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。

2. 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。

2) 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。

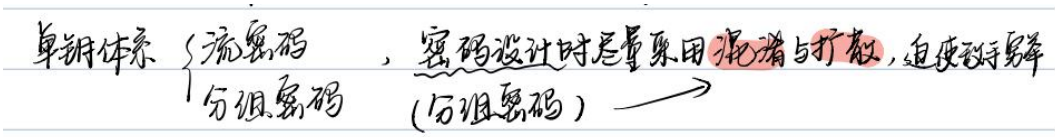
选择密文攻击：分析者可以任意选择密文，并可获得相应明文

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



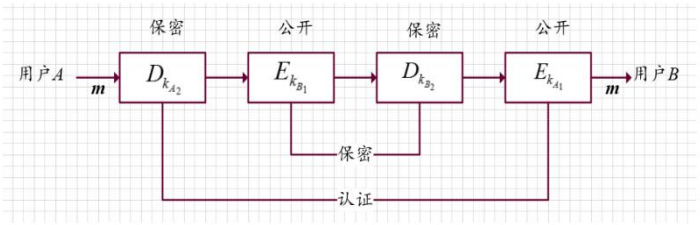
2） 双钥体制：加密密钥和解密密钥不同 Diffie 和 Hellman 1976 年首次提出，用户有公钥 k1，私钥 k2，公钥可以公开从而将加密和解密能力分开 安全性：可实现对 A 所发消息的验证

双钥认证体制：用户A以自己的秘密钥k_{A_2}对消息m进行A的专用变换$D_{k_{A_2}}$，A计算密文：$c = D_{k_{A_2}}(m)$送给用户B，B验证m：

$$m = E_{k_{A_1}}(c) = E_{k_{A_1}}(D_{k_{A_2}}(m)) \qquad (5)$$

双钥保密和认证体制

为了要同时实现保密性和确证性，要采用双重加、解密，如图5 所示：



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。

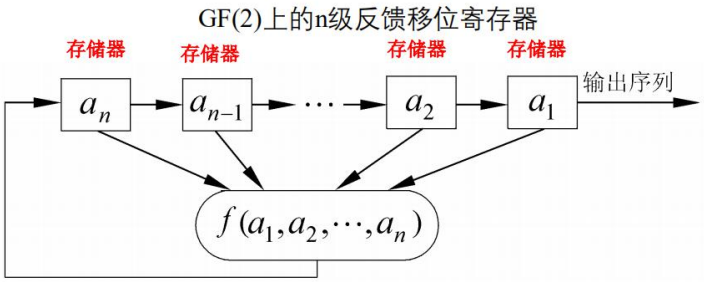
流密码强度完全依赖于密钥流生成器生成密钥流的随机性和不可预测性

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 $f(a_1,a_2,\ldots,a_n)$组成。



例 图3 是一个3级反馈移位寄存器，其初始状态为$(a_1,a_2,a_3)=(1,0,1)$，输出见右下表。

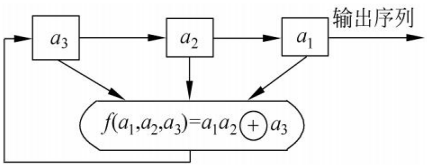


图 一个3级反馈移位寄存器

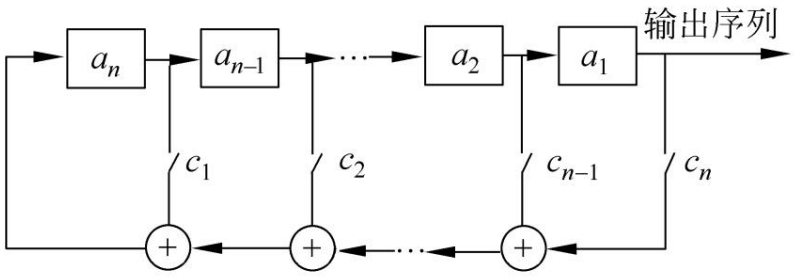
即输出序列为101110111011...，周期为4。

表 一个3级反馈移位寄存器的状态和输出

状态 (a_3,a_2,a_1)	输出
1 0 1	1
1 1 0	0
1 1 1	1
0 1 1	1
1 0 1	1
1 1 0	0

4. 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，$f(a_1,a_2,\ldots,a_n)= c_na_1 \oplus c_{n-1}a_2 \oplus \cdots \oplus c_1a_n$。



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 2^n 个。

n 级 LFSR 的状态周期： $\leq 2^n-1$。

输出序列的周期=状态周期，$\leq 2^n-1$。

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

LFSR 的特征多项式：$p(x)=1+c_1x+\ldots+c_{n-1}x_{n-1}+c_nx_n$

$$a_{n+k}=c_1a_{n+k-1}\oplus c_2a_{n+k-2}\oplus\cdots\oplus c_na_k$$
$$\Leftrightarrow a_{n+k}\oplus c_1a_{n+k-1}\oplus c_2a_{n+k-2}\oplus\cdots\oplus c_na_k=0$$

$D(a_k)=a_{k-1}$，用 $p(D)=1+c_1D+\cdots+c_nD^n$ 作用于 a_{n+k} 后恰好就是上式的左边，即

$$\begin{aligned}p(D)(a_{n+k}) &= (1+c_1D+\cdots+c_nD^n)(a_{n+k})\\ &= a_{n+k}+c_1D(a_{n+k})+\cdots+c_nD^n(a_{n+k})\\ &= a_{n+k}+c_1a_{n+k-1}+c_2a_{n+k-2}+\cdots+c_na_k\end{aligned}$$

5. 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{a_i}满足线性递推关系：$a_{h+n}=c_1a_{h+n-1}\oplus c_2a_{h+n-2}\oplus\cdots\oplus c_na_h$

