该重点为20年秋季学期密码学老师期末画的重点，**不过没有椭圆曲线部分，**自己补充。

# 一 概述

1. 信息安全三要素

**保密性(Confidentiality)：**使截获者在不知密钥条件下不能解读 5

**完整性(Integrity)：**保证信息从真实的发送者传送到真实的接收者手中，传送过程中没有非法用户添加删除和替换等

**可用性（Availability）**：是指保障信息资源随时可提供服务的能力特性/保证经过授权的客户能及时准确的不间断的访问数据

**认证性**：使任何不知密钥的人不能构造一个密报，使意定的接收者脱密成一个可理解的消息 (合法的消息)。

**Kerckhoff 原则：**系统的保密性不依赖于对加密体制或算法的保密，而依赖于密钥。

1. 古典密码

根据是对单个字母逐个进行还是多个字母同时进行，分为单表代换和多表代换

P9

单表代换

1）凯撒密码，对每个字母进行循环移位（3）

2）移位变换，移位k

3）仿射变换，变成一个线性函数，涉及到求逆元，有两个密钥

多表代换

将n个字母进行分组，M1,M2,M3…

每个分组进行加密，类似于放射变换，只是维度不同A，B密钥

基本原理，加解密

**要掌握**

3. 密码分析

**3.1分析方法**

1. **穷举破译法**

对截收的密文依次用各种可解的密钥试译，直到得到有意义的明文；或保持密钥不变的，对所有可能的明文加密直到得到与截获密文一致为止，此法又称为完全试凑法。

为了减少搜索计算量，可以采用改进的试凑法，将密钥空间划分成几个等可能的子集，对密钥可能落入哪个子集进行判断。

1. **分析破译法**

①确定性分析法：利用一个或几个已知量 (如已知密文或明文-密文对)用数学关系式表示出所求未知量 (如密钥等)。

②统计分析法：利用明文的已知统计规律进行破译。

**3.2密码可能经受的不同水平的攻击**

依据攻击者知道的信息多少，密码可能经受的不同水平的攻击：

**唯密文攻击：**分析者仅拥有的截获密文，试图进行分析得出明文或密钥。

**已知明文攻击 ：**分析者已有很多明文-密文对，试图进行分析得出明文或密钥。

**选择明文攻击：**分析者可以选定任何明文-密文对来进行攻击，以确定未知的密钥。

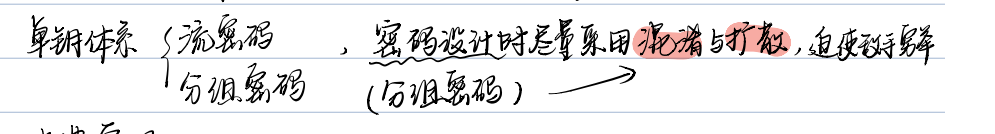
**选择密文攻击：**分析者可以任意选择密文，并可获得相应明文

***CPA安全***：CPA（Chosen-Plaintext Attack），选择明文攻击

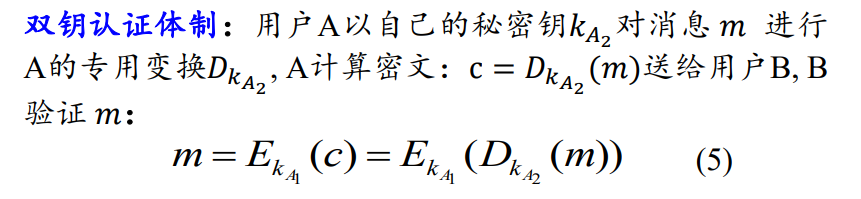
* **密码可能经受的不同水平的攻击**

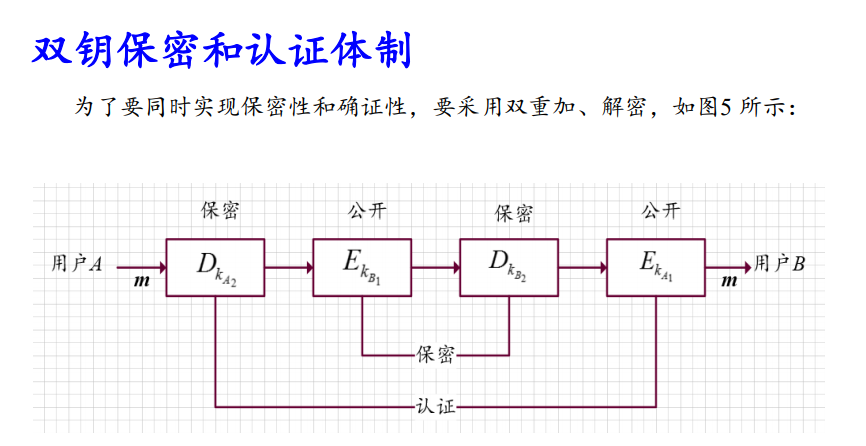
4. 密码体制分类

1. 单钥体制：加密密钥和解密密钥相同主要问题是密钥产生和密钥管理（流密码/分组密码）包括流密码，分组密码



1. 双钥体制：加密密钥和解密密钥不同 Diffie 和 Hellman 1976 年首次提出，用户有公钥k1，私钥k2，公钥可以公开从而将加密和解密能力分开 安全性：可实现对A所发消息的验证





# 二 流密码

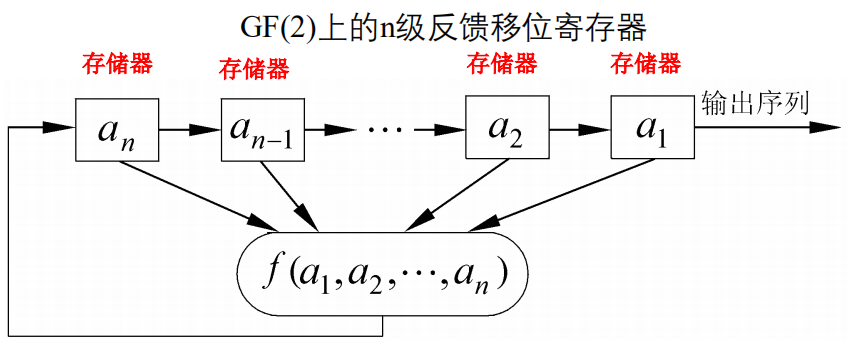
1. 基本概念、分类

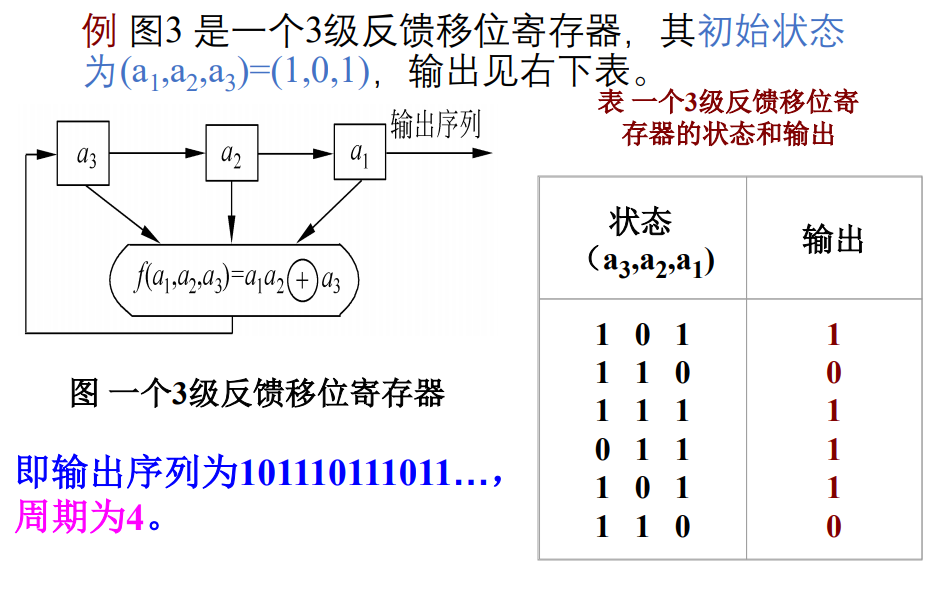
流密码是将**明文划分成字符（如单个字母）**，或其编码的基本单元（如0，1数字），每一字符分别与密钥流对应字符"作用"，从而进行加密，解密时以同步产生的同样的密钥流实现。

