前言

包含了绝大部分的ppt1-13内容，即考试最重要的内容。
数论的各种证明（虽然不是绝对完全），绝对能看懂。
建议：

1. 至少完成一次顺序阅读。非常希望您是在学习期间而非复习期间阅读（以免我这种上课没听懂挂科的）
2. 数论部分并不难（至少考试的那些）。
3. 配合同项目中学长的《密码学不过你来找我》。
4. 过早的考试题已经没有意义。
5. 本文档由obsidian工具书写，不完全兼容正常markdown（大概率没事）

ppt-1

安全攻击

对任何机构的信息资源进行破坏的行为。
被动攻击：

1. 消息内容的泄露：观看网络信息
2. 流量分析：观看网络信息，但是只能分析其模式，没有具体内容
   主动攻击：
3. 伪装：扮演通信的一方
4. 重放：不直接影响正常链路的通讯，而是在其上重新发送这些讯息。造成重复，但是相位有区别的两组相同信息
5. 修改：截取信息，修改，发给另一方
6. 拒绝服务：任何方式毙了服务器的服务

安全服务

5类，14个特定服务

1. 认证：确保通信实体是合法真实的。
   1. 同等实体认证：身份认证，确保通信对端身份是预定的对端
   2. 数据源认证：确保数据来源于指定实体，并且未被篡改（因而常与数据完整性认证结合）
2. 访问控制：防止资源非授权使用
3. 数据保密性：保密和避免流量分析

4. 数据完整性：确保数据是未被修改的
5. 不可否认性：防止通信方对通信的否认，包括源和宿的两个不可否认性

设计安全服务的内容

设计一个算法，执行变换，该变换无法被攻破或代价过高
产生算法所使用的秘密信息
设计分配和共享秘密信息的方法
指定通信协议，使用安全算法和秘密信息实现安全服务

安全机制

用于检测，防止安全攻击。或从攻击中恢复的机制
单一机制有其极限，不能覆盖所有情况。

最重要之一，也是我们集中的，是密码编码机制。

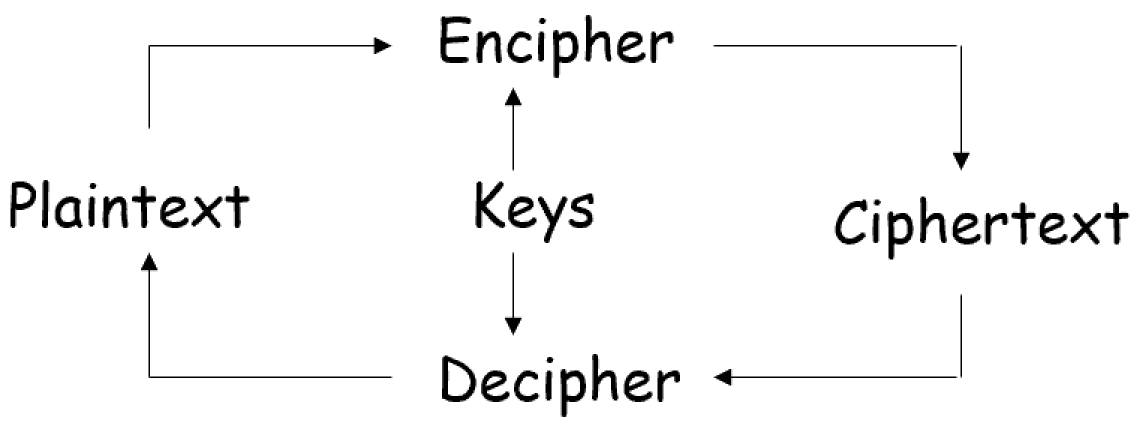
特定安全机制：

1. 加密
2. 数字签名
3. 访问控制
4. 数据完整性
5. 认证交换
6. 通信填充
7. 路由控制
8. 公证
   普遍安全机制
9. 可信功能
10. 安全标示
11. 事件检测
12. 安全审计追踪
13. 安全恢复

ppt-2

课程第一部分——对称密码

简单加密系统模型



参数/秘钥Key：K
加密变换E
令明文/消息m，用变换Ek得密文C

传统保密通信机制：

理论安全和实际安全

理论安全：
绝对安全，即使密文全部送给攻击者，也没有任何可能唯一确定明文。
秘钥长度须大于等于明文长度，并且一次一密。
实际安全：
用无限资源理论可解。但如果资源有限则无法破解，就是**计算上不可行**