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）

例2-6 设敌手得到密文串: 101101011110010

和相应的明文串: 011001111111001

相应的密钥流: 1101001000010 11

进一步假定敌手还知道密钥流是使用5级线性反馈移位寄存器产生的，那么敌手可分别用密文串中的前10个比特和明文串中的前10个比特建立如下方程

$$(a_6\ a_7\ a_8\ a_9\ a_{10})=(c_5\ c_4\ c_3\ c_2\ c_1)\begin{pmatrix}a_1&a_2&a_3&a_4&a_5\\a_2&a_3&a_4&a_5&a_6\\a_3&a_4&a_5&a_6&a_7\\a_4&a_5&a_6&a_7&a_8\\a_5&a_6&a_7&a_8&a_9\end{pmatrix}$$

即

$$(0\ 1\ 0\ 0\ 0)=(c_5\ c_4\ c_3\ c_2\ c_1)\begin{pmatrix}1&1&0&1&0\\1&0&1&0&0\\0&1&0&0&1\\1&0&0&1&0\\0&0&1&0&0\end{pmatrix}$$

而

$$\begin{pmatrix}1&1&0&1&0\\1&0&1&0&0\\0&1&0&0&1\\1&0&0&1&0\\0&0&1&0&0\end{pmatrix}^{-1}=\begin{pmatrix}0&1&0&0&1\\1&0&0&1&0\\0&0&0&0&1\\0&1&0&1&1\\1&0&1&1&0\end{pmatrix}$$

$$(c_5\ c_4\ c_3\ c_2\ c_1)=(0\ 1\ 0\ 0\ 0)\begin{pmatrix}0&1&0&0&1\\1&0&0&1&0\\0&0&0&0&1\\0&1&0&1&1\\1&0&1&1&0\end{pmatrix}$$

则：$(c_5\ c_4\ c_3\ c_2\ c_1)=(1\ 0\ 0\ 1\ 0)$

密钥流的递推关系为

$$a_{i+5}=c_5a_i\oplus c_2a_{i+3}=a_i\oplus a_{i+3}$$

三 分组密码

1. 基本概念

1) **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。

2) 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。

2. 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 2^n 加

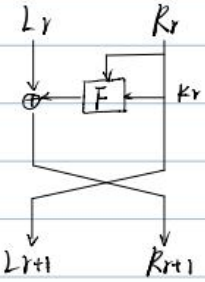
4.分组密码设计

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。

F-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、替换/置换网络（Feistel网络）



Feistel网络：实际上分组网络都采用

$$\begin{cases} L_{r+1} = R_r \\ R_{r+1} = L_r \oplus F(k_r, K_r) \end{cases}$$

Feistel 能否直接用作分组密码算法？

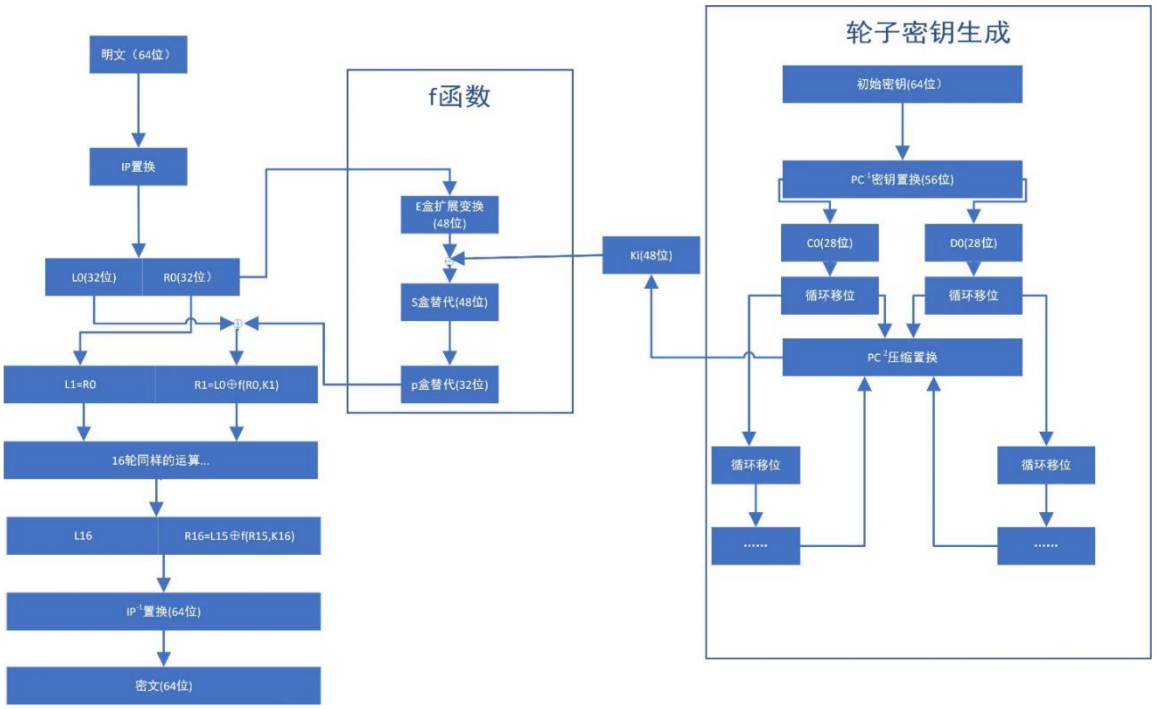
不能，首先明文(L_0, k_0)与密文(L_1, R_1)中，$L_1 = k_0$，即密钥只覆盖一半