流密码强度完全依赖于密钥流生成器生成密钥流的随机性和不可预测性

1. **同步流密码：*密钥流产生算法和明文（密文）无关***。
2. **自同步流密码：**密钥流产生算法和明文（密文）相关。
3. n级反馈移位寄存器

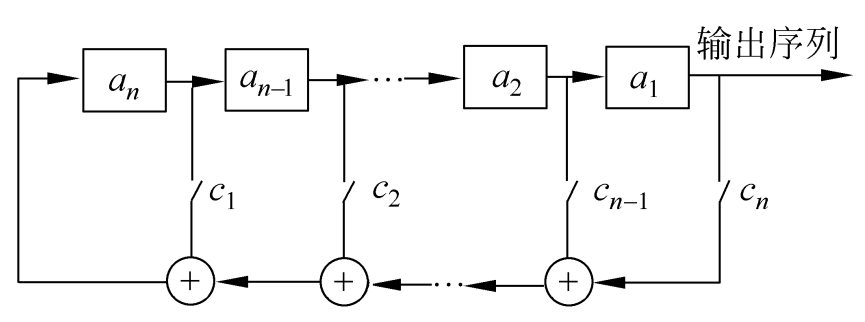
GF(2)上一个n级反馈移位寄存器由n个二元存储器与一个反馈函数f(a1,a2,…,an)组成。





1. 线性反馈移位寄存器(LFSR)

GF(2)上的n级线性反馈移位寄存器，f(a1,a2,…,an)= cna1⊕cn−1a2⊕⊕c1an。



LFSR输出序列的性质：完全由其反馈函数决定。

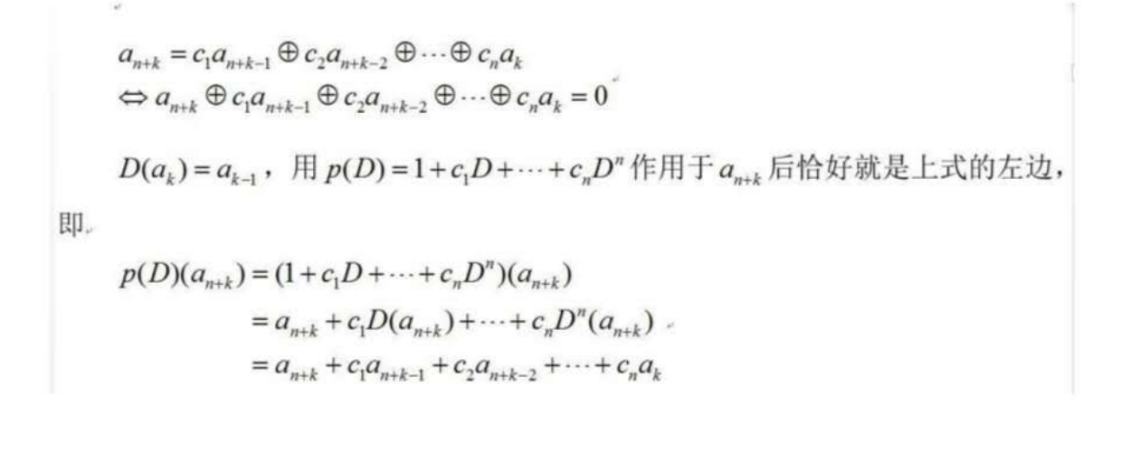
n级LFSR状态数：最多有2n个。

n级LFSR的状态周期： ≤ 2n-1。

输出序列的周期=状态周期，≤ 2n-1。

选择合适的反馈函数可使序列的周期达到最大值2n-1，周期达到最大值的序列称为m序列。

**LFSR的特征多项式：**p(x)=1+c1x+…＋cn-1xn-1＋cnxn



1. m序列

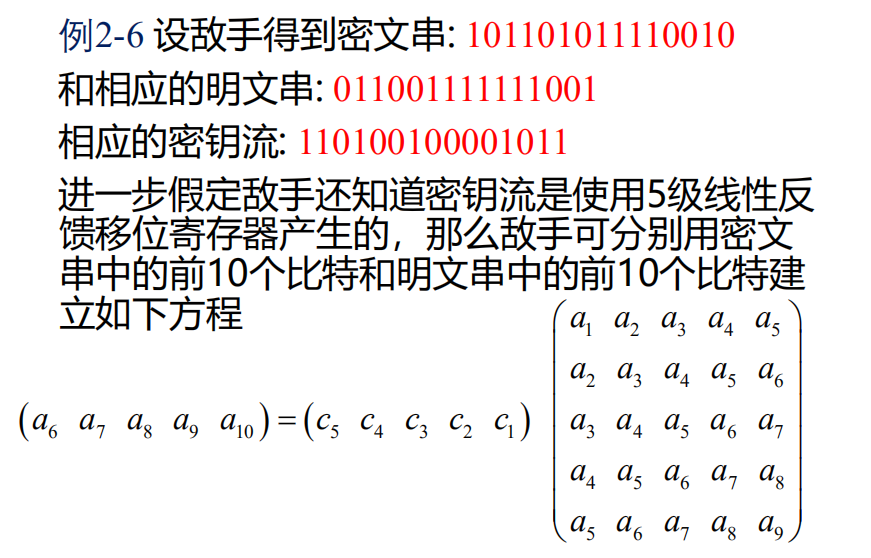
伪随机序列：秘钥流不可能做到随机，只能要求截获比周期短的一段序列时不会泄露更多信息（这样的序列就称作伪随机序列）