什么是加密

密码体制

加密系统的基本工作方式。
基本要素是**密码算法和秘钥**

有几种（组）密码体制，互相关联：

1. 对称密码体制：加解密秘钥相同，能加就能解。开放性差。可以使用序列密码（如AES）或

分组密码（如RC4）
2. 非对称密码体制：加解密秘钥不同，互相推导计算不可行。开放性好。通常与对称密码体制（加密消息）结合，加密秘钥。
3. 序列密码体制：密文与算法，秘钥，明文位置有关。

   令明文序列

   $$p = p_{n-1} \dots p_1 p_0$$

   密钥序列

   $$k = k_{n-1} \dots k_1 k_0$$

   密文序列

   $$c = c_{n-1} \dots c_1 c_0 = E_{k_{n-1}}(p_{n-1}) \dots E_{k_1}(p_1) E_{k_0}(p_0)$$

   若

   $$c_i = E_{k_i}(p_i) = p_i \oplus k_i$$

   则称此类为加法序列密码。

   序列密码一般分为：

   - **同步序列密码**：密钥序列的产生需要收发双方进行同步，密钥序列的产生完全独立于明文消息和密文消息
   - **自同步序列密码**：密钥序列的产生是密钥及固定大小的以往密文位的函数

   ### 1.2 基本原理

   序列密码的加解密只是简单的**模二加运算**，其安全强度主要依赖于**密钥序列的随机性**。密钥序列产生器（KG，Keystram Generator）的要求如下：
4. 分组密码：仅与给定的密码算法和秘钥有关，与明文位置无关。通常是64位数据块转64位密文块
5. 确定型和概率型：明文和秘钥确定后，密文确定或不确定。
6. 单向函数型和双向变换型：明文到密文是否可逆

现代密码学原则

假定密码算法可以公开，秘钥需要保密。

对称密码系统

五部分：明文，加密算法，秘钥，密文，解密算法。
两点：

1. 无法仅根据密文破译
2. 双方以安全形式获得秘钥并保证秘钥安全
   同样，算法需要公开。

密码学

密码编码学

三类：

1. 根据明文转密文的操作：代换和置换
2. 根据秘钥数量和方式：对称密码体制（单钥，秘密密钥）和非对称密码体制（双钥，公开密钥）
3. 根据明文处理方式：分组加密和流加密

密码分析学

第一类：密码分析：尝试获得明文或秘钥。
方法如下，越来越难防：

1. 唯密文攻击：只有密文和算法，基本是穷举
2. 已知明文攻击：已知一些明密对和算法，试图破译另外的一些密文
3. 选择明文攻击：选择的明文能够知道匹配的密文，破译其他密文
4. 选择密文攻击：选择的密文能知道匹配的明文，破译其他密文
5. 选择文本攻击：选明和选密都能做到，破译其他密文
   第二类：穷举攻击：尝试所有可能的秘钥，期望是一半的秘钥被测试过。

代换技术

以下是序列密码（对应下节的分组密码^2aff3d）内容
改变内容的表现形式，保持元素之间相对位置不变。

凯撒密码

明文字母表 P = {p0, p1, …, pn-1}
密文字母表 C = {c0, c1, …, cn-1}
两个表视作环，n位，秘钥为正整数k
i位为原文，j位为密文：