再者，如果下函数设计不好，极其容易被破解

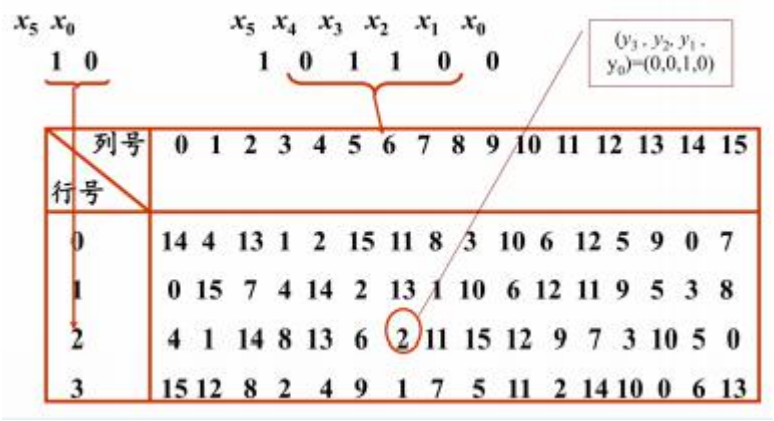
乘积密码：顺序地执行两个或多个密码系统，使得最后产生的密码强度大于每个基本系统的强度

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

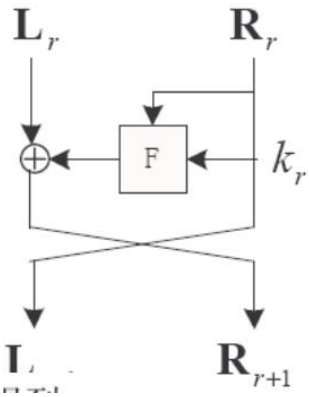
DES 算法原理：



DES:



三重 DES： 先加密后解密在加密



$$L_{r+1} = R_r$$

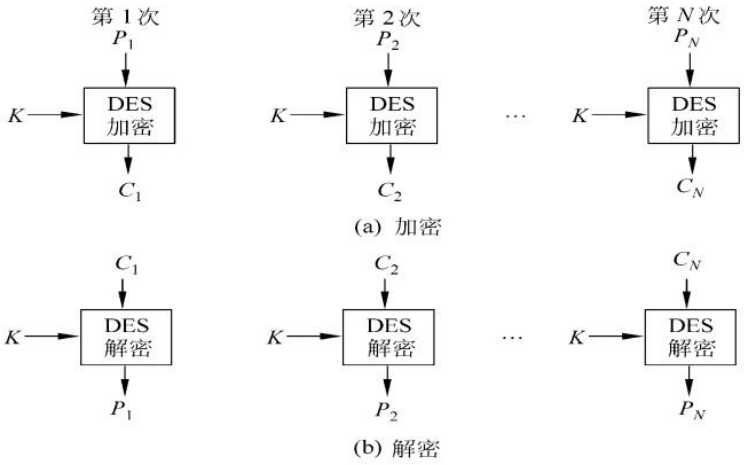
$$R_{r+1} = L_r \oplus F(k_r, R_r)$$

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

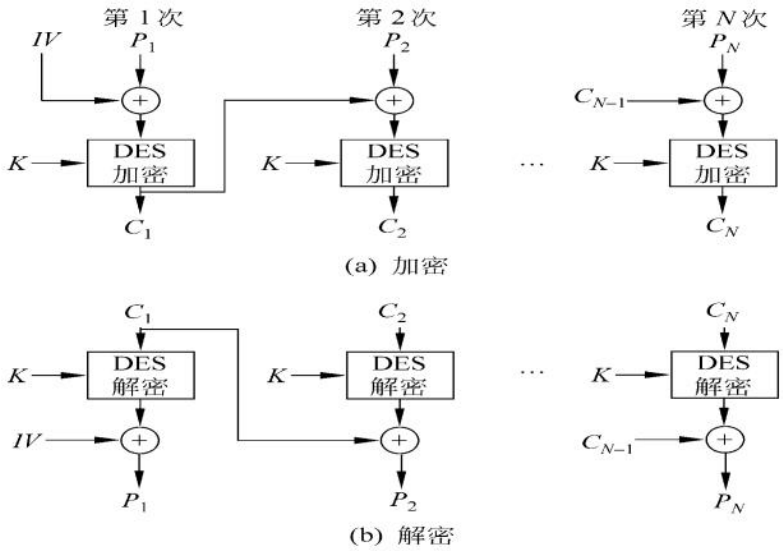
（**2**）不能隐蔽明文分组的统计规律和结构规律,不能抵抗替换攻击

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

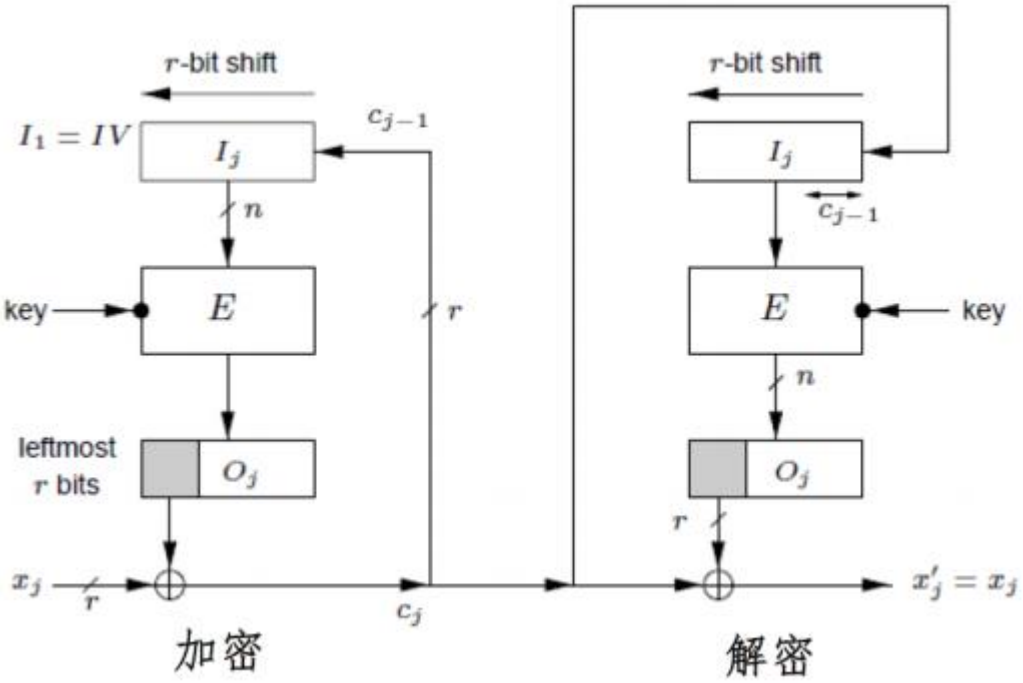


明文块的统计特性得到了隐蔽

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**

设攻击者Eve获得了两组明文/密文对

(m_1, c_1)，(m_2, c_2)。

如果Bob新截获了一个密文c，并发现：

- $c=c_1c_2 \pmod n$，则c所对应的明文一定是$m=m_1m_2 \pmod n$；
- $c=c_1/c_2 \pmod n$，则c所对应的明文一定是$m=m_1/m_2 \pmod n$；
- $c=c_2/c_1 \pmod n$，则c所对应的明文一定是$m=m_2/m_1 \pmod n$。

（模除运算$\cdot/\cdot \pmod n$是一个数论运算）

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

$2^{511}<p<2^{512}$

这样要穷举 2^{510} 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：

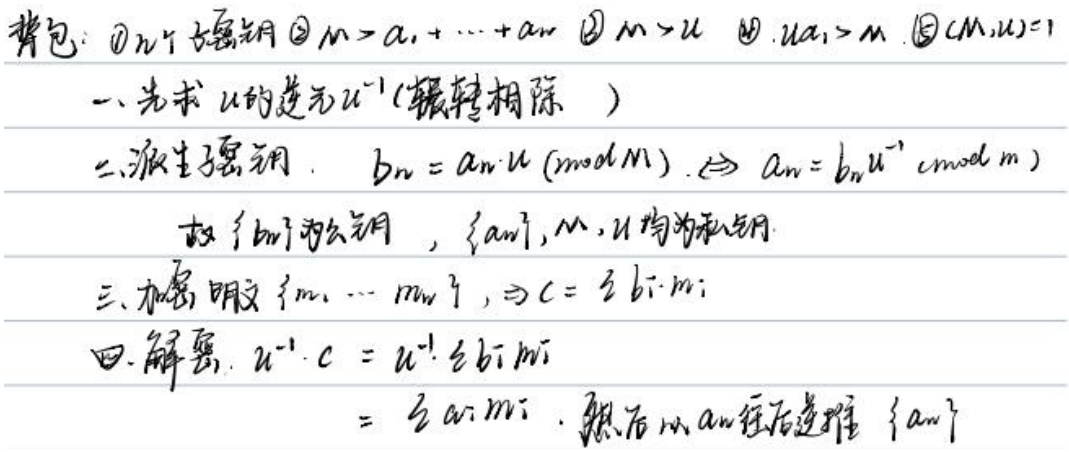
$C=U^{-1}c(\text{mod } M)$

$=U^{-1}(m1b1+m2b2+m3b3+\ldots+mnbn)(\text{mod } M)$

$=m1a1+m2a2+m3a3+\ldots+mnan(\text{mod } M)$

$=m1a1+m2a2+m3a3+\ldots+mnan$。

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：$c=m^2 \pmod n$，c 是密文 B 在收到密文后，用私钥解密：

$$m_p = c^{\frac{p+1}{4}} \pmod p;$$

$$m_q = c^{\frac{p+1}{4}} \pmod q.$$

计算四个数 m(1,1),m(1,2),m(2,1),m(2,2)满足：

$0<m(1,1)<n$； $0<m(1,2)<n$；

$0<m(2,1)<n$； $0<m(2,2)<n$；

$m(1,1)(\mathrm{mod}\ p)=m_p$； $m(1,1)(\mathrm{mod}\ q)=m_q$；

$m(1,2)(\mathrm{mod}\ p)=m_p$； $m(1,2)(\mathrm{mod}\ q)=q-m_q$；

$m(2,1)(\mathrm{mod}\ p)=p-m_p$； $m(2,1)(\mathrm{mod}\ q)=m_q$；

$m(2,2)(\mathrm{mod}\ p)=p-m_p$； $m(2,2)(\mathrm{mod}\ q)=q-m_q$

Rabin的解密原理

因为$n=pq$是两个不同的素数的乘积，所以，关于未知数x的二次方程

$$c=x^2(\mathrm{mod}\ n)$$

恰好有4个不同的根x，分别有以下形状：

一个根的$(\mathrm{mod}\ p)$、$(\mathrm{mod}\ q)$值是m_p、m_q；

一个根的$(\mathrm{mod}\ p)$、$(\mathrm{mod}\ q)$值是m_p、$q-m_q$；

一个根的$(\mathrm{mod}\ p)$、$(\mathrm{mod}\ q)$值是$p-m_p$、m_q；

一个根的$(\mathrm{mod}\ p)$、$(\mathrm{mod}\ q)$值是$p-m_p$、$q-m_q$。

Legendre 的符号表示：

若$\left(\frac{n}{p}\right)=1$，表示 n 是模 p 的平方剩余，二次同余式此时有解

$p\equiv 3(mod\ 4)$时，$\pm n^{\frac{p+1}{4}}$是此二次同余式的解

Rabin的解密原理

设p,q是两个形如$4k+3$的不同素数，如果整数a满足$\left(\frac{a}{p}\right)=1,\left(\frac{a}{q}\right)=1$,则同余方程