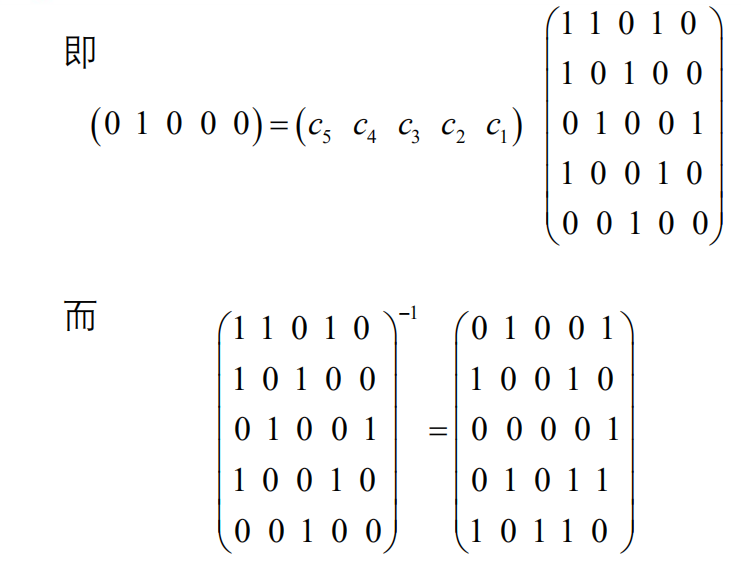
游程：连续的0或者1的个数

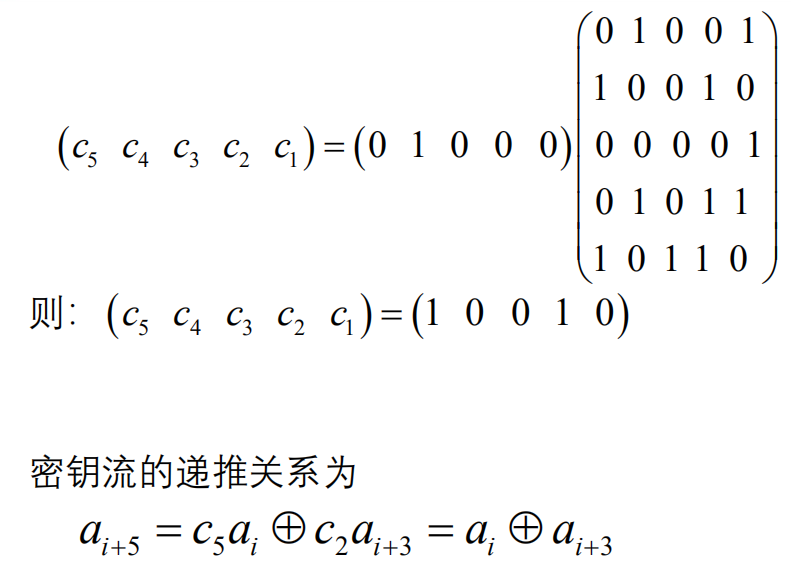
GF(2)上周期为T的序列{ai }的自相关函数

设序列{ai}满足线性递推关系：

**M序列的破译（具体知识点可以看书上，比较难理解，通过例题会比较容易懂）**







# 三 分组密码

1. 基本概念
2. **分组密码**：若明文流被分割成等长串，各串用相同的加密算法和相同的密钥进行加密，就是分组密码。
3. **分组密码优缺点、与流密码对比：**

①分组密码的加解密算法（E，D）简洁快速，所占用的计算资源小，易于软件和硬件实现。（一般来说，用硬件实现时，流密码比分组密码更简单快速；用软件实现时，分组密码比流密码更简单快速）

②加解密算法（E，D）参数固定，比流密码更容易实现标准化。

③由于明文流被分段加密，因此容易实现同步，而且传输错误不会向后扩散。

但是分组密码的安全性很难被证明，至多证明局部安全性。

1. 对分组密码的攻击

分组密码的密钥z

被重复使用，即多次一密。 因此最主要的威胁就是已知明文攻击。有两种已知明文攻击：穷举、解方程

如果某一组明文/密文对（*m*，*c*）使得方 程*m*=*D*(*c*, *z*)特别容易解出*z*，*m*就称为一个弱明文， *z*就称为一个弱密钥。加解密算法（E，D）不能存在弱明文和弱密钥

为了抵抗已知明文攻击（甚至选择明文攻击），分组密码应该满足的性质

混淆性：所设计的密码应使得明文、密文、密钥间的依赖关系相当复杂，以至于这种依赖关系对密码分析者来说是无法利用的。

扩散性：所设计的密码应使得

（1）密钥的每一个比特影响密文的每一个比特，以防止对密钥进行逐段破译；

（2）明文的每一个比特影响密文的每一个比特，以便最充分地隐蔽明文。目的：抵抗攻击者对密码系统的统计分析

高非线性度：抵抗线性密码分析的强度就是非线性度

**3.分组密码的设计准则**

安全性：从任何角度难以攻破。

简洁性 ：分组密码算法在满足安全性的同时尽可能简单快速。

有效性 ：分组密码的设计应使密钥最大限度地 起到安全性作用。

透明性和灵活性：透明性即要求算法是可证明安全的，灵活性即要求算法的实现可以适应多种计算环境；明文分组长度可以伸缩；算法可以移植和变形。

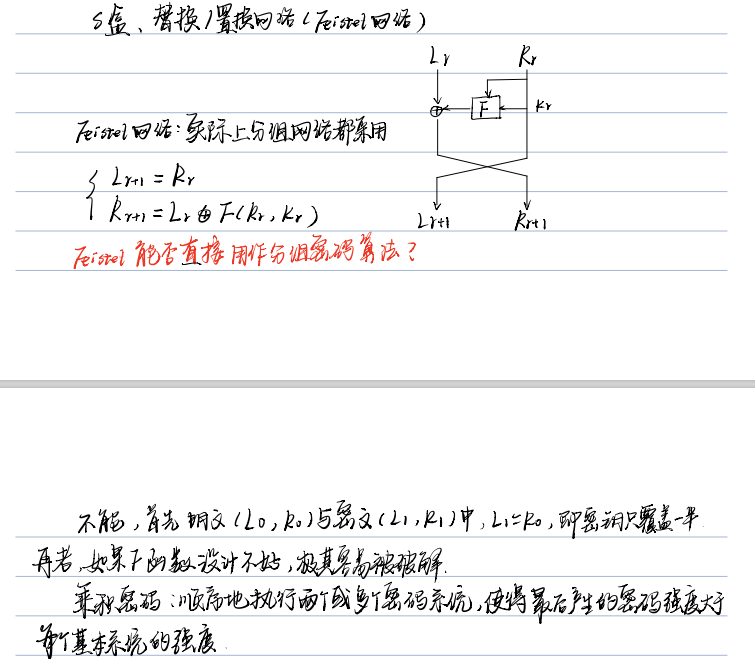
加解密相似性：加密算法和解密算法相同仅仅 密钥的编排不同。

这里举的两个例子，异或和模2n加

**4.分组密码设计**

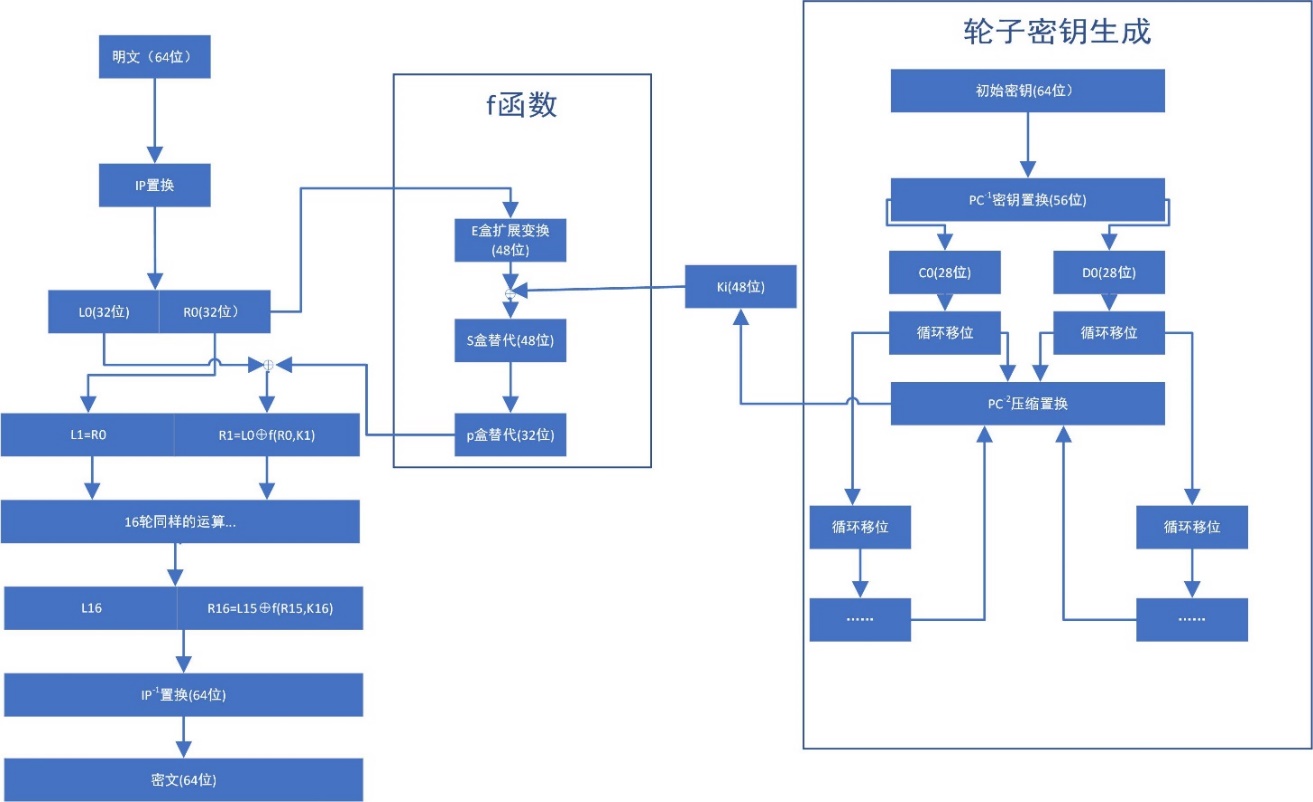
替换/置换网络（SPN）：集掩蔽、混淆、扩散为一体的综合性部件，由若干子部件组合而成典型代表就是Feistel网络，分组密码的结构从本质上说都是基于一个称为Feistel网络的结构。