加密：$i+k \equiv j \pmod n$
解密：$j-k \equiv i \pmod n$

破解：穷举法高光时刻。

单表代换

每个字母能映射到任何一种密文字母。映射终点不能重复。

```
这样密钥有26个字母长
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

由于语言字母使用频率不同，根据密文中的频率，能猜出一部分字母/双字母/三字母。
这使得它难以抵御穷举

一次一密

秘钥与消息一样长。秘钥只加密一个信息，用后即弃。不可攻破。
运算基于二进制而非字符。$c_i = p_i \oplus k_i$，$p_i = c_i \oplus k_i$。p为明文位，c为密文位。

然而，产生大规模随机密钥有实际困难，密钥的分配和保护无法保证。

置换技术

和单表代换大差不差吧。频率问题。

- 一维变换—矩阵转置

输入 →

输出 ↓

C	A	N	Y
O	U	U	N
D	E	R	S
T	A	N	D

明文：can you understand

密文：codtaueanurnynsd

- 二维变换—图形转置

```
        D        密文
      T A N
    N D E R S
  C A N Y O U U
明文 →
```

密文：dnsuaruteodynnac

明文：can you understand

栅栏技术

- 按照对角线的顺序写出明文，按行的顺序读出作为密文
- 如加密 meet me after the toga party:

  m e m a t r h t g p r y

  e t e f e t e o a a t

- 可以得到密文

  MEMATRHTGPRYETEFETEOAAT

没什么好说的。

ppt-3

分组密码

^2aff3d

加密解密体制，把明文作为整体，输出一个等长密文。

常用的Feistel结构：由轮函数组成。
扩展秘钥使得每一轮使用不同的子密钥。
每一轮用输入的右半部分作为下一轮的左半部分。
秘钥代换右半的数据，然后它和原有左半异或，作为下一轮右半部分。
典型的是分组密码DES，64位分组，56位秘钥

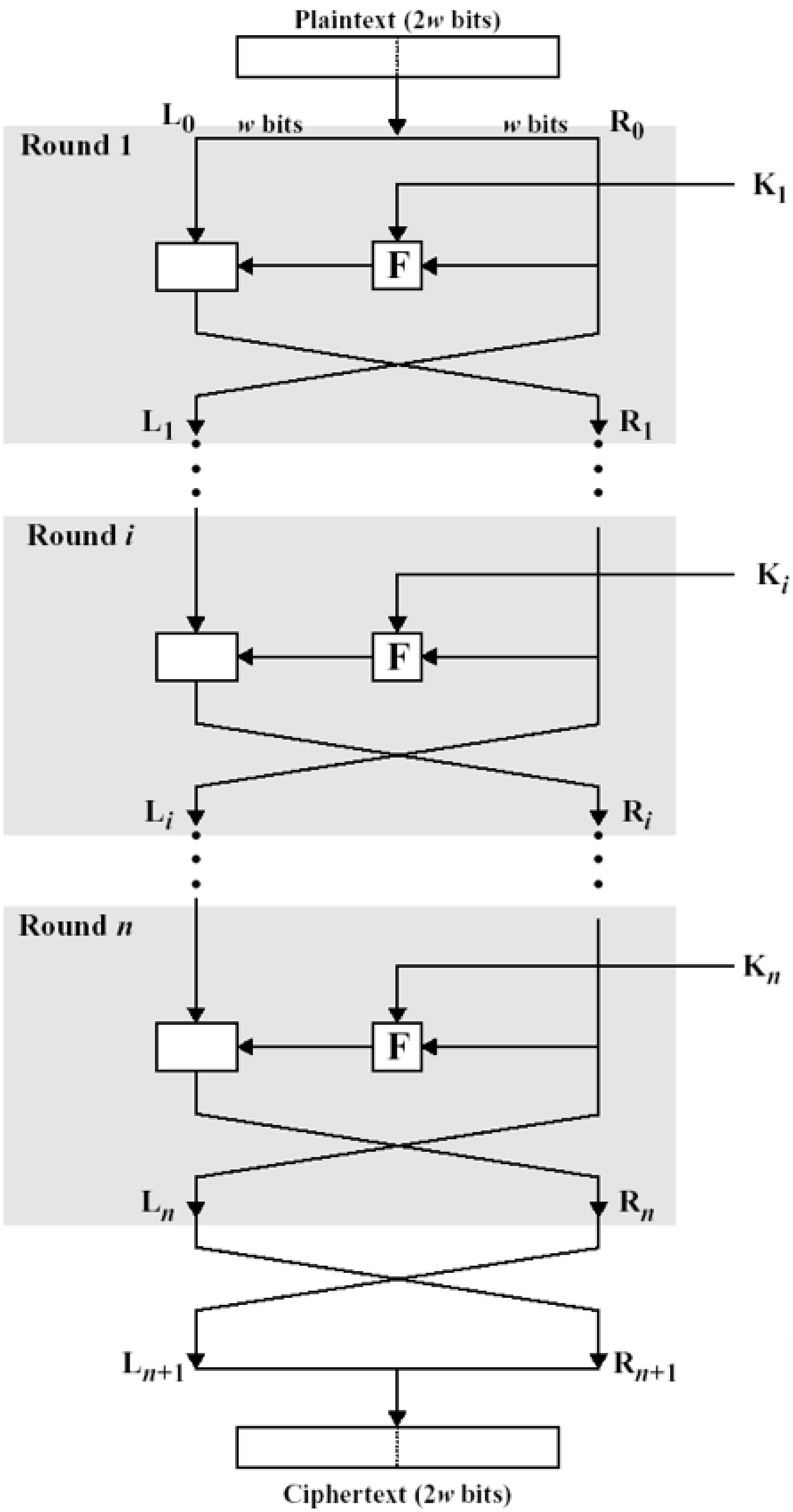


Figure 3.5 Classical Feistel Network

乘积思想：为了应对统计分析和破译，需要对明文作混淆和扩散处理。方法泄露与否不影响提高多少破译难度。

2^n个明文分组要一一对应唯一密文分组。这个变换称作可逆的和非奇异的

DES

https://blog.csdn.net/Demonslzh/article/details/129129493