$$x^2 \equiv a\ (mod\ pq)$$等价于

$$\begin{cases} x^2 \equiv a\ (mod\ p) \\ x^2 \equiv a\ (mod\ q) \end{cases}$$

而由上述定理可知，$x^2 \equiv a\ (mod\ p)$的解是$\pm a^{\frac{p+1}{4}}$，

$x^2 \equiv a\ (mod\ q)$的解是$\pm a^{\frac{q+1}{4}}$

Rabin的解密原理

由中国剩余定理可得同余方程$x^2 \equiv a\ (mod\ pq)$的解是

$$x \equiv \pm\left(a^{\frac{p+1}{4}}(mod\ p)\right)uq \times$$

$$\pm\left(a^{\frac{q+1}{4}}(mod\ q)\right)vp\ (mod\ pq)$$

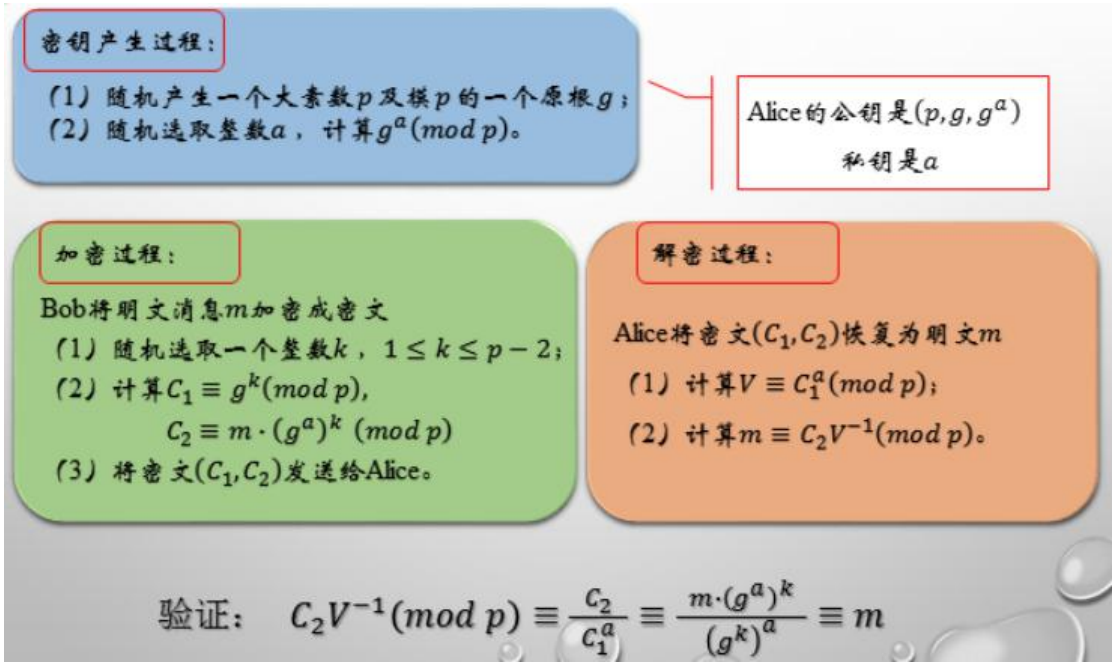
其中u,v满足$uq \equiv 1(mod\ p), vp \equiv 1(mod\ q)$

求 n 的分解式 n=pq 是大数分解问题。

RSA与Rabin比较

比较项目	RSA	Rabin
公钥	(n, e)	n
私钥	d	(p, q)
加密算法	$c=m^e(\text{mod } n)$	$c=m^2(\text{mod } n)$
解密算法	$m=c^d(\text{mod } n)$	若干步
安全基础	大数分解问题的困难性	大数分解问题的困难性

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

至少两个人生日相同的概率，可以先算出所有人生日互不相同的概率，再用 1 减去这个概率。

我们把这个问题设想成，每个人排队依次进入一个房间。第一个进入房间的人，与房间里已有的人（0人），生日都不相同的概率是 365/365；第二个进入房间的人，生日独一无二的概率是 364/365；第三个人是 363/365，以此类推。

因此，所有人的生日都不相同的概率，就是下面的公式。

$$\bar{p}(n) = 1 \cdot \left(1 - \frac{1}{365}\right) \cdot \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right)$$

上面公式的 n 表示进入房间的人数。可以看出，进入房间的人越多，生日互不相同的概率就越小。

这个公式可以推导成下面的形式。

$$\frac{365!}{365^n (365-n)!}$$

那么，至少有两个人生日相同的概率，就是 1 减去上面的公式。

$$p(n) = 1 - \bar{p}(n) = 1 - \frac{365!}{365^n (365-n)!}$$

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 $s=h^d \pmod n$ 。

（3） Alice 将(m，s)发送给 Bob。

（4） Bob 用 Alice 的公钥 e，检验是否 $H(m)= s^e \pmod n$。

若是则(m，s)是 Alice 发送的签名消息。

- ElGamal 数字签名

（1） Alice用H将消息m进行处理，得$h=H(m)$。

（2） Alice选择秘密随机数k，满足

$$0<k<p-1，且(k, p-1)=1，$$

计算

$$r=g^k \pmod p;$$

$$s=(h-xr)k^{-1} \pmod{(p-1)}。$$

（3） Alice将$(m，r，s)$发送给Bob。

（4） Bob用Alice的公钥，检验是否

$$y^r r^s=g^{H(m)} \pmod p。$$

若是则$(m，r，s)$是Alice发送的签名消息。

- Schnorr 数字签名（看一看）

Schnorr数字签名

Alice拥有3个正整数$(p，q，g)$，向自己的通信伙伴公开。其中：

- p是模数（即将要进行$(\text{mod } p)$模数运算），它是一个素数，值的范围在2^{511}到2^{512}之间（即p是一个长度为512的比特串）。
- q也是模数（即将要进行$(\text{mod } q)$模数运算），它是一个素数，$2^{159}<q$（即q是一个长度不小于160的比特串），并且q是$p-1$的一个因子。
- g是域$GF(p)$的元素，且$g^q=1 \pmod p$。

