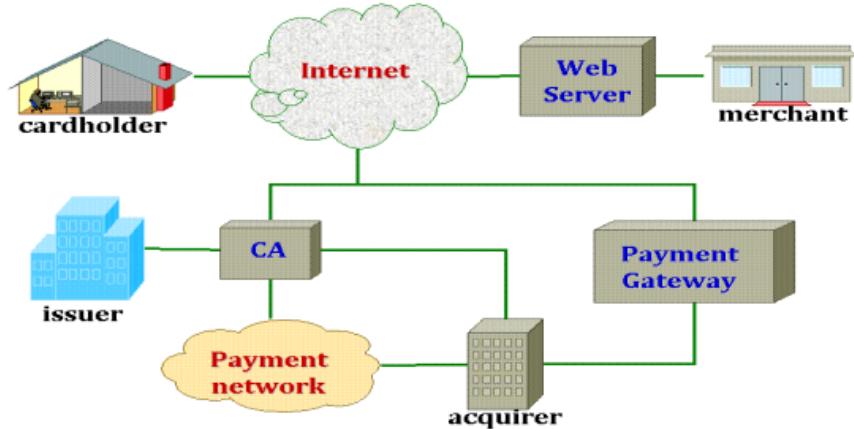
### ● 网络交易安全

- 参与对象之间交易过程的安全，如安全套接层协议（SSL）、安全电子交易协议（SET）、公钥基础设施（PKI）

## SET安全电子交易

visa和master card搞的，基于X.509v3证书。私密保密+完整+双方认证+抗抵赖

- 持卡人(cardholder)
- 商家(Merchant )
- 发卡方(Issuer)
- 收款行(Acquirer)
- 支付网关(Payment Gateway)
- 证书权威(CA)