Feistel网络不能用作分组密码算法。Feistel网络的基本模块是F-函数。

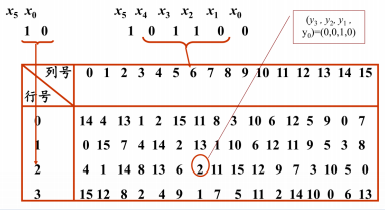
1. 函数：一个依赖于密钥的把输入串映射到输出串的映射，通常是非线性且不可逆的

S盒：输入/输出长度比较小的，用输入/输出真值表来 计算的，仅仅实现高度非线性功能的计算部件称为S盒

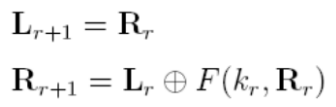
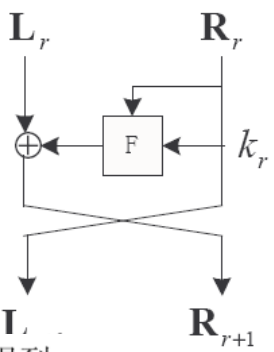
DES算法原理：



DES:



三重DES： 先加密后解密在加密

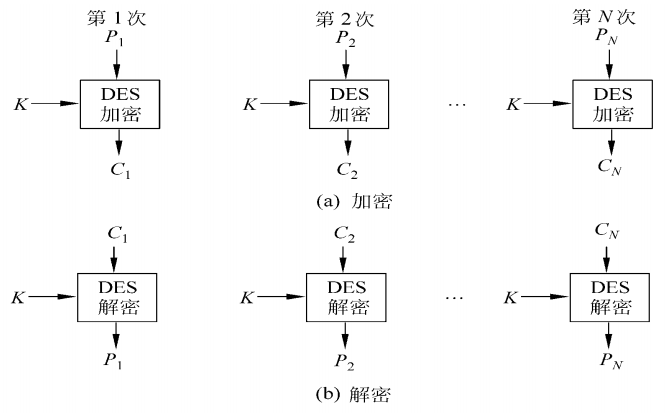


AES的明文分组长度是可变的：128,192,256bit

AES的密钥长度：128,192,256bit

**5.五个工作模式**

5.1 电码本(ECB)模式 ：每个明文分组独立加/解密



优点：

**简单、高速**

**无差错传播：单个密文分组出现错误只会影响该分组的解密，不会影响到其他分组**

缺点：

（**1**）相同明文分组对应相同密文分组

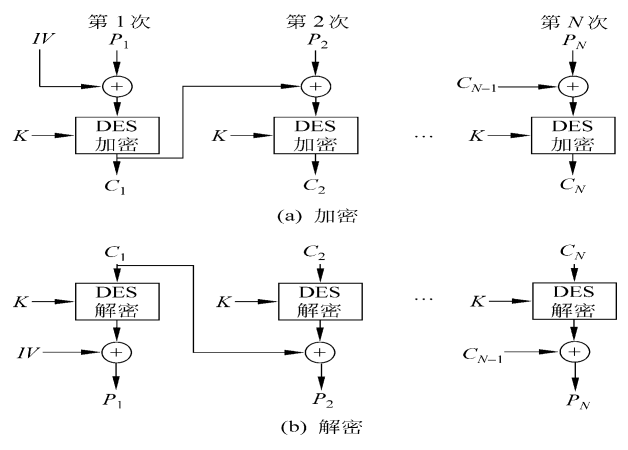
（**2**）不能隐蔽明文分组的统计规律和结构规律**,**不能抵抗替换攻击

应用：

（**1**）用于随机数的加密保护

（**2**）用于单分组明文的加密

5.2 密码分组链接(CBC)模式 ：每个明文分组先与前一密文分组异或，再进行加密

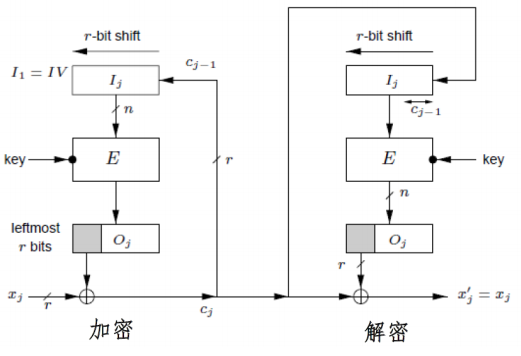


明文块的统计特性得到了隐蔽

有限差错传播：单个密文分组出现错误会影响该分组和后面一个密文分组的解密

具有自同步功能

5.3密码反馈(CFB)模式 ：加密消息需按字符、字节或比特处理时，可采用CFB模式



实现简单（只要求加密算法）

比CBC慢很多，每次只有少数比特完成加密

单个密文分组出现一个比特错误，不仅影响该分组，还会最多影响后续[64/j]个密文分组的解密

5.4 输出反馈(OFB)模式（没看明白）

5.5 计数器模式（没看）

四类工作模式比较和选用

ECB模式简单、高速，但最弱，易受重放和替换攻击，

一般用于加密长度小于等于分组长度的消息。

CBC，CFB，OFB模式的选用取决于实际的特殊需求。

明文不易丢信号，对明文的格式没有特殊要求的环境可选用CBC模式。需要完整性认证功能时也可选用该模式。

容易丢信号的环境，或对明文格式有特殊要求的环境，可选用CFB模式。

不易丢信号，但信号特别容易错，且明文冗余特别多，可选用OFB模式。

# 四、公钥加密

1. 公钥体制的基本原理是**陷门单向函数**。**陷门单向函数和单向函数不一样**

陷门单向函数(Trapdoor one-way function)，是这样的单向函数：

在不知陷门信息下，由f(x)求x“极为困难” ，

当知道陷门信息后，由f(x)求x是易于实现的

单向函数举例：（各自对应一种加密方法）

离散对数

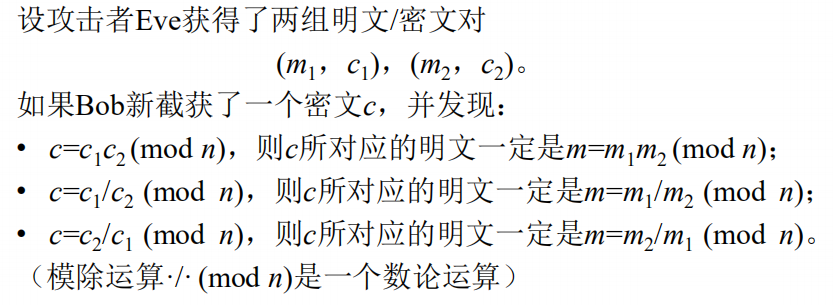
大整数分解

背包问题

**RSA**

**重点记忆：密钥生成、加密、解密、安全性**

基本RSA的一个安全性漏洞：**可传递性**



这个漏洞使得攻击者Eve对某些新的密文能够轻而易举地找到其对应明文。这个漏洞还有更深刻的隐患，比如在消息认证过程中容 易产生伪造。

**步骤：**

第一步：选择两个大素数p和q

第二步：计算p和q的乘积n（n转化成二进制一般是1024位，重要的话2048位）

第三步：计算n的欧拉函数φ(n)=（p-1）\*（q-1）

第四步：随机选整数e，1< e < φ(n)，且e与φ(n)互质，计算e对于φ(n)的乘法逆元d

ed ≡ 1 (mod φ(n))