基于流的。按比特处理。

首先，把64位明文进行初始置换。如第50位放在第二位。

(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

然后，进入轮函数。

- 将明文分成左右两部分

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \text{ xor } F(R_{i-1}, K_i)$$

右半部分作为下一轮的左半部分。

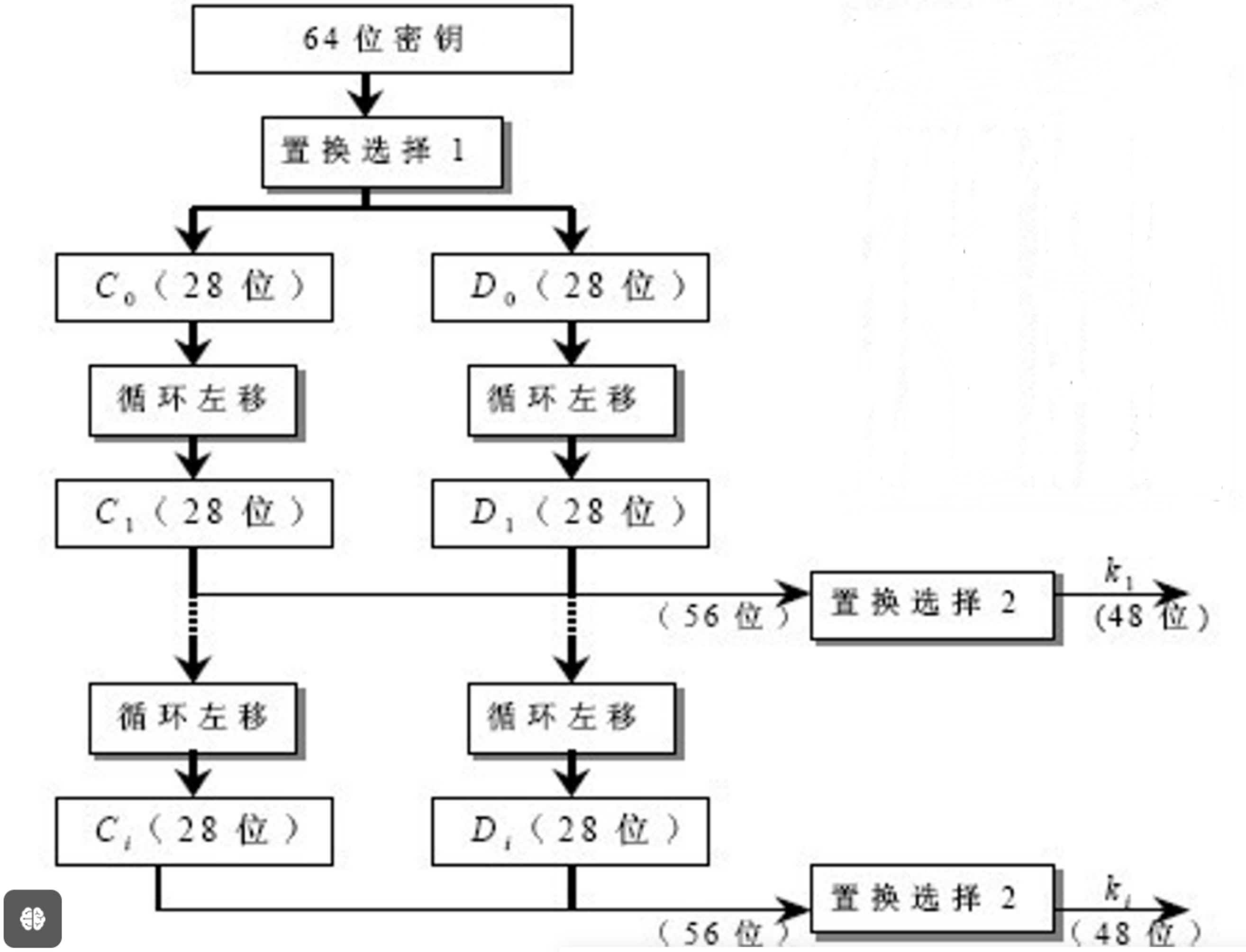
左半部分（32位）第一步被扩展为48位。

(c) Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

原有32位复制一份放在第一位，原有第一位现在是新的的第二位和最后一位。

DES将64位主密钥转为16个48位子密钥，用于每轮的运算。



置换选择1（pc-1）

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

置换出现在轮函数之前。

舍弃每八位数据的第八位（奇偶校验位），重排每一位。

循环左移：第1到第16轮，若i为1,2,9,16，两块左移两位。否则左移一位。

置换选择2（pc-2）

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

56转48，生成当前轮次的子密钥。

第二步，和48比特**子秘钥**做异或。

第三步，48位的结果送给8个S盒获得32位。
48位分为8个6位，每个6位送给一个s盒，输出4位。这个过程和十进制有关。

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

以S1为例

6位的第一位，第六位组合为行，剩下的四位组合为列。

对于101010输入，行10->2, 列0101->5, 输出0110->6。六位输入，4位输出

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

第四步，p盒替换

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

上述8个s盒的32位输出转换为32位新输出，即为轮函数结果。**作为下一轮的右半部分。**

逆置换

在最后16次结束后，进行一次64转64的逆置换。

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

得到最终密文

ppt-4

死去的线性代数知识开始攻击我

群