Alice选择x，其中$1<x<q$。

Alice计算$y=g^x(\mathrm{mod}\ p)$。

Alice的公钥是(p, q, g, y)，Alice的私钥是x。

（1）Alice选择秘密随机数k，满足$0<k<q$，计算

$$r=g^k(\mathrm{mod}\ p);$$

$$e=H(r, m);$$

$$s=k+xe(\mathrm{mod}\ q)。$$

（3）Alice将(m, e, s)发送给Bob。

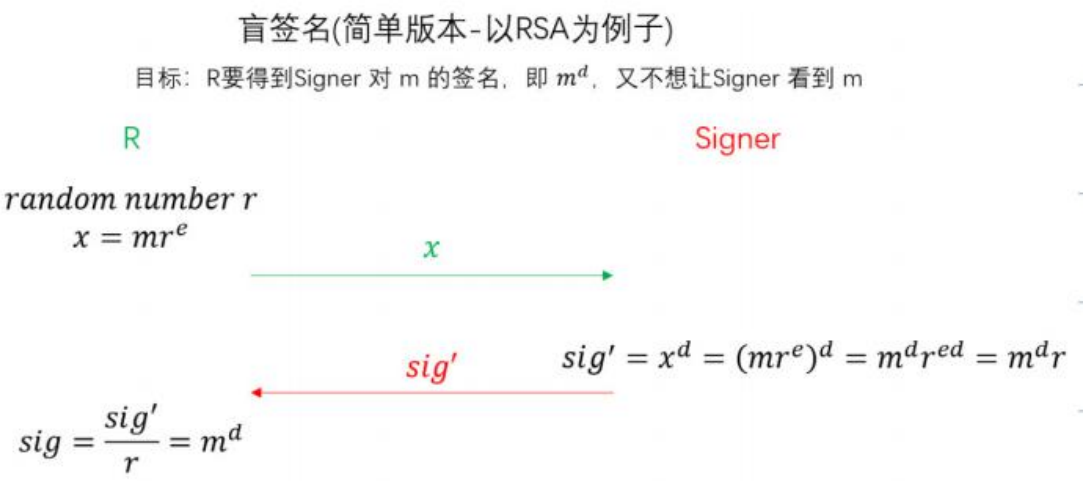
（4）Bob用Alice的公钥，计算$r'=g^s y^{-e}(\mathrm{mod}\ p)$。检验是否

$$e=H(r', m)。$$

若是则(m, e, s)是Alice发送的签名消息。

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。

习题 设：p=17；n=5； t=3；
($ID(1)$， $h(ID(1))$=(1， 8)；
($ID(2)$， $h(ID(2))$=(2， 7)；
($ID(3)$， $h(ID(3))$=(3， 10)；
($ID(4)$， $h(ID(4))$=(4， 0)；
($ID(5)$， $h(ID(5))$=(5， 11)。
当第1位~第3位参与者同时到场，求共享的秘密

$$h(x) = a_0 + a_1 x + a_2 x^2 (\mathrm{mod} 17)$$

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

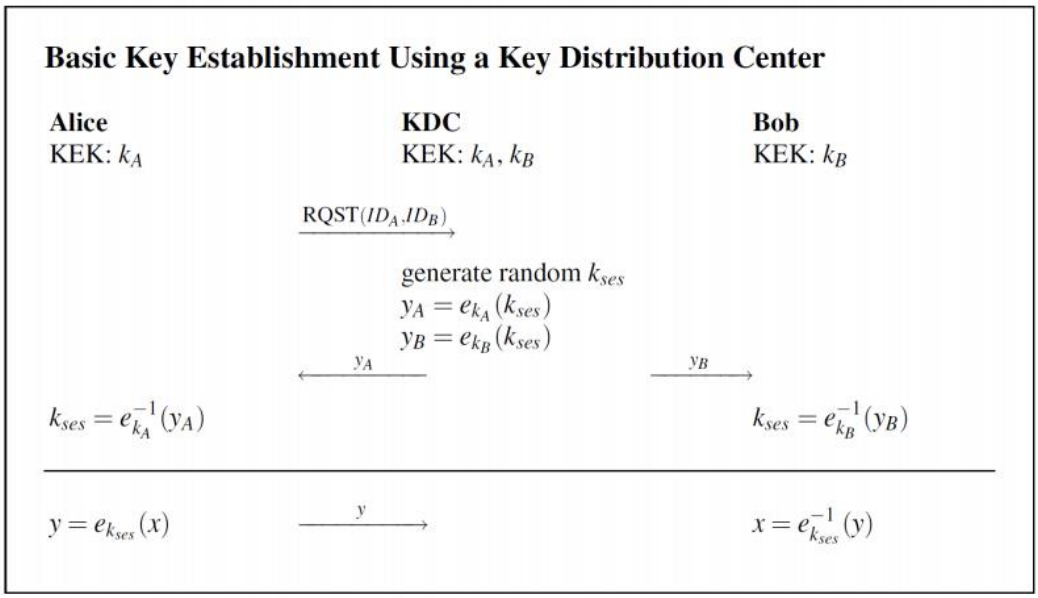
The n^2 Key Distribution Problem

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）≈n^2 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