第五步：将n和e封装成公钥，n和d封装成私钥

※加密：c≡ m^e mod n; c为密文，且0<=c<n

※解密：对于密文0<=c<n，解密算法为：m≡ c^d mod n;

**RSA的安全性分析**

2511<p<2512

**这样要穷举2510次，似乎足够安全**

**背包密码（可能出计算）**

加密：

设明文m=(m1, m2, m3, …, mn)是长度为n的比特串。使用公钥 {b1，b2， b3，…bn} 计算密文 c ： c=m1b1+m2b2+m3b3+…+mnbn。 密文c是一个正整数。 （密文c是背包重量，由n个物品重量b1，b2，b3，…，bn中 的某些物品重量相加而成。若截获了密文c，又知道n个物 品重量b1，b2，b3，…，bn，求解明文m就是背包问题。 ）

解密：

使用私钥{a1，a2，a3，…，an} ， M，U，计算变换课文C：

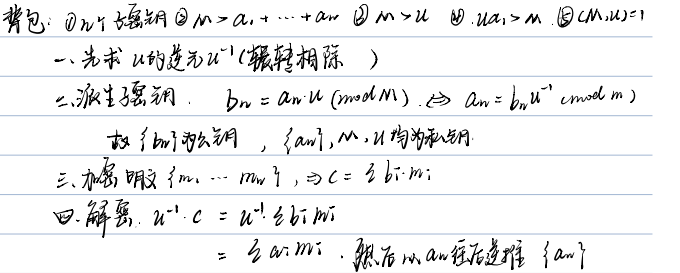
C=U-1c(mod M)

=U-1(m1b1+m2b2+m3b3+…+mnbn )(mod M)

=m1a1+m2a2+m3a3+…+mnan(mod M)