群G拥有集合G和二元运算。集合G中任意两个元素组成序偶(a,b)进行二元运算得到新元素。满足：

1. 封闭性：运算结果必然属于G
2. 结合律：`(a*b)*c = a*(b*c)`
3. 单位元：G存在元素e，任何元素a，有`a*e = e*a = a`
4. 逆元：G存在元素t，任何元素a，有`t*a = a*t = e`
   例子：乘法运算群`{1,-1,*}`

N集合有n个符号，从N到N的所有可能的映射构成群S。二元运算为映射结果的二次映射。显然，S为群

有限群：元素有限，阶为元素个数。
无限群：不是有限群

交换群（阿贝尔群）满足第五个要求：
5. 交换律：G中任意元素a,b，满足`a*b = b*a`
循环群：群中每一个元素都是固定元素a的幂a^k（k为整数）。a生成了G，a为生成元。

环

环R，有集合R和加法，乘法两个固定的运算。注意，加法，乘法的运算定义是可变的，如同重载运算符。这里是抽象代数。
满足：

1. 群的5条性质（加法运算）。单位元为0，a的逆为-a
2. (M1), 乘法封闭性, 如果a和b属于R, 则ab也属于R
3. (M2), 乘法结合律,对于R中任意a, b, c有a(bc)=(ab)c.
4. (M3), 乘法分配律, a(b+c)=ab+ac or (a+b)c=ac+bc
5. (M4), 乘法交换律, ab=ba，交换环
6. (M5), 乘法单位元, R中存在元素1使得所有a有 a1=1a.
7. (M6), 无零因子, 如果R中有a, b且ab=0, 则 a=0 or b=0.

域

整环。额外满足：
8. (M7), 乘法逆元。对于F中的任意元素a(除0以外), F中都存在一个元素a-1, 使得aa-1=(a-1)a=1.
9. 域就是一个集合，在其上进行加减乘除而不脱离该集合, 除法按以下规则定义: a/b=a(b-1).
有理数集合, 实数集合和复数集合都是域；整数集合不是域，因为除了1和-1有乘法逆元，其他元素都无乘法逆元

同余

a=mb, 全整数，整除写作b|a。有以下性质

1. a|1，a=正负1
2. a|b，b|a，a=正负b
3. 非零整数能整除0：b|0
4. b|g，且b|h, 则对任何整数m和n有b|(mg+nh)

同余：整数a，b，n不等于0，a与b在模n时同余等价于n|(a-b)，记作a≡b mod n

- a≡b mod n 隐含b≡a mod n
- a≡b mod n 和b≡c mod n 隐含a≡c mod n

模的术数运算

模的计算方法：x mod y = x - (x/y下取整) 乘 y

反身性：a=a mod n
对称性：若a=b mod n，则b=a mod n