下半部分是中国银联，属于银行网络系统了。

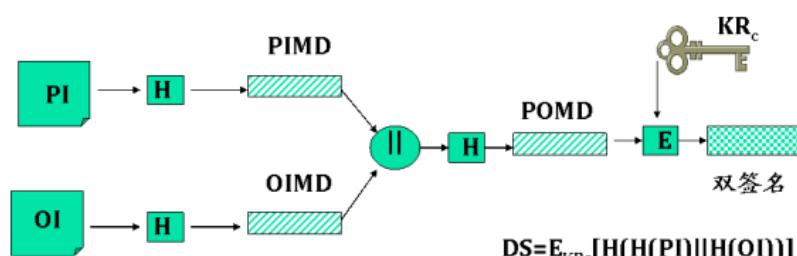
支付网关：由收款行操作，处理商家的支付报文，属于SET和银行支付网络的接口

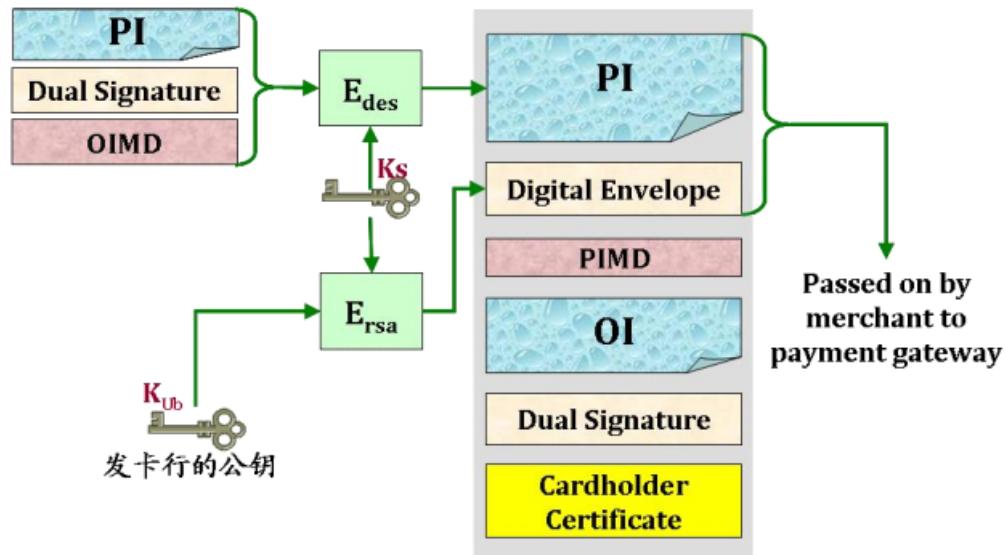
收款行去找对应的发卡行协商请求支付

**order info 和 payment info** 背靠背，互相不可见。必须分开加密和签名，但是又不能分开发不然难以确认。

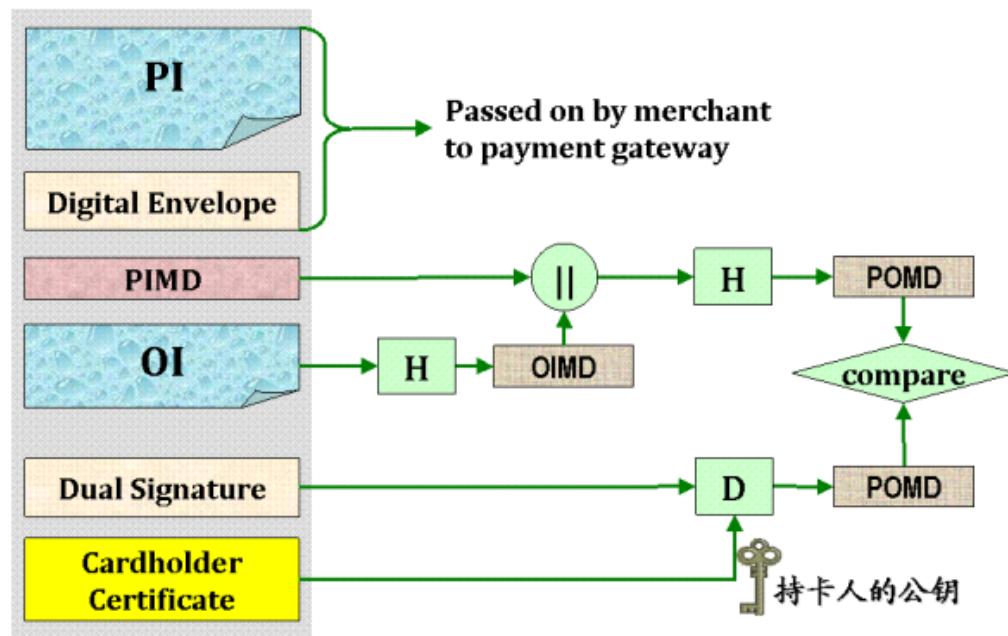
对不能读信息的一方，只给MD作为验证，但是原文无法解密

- 双方开户并获得资质证书
- 【1购买请求】用户向商家发起购买请求
  - 发起请求：ID和卡信息
  - 发起响应：商家私钥，交互ID，整数
  - 购买请求：验证证书，
    - 生成PIOI，把transactionID放入PIOI，然后生成购买请求
    - 商家需要对OI做屏蔽（加密）放入支付网关（收款行）
    - 双签名，两层哈希。连接发送给不同接受者的报文
- 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
  - KR<sub>c</sub>=顾客的私有签名密钥





- 购买响应
  - 商家收到购买请求，验证订单



- 【2支付授权】商家通过支付网关，向发卡方请求支付授权
  - 授权请求
    - PI 双签名 Es 双方证书
    - 银行要验证所有证书来核验双方身份
    - 解密数字信封获得Es，解密authorization block
    - 验证商家前缀ing
    - 解密payment block数字信封，解密PI
    - **验证双签名**
    - 验证transaction ID和PI
    - 从发卡行**申请支付**
  - 授权响应
    - 授权信息和权标
    - 证书
- 【3支付获取】商家通过支付网关，向发卡方请求付款

- 获取请求
  - 支付量, 交易ID, 权标, 商家的签名密钥、证书
- 获取响应
  - 网关签名, 加密获取数据块, 网关签名证书