=m1a1+m2a2+m3a3+…+mnan。

根据定理中的方法，容易解出明文m。



**Rabin**

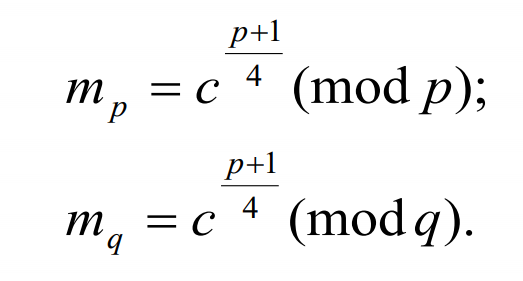
第一步：选择两个大素数p和q，要求p和q都是4的倍数加上3

第二步：n=p\*q

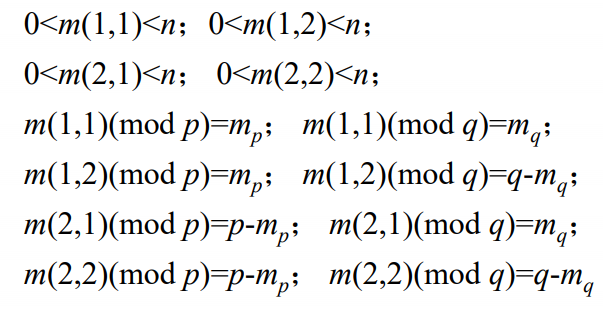
第三步：公钥n，对外公布；私钥（p,q）则自己收着。

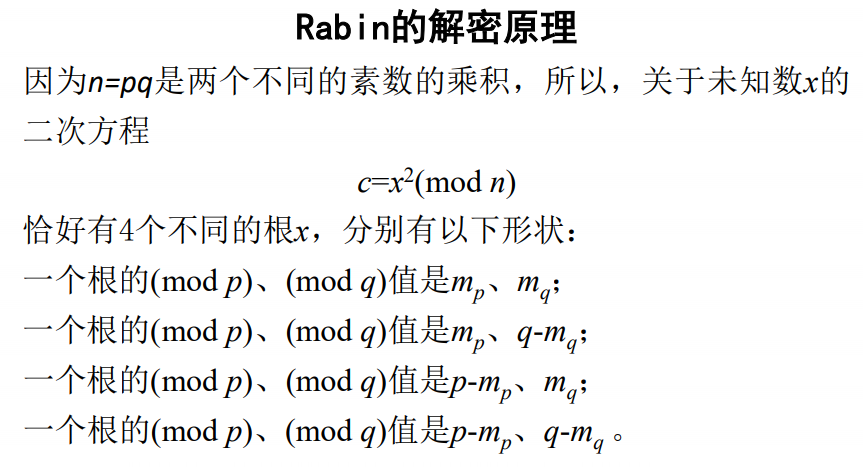
Rabin加密体制加密过程：

明文m范围(0,n)计算密文方法：c=m2(mod n)，c是密文B在收到密文后，用私钥解密：

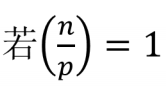


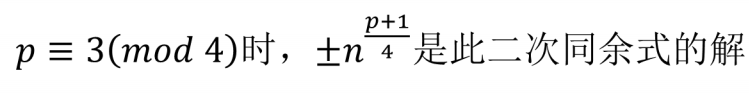
计算四个数m(1,1),m(1,2),m(2,1),m(2,2)满足：

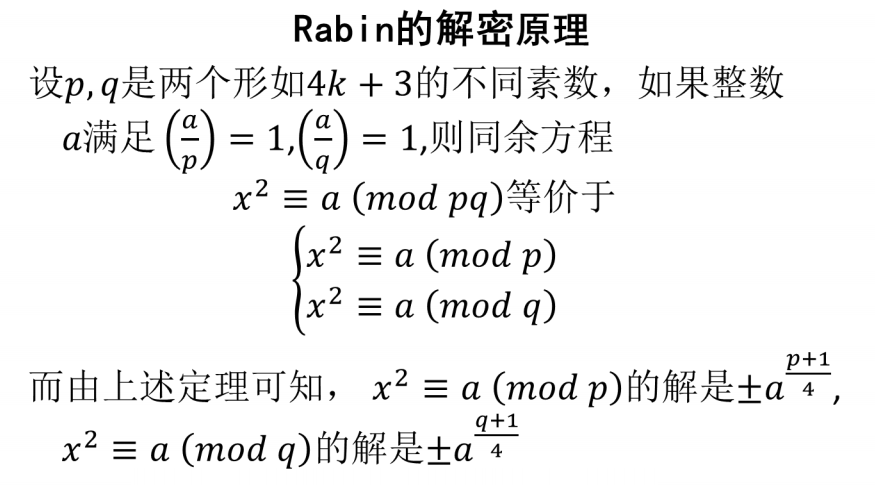


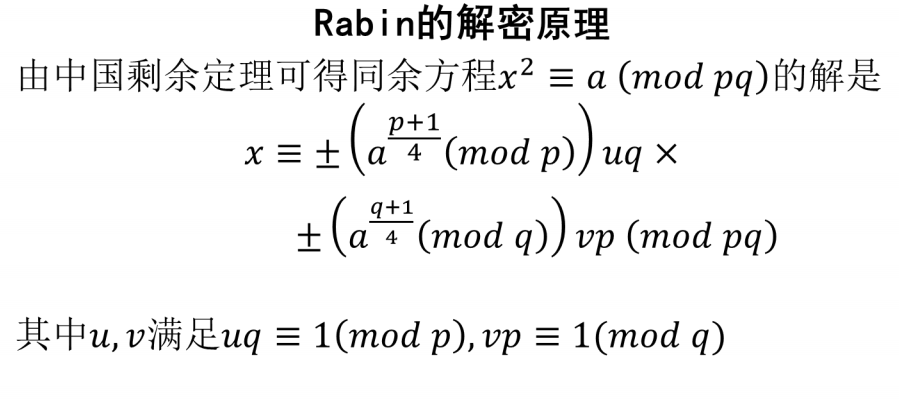


Legendre的符号表示：

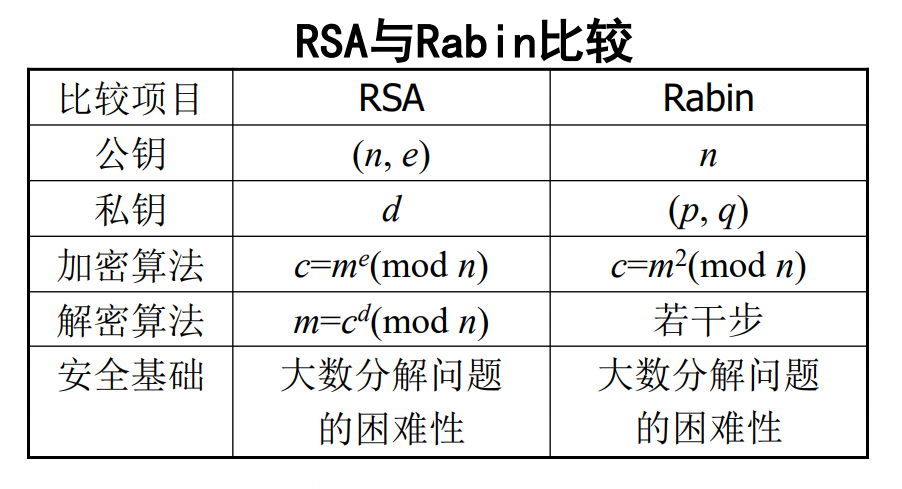
，表示n是模p的平方剩余，二次同余式此时有解



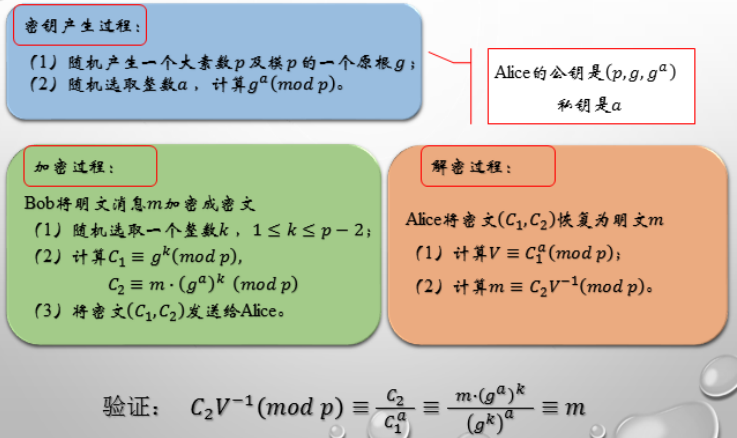




求n的分解式n=pq是大数分解问题。



# ElGamal



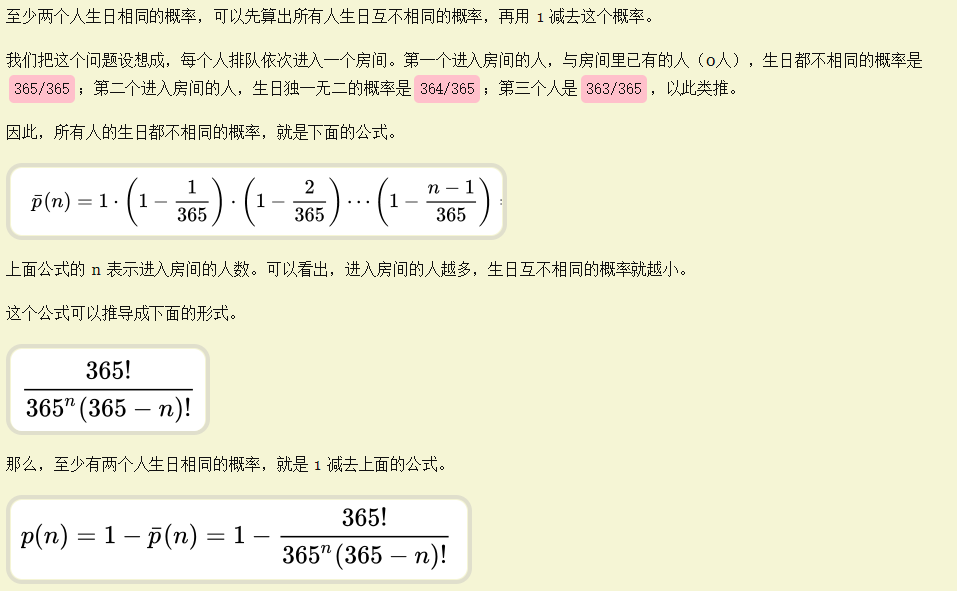
**特点**：密文由明文和所选随机数*k*来定，因而是**非确定性**加 密，一般称之为**随机化**加密，对同一明文由于不同时刻的 随机数*k*不同而给出不同的密文。代价是使数据扩展一倍。