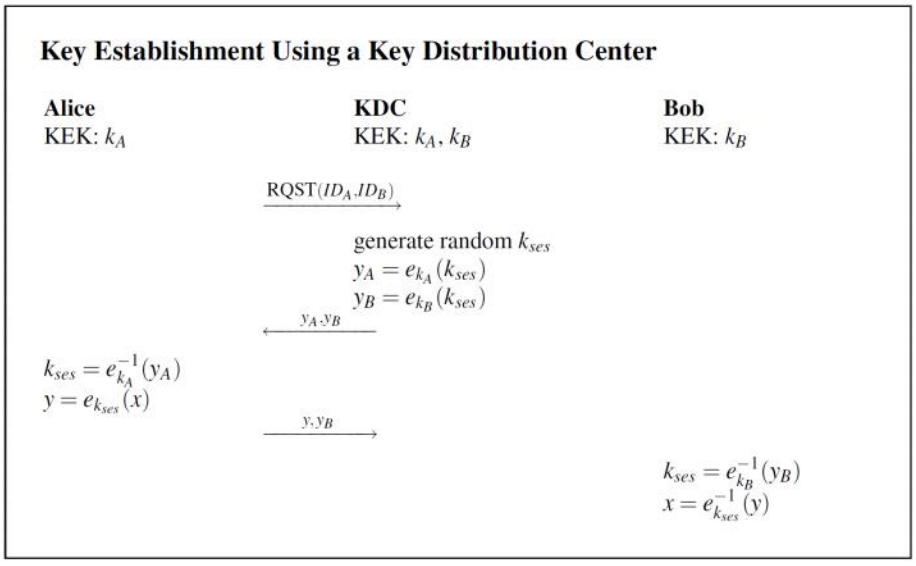
分放双方方案

Key Establishment Using Symmetric-Key Techniques



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：

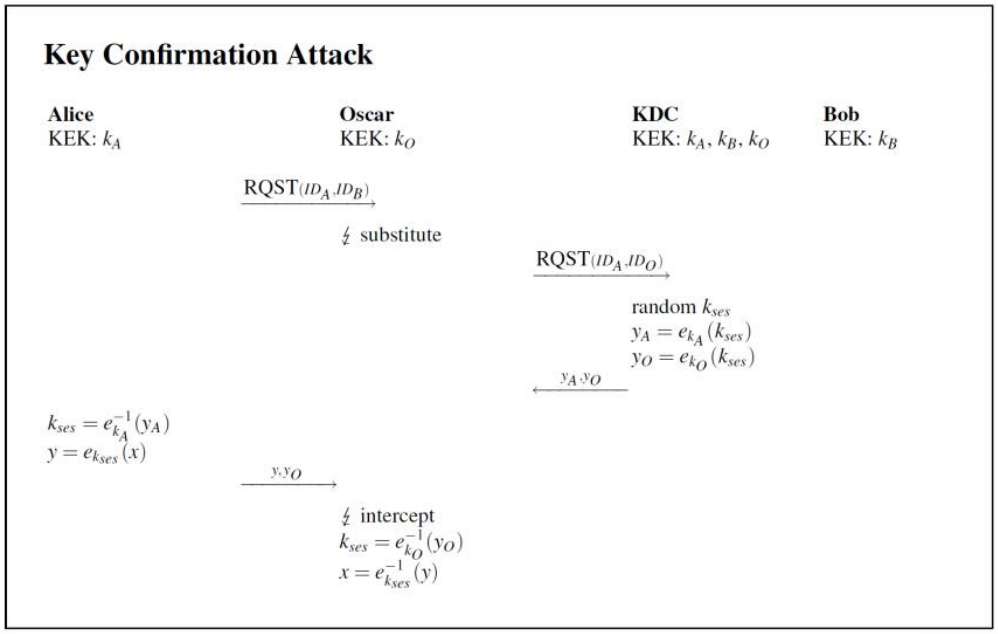
重播攻击一个可能的漏洞是重播攻击。这种攻击利用了一个事实，即爱丽丝和鲍勃都不知道他们收到的加密会话密钥是否实际上是新的。 如果一个旧的被重用，则会破坏密钥的新鲜度。 如果旧的会话密钥已损坏，这可能是一个特别严重的问题。 如果旧密钥（例如通过黑客）泄露，或者由于加密技术的进步，旧密钥使用的加密算法变得不安全，则可能发生这种情况。