传递性: 若a=b mod n 且b=c mod n，则a=c mod n

如果 a=b mod n且 c=d mod n，则
a+c=(b+d) mod n
a-c=(b-d) mod n
a•c=(b•d) mod n
(a^b) % p = ((a % p)^b) % p

运算为op，有`(a1 op a2) mod n =[(a1 mod n ) op (a2 mod n)] mod n`

证明

任何模运算的证明，谨记将a,b表示为`i*n + ra, j*n + rb`，代入原式，如mod n，消去n的倍数的项。

欧几里得算法

求最大公约数。
算法：对于任何非负的整数a和n，gcd(n,a)=gcd(a,n mod a)。

过程 ¶

如果我们已知两个数 a 和 b，如何求出二者的最大公约数呢?

不妨设 $a > b$。

我们发现如果 b 是 a 的约数，那么 b 就是二者的最大公约数。 下面讨论不能整除的情况，即 $a = b \times q + r$，其中 $r < b$。

我们通过证明可以得到 $\gcd(a, b) = \gcd(b, a \bmod b)$，过程如下：

证明

设 $a = bk + c$，显然有 $c = a \bmod b$。设 $d \mid a$, $d \mid b$，则 $c = a - bk$, $\frac{c}{d} = \frac{a}{d} - \frac{b}{d}k$。

由右边的式子可知 $\frac{c}{d}$ 为整数，即 $d \mid c$，所以对于 a, b 的公约数，它也会是 $b, a \bmod b$ 的公约数。

反过来也需要证明：

设 $d \mid b$, $d \mid (a \bmod b)$，我们还是可以像之前一样得到以下式子 $\frac{a \bmod b}{d} = \frac{a}{d} - \frac{b}{d}k$, $\frac{a \bmod b}{d} + \frac{b}{d}k = \frac{a}{d}$。

因为左边式子显然为整数，所以 $\frac{a}{d}$ 也为整数，即 $d \mid a$，所以 $b, a \bmod b$ 的公约数也是 a, b 的公约数。

既然两式公约数都是相同的，那么最大公约数也会相同。

所以得到式子 $\gcd(a, b) = \gcd(b, a \bmod b)$

既然得到了 $\gcd(a, b) = \gcd(b, r)$，这里两个数的大小是不会增大的，那么我们也就得到了关于两个数的最大公约数的一个递归求法。

证明

1. 证明是双向的：证明a，b的公约数一定是b，a mod b的公约数，以及反过来
2. 纯记过程。

极高的速度，极端适合计算机的形式。