# 五、哈希

**1、杂凑函数**

将任意长度的比特串x压缩成为固定长度的比特串y、单向性、无碰撞性

**2、生日悖论**



生日攻击告诉我们：为了能达到n-bit的安全性，你所选择的Hash函数的散列值长度应该是2n。

**3、数字签名应具有的性质：**

完整性：一个被签了名的消息，无法分割成为若干个被签了名的子消息。

身份唯一性（不可伪造性）：被Alice签名的消息只能由Alice生成

不可否认性（公开可验证性）：被Alice签名的消息，在未来不能被Alice否认。

**4、公钥密码的签名方案（一）**

私钥签名，公钥验证

安全性分析：

第一种：选定消息m，对签名值s进行伪造

Eve没有Alice的私钥是无法伪造出签名s的，所以对事先设定的消息m来说，签名消息(m，s)具有身份唯一性和不可伪造性

第二种：给定签名值s，反过来对消息m进行伪造

Eve拥有公钥，对给定签名值，可以直接得到消息m，**攻击成功。**为了抵抗这种攻击，合法签名消息(m，s)中的消息m必须是有意义的内容，而不是乱码。

**5、公钥密码的签名方案（二）**

加入了Hash函数

Alice先用Hash函数对消息m进行处理，再使用私钥加密，将消息和签名一起发出Bob用Hash对消息m进行处理，使用公钥解密签名，对比二者是否一样

安全性分析：

第一种：选定消息m，对签名值s进行伪造

Eve没有Alice的私钥是无法伪造出签名s的

第二种：给定签名值s，反过来对消息m进行伪造

Eve拥有公钥，对给定签名值，可以直接得到消息的Hash值H(m)，却无法得到消息m。

此时该签名方案似乎具有身份唯一性和不可伪造性但是，如果Eve截获了许多Alice的签名消息，还允许发送，Eve可以一个个试，称为重放攻击。

解决方案：不得重复发送、时间戳

* RSA数字签名

（1） Alice用H将消息m进行处理，得散列值h=H(m)。

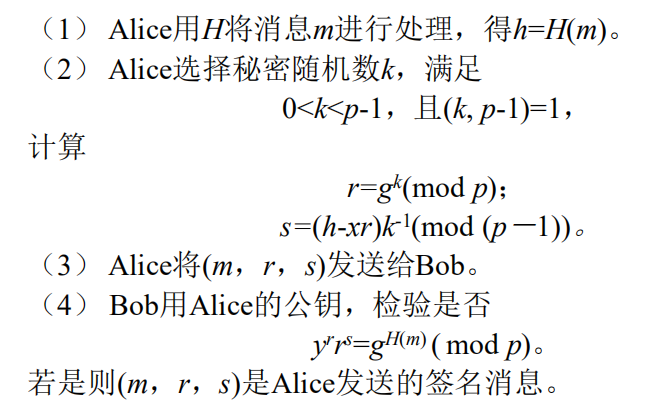
（2） Alice用自己的私钥d对h “解密”得s=hd(mod n) 。

（3） Alice将(m，s)发送给Bob。

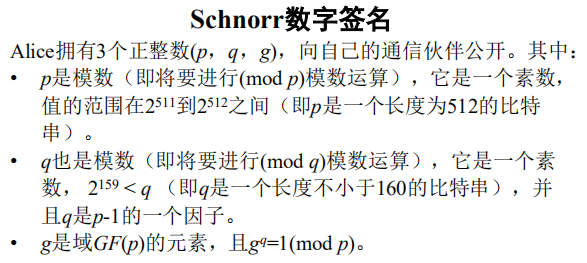
（4） Bob用Alice的公钥e，检验是否H(m)= se(mod n)。

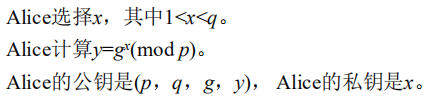
若是则(m，s)是Alice发送的签名消息。

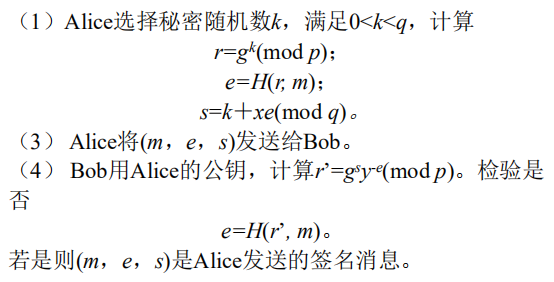
* ElGamal数字签名



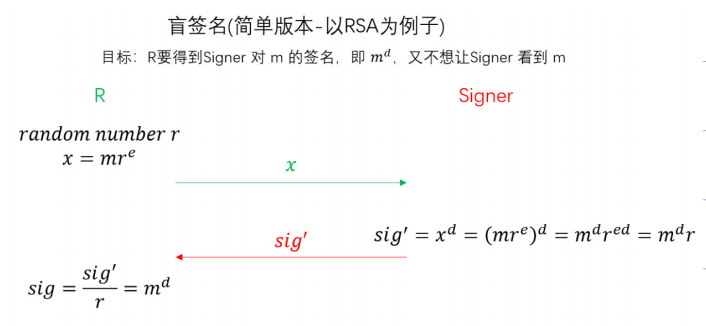
* Schnorr数字签名（看一看）







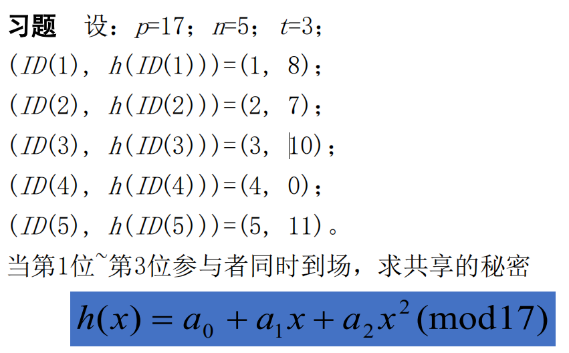
盲签名：一般数字签名中，总是要先知道文件内容而后才签署。但有时 需要某人对一个文件签名，但又不让他知道文件内容，称此为 盲签名(Blind Signature)，它是由Chaum[1983]最先提出的。在 选举投票和数字货币协议中将会碰到这类要求。 设B是一位仲裁人，A要B签署一个文件，但不想让他知道所 签的是什么，而B也并不关心所签的内容，他只是要确保在需 要时可以对此进行仲裁



# 六、密码技术应用

* Shamir的秘密共享门限方案**（掌握孙子定理！不然不会做题）**

多项式h(x)（即多项式h(x)的系数{a0, a1, a2, …, at-2, at-1}）就是n个参与者所共享的秘密。



a0=13,a1=-7,a2=2,

检测结论（一）

当某t个参与者组合计算出的系数向量与另外t个参与者组合计算出的系数向量不相等时，这两次被组合的全部人员中必有“欺骗者”。

检测结论（二）

当某t个参与者组合计算出的系数向量与另外t个参与者组合计算出的系数向量相等时，（几乎可以断定）这两次被组合的全部人员都是诚实的参与者。

* 不经意传输(OT)

Alice将秘密通过不经意传输协议发送给Bob， Bob成功地获得秘密的概率只是

1/2。而且，Alice无法知道Bob是否得到了秘密。

* 零知识证明

P使用一种证明方法，让V相信他知道该秘密，又能保证不泄露该秘密。

# 七、密钥协商



我们可以为n个用户推断这种简单方案的几个功能：

■每个用户必须存储n-1个密钥。

■网络中总共有n（n-1）≈n2个密钥。

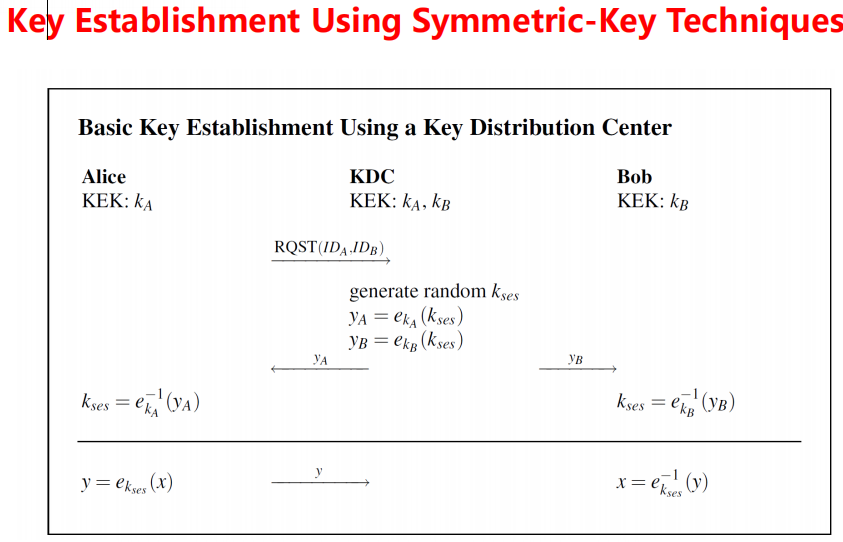
■总有n（n-1）/ 2个对称密钥对在网络中。

■如果新用户加入网络，则必须与每个用户建立安全通道

其他用户以便上传新密钥。

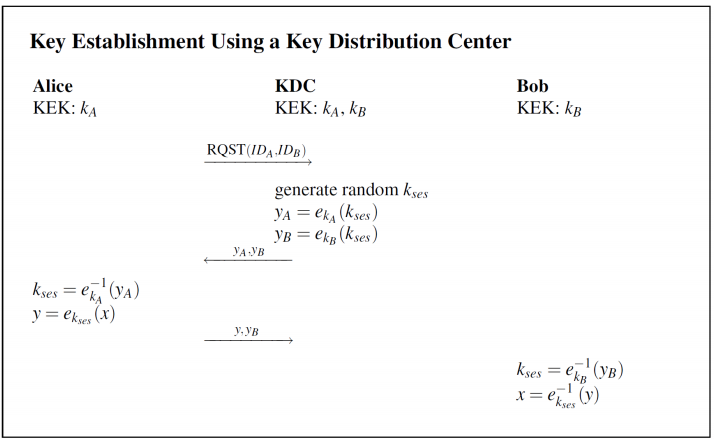
1. **基于对称密钥技术的密钥协商**

分放双方方案



密钥分发中心随机生成一个会话密钥，使用Alice和Bob的公钥加密后分别发给二者，两人分别用自己的密钥进行解密得到会话密钥。KDC(key distribution center)**包括了通信二者的密钥**。