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。

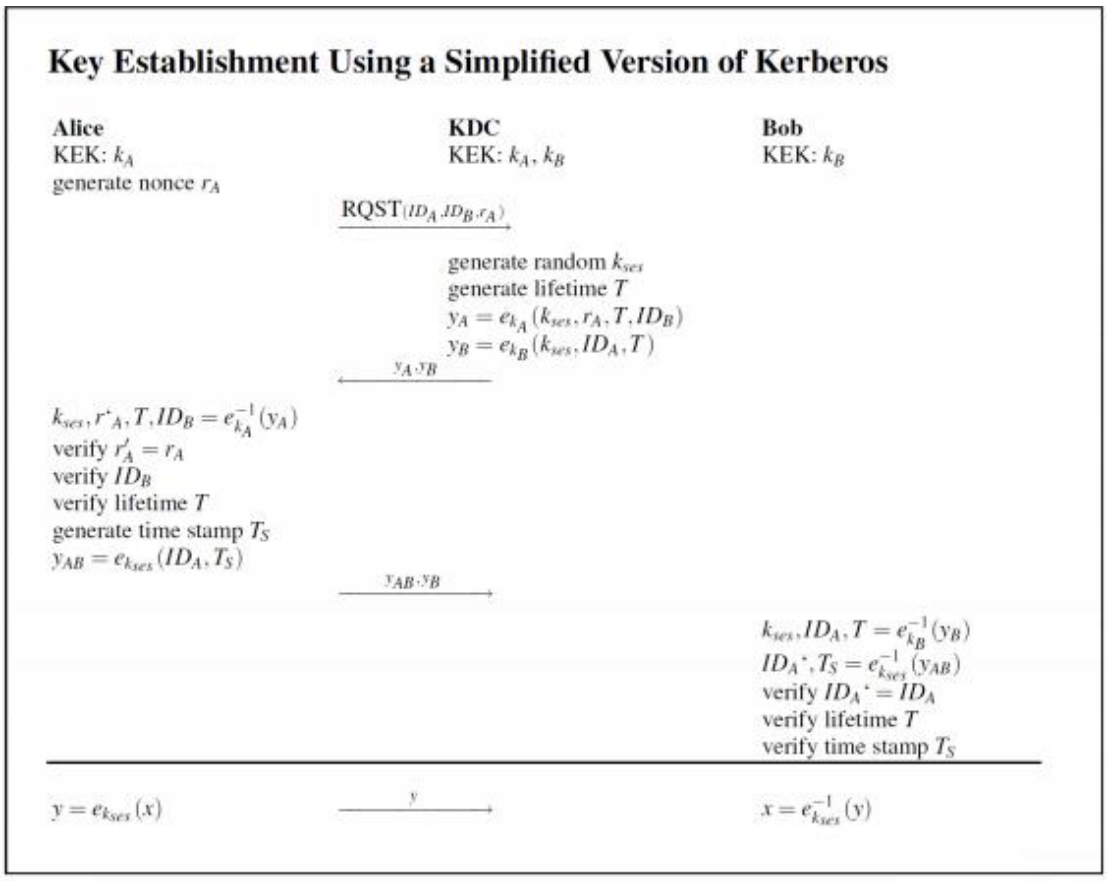
1.中间人攻击：

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。

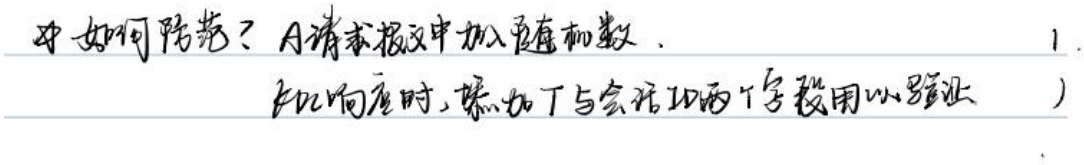
（Alice 请求与 Bob 建立会话，消息被 Oscar 拦截，其伪造请求发送到 KDC，然后 KDC 发送加密后的密钥给 Alice，Alice 把加密后的信息以及会话密钥加密信息发送给 O）



下面这个是加强版，***防止上面两种攻击***



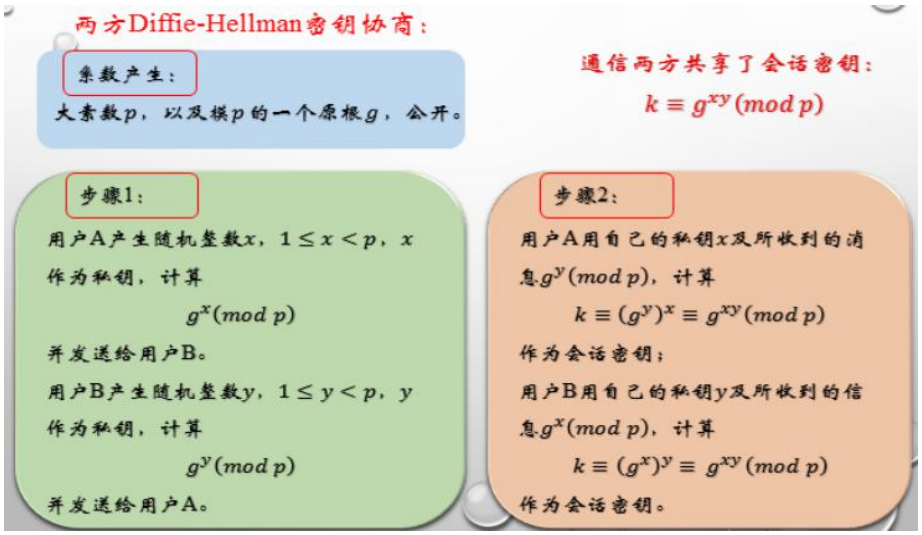
防范策略：

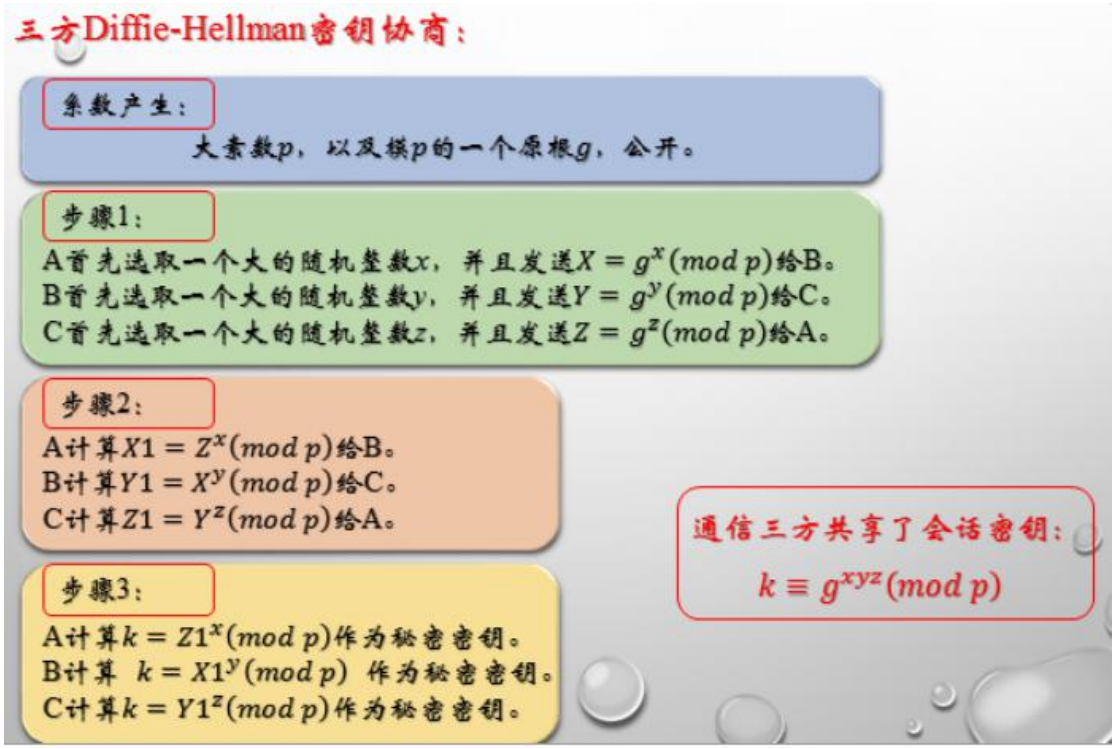


在 A 的 request 请求中加入会话建立双方的会话 ID，并且彼此发送的消息有生命周期 T,接收方在收到消息后要验证会话 ID 以及生命周期

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 g^a 与 g^b，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。