```
//辗转相除法，求两个数的最大公因数
int gcd(int a, int b)
{
    if (b == 0)
        return a;
    else
        return gcd(b, a%b);
}
```

扩展欧几里得

https://zhuanlan.zhihu.com/p/100567253
tnnd谁爱看谁看吧，工程问题比数学好搞多了。

乘法逆元

先来说说我们需要什么，对于 b mod n 这个式子来说，我们需要求出一个整 数 a，使得 `a*b mod n=1`。另外，当且仅当b与n互素有逆元。
扩展欧几里得能求，但不要求。

具体过程如下：尝试`n+1,2n+1,3n+1...`直到有一个能整除b。整除结果即为逆元。

ppt-6

多重加密和三重DES

DES可被穷举攻击所以出现了三重DES。
双重有中间攻击，三重即使只用两个秘钥也无法被攻击。

分组密码工作模式

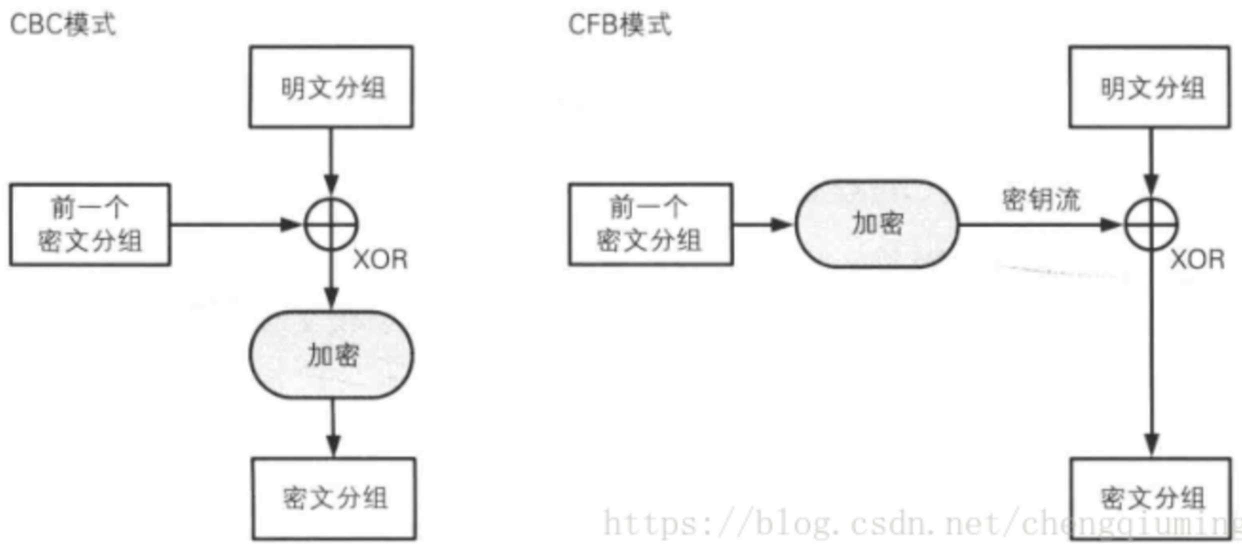