发送一方方案：



密钥分发中心随机生成一个会话密钥，分别使用Alice和Bob的公钥加密后都发给Alice，Alice 用自己的私钥解密得到会话密钥，然后加密信息x得到y，将加密后的信息和Bob的那一份会话密钥一起发给Bob (y,yB)。Bob在收到信息后先根据yB解密得到会话密钥，然后用会话密钥解密信息y得到明文x。

**存在的安全性问题：**

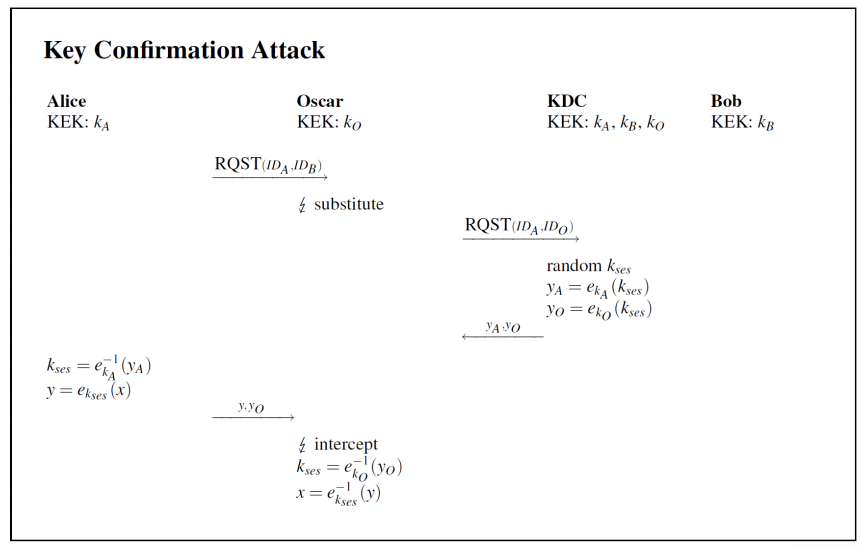
重播攻击一个可能的漏洞是重播攻击。这种攻击利用了一个事实，即爱丽丝和鲍勃都不知道他们收到的加密会话密钥是否实际上是新的。 如果一个旧的被重用，则会破坏密钥的新鲜度。 如果旧的会话密钥已损坏，这可能是一个特别严重的问题。 如果旧密钥（例如通过黑客）泄露，或者由于加密技术的进步，旧密钥使用的加密算法变得不安全，则可能发生这种情况。

如果Oscar掌握了先前的会话密钥，则他可以模拟KDC并将旧消息yA和y重新发送给Alice和Bob。 由于Oscar知道会话密钥，因此他可以解密将由Alice或Bob加密的纯文本。

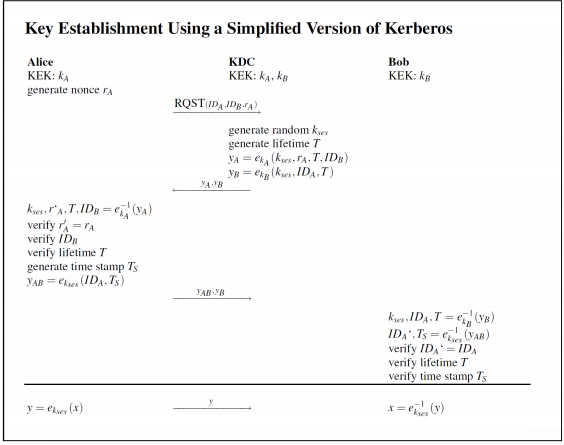
**1.中间人攻击：**

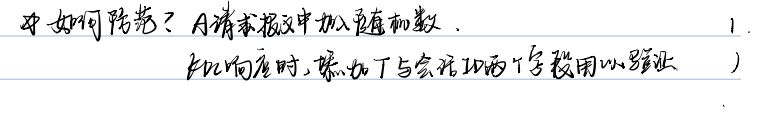
密钥确认攻击上述协议的另一个弱点是，爱丽丝不能保证她从KDC收到的密钥材料实际上是她和鲍勃之间进行的会话。 此攻击假设Oscar也是合法（但恶意）用户。 通过更改会话请求消息，Oscar可以欺骗KDC和Alice在他和Alice之间（而不是在Alice和Bob之间）建立会话。

（Alice请求与Bob建立会话，消息被Oscar拦截，其伪造请求发送到KDC，然后KDC发送加密后的密钥给Alice，Alice把加密后的信息以及会话密钥加密信息发送给O）



下面这个是加强版，***防止上面两种攻击***

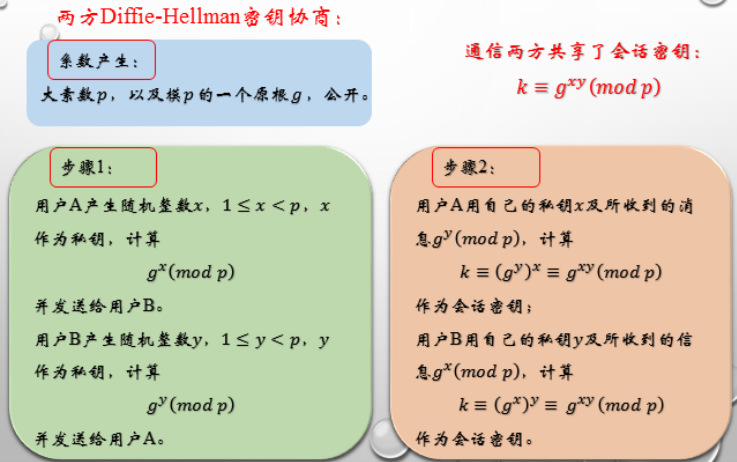


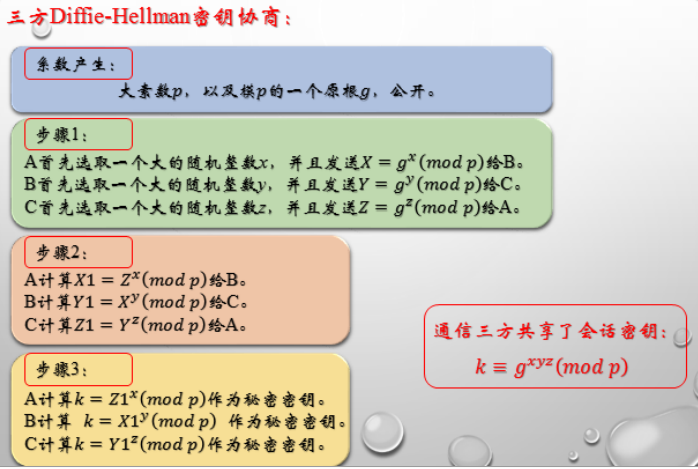
防范策略：

在A的request请求中加入会话建立双方的会话ID，并且彼此发送的消息有生命周期T,接收方在收到消息后要验证会话ID以及生命周期

并且A在向会话中心发送会话请求时，要加上一个随机数

**非对称方案**





**该方法同样也存在中间人攻击：**

比如中间人劫持了ga与gb，然后分别向A和B发送自己的x，从而中间人和A, B都建立了会话，只需要窃听每一条A到B的信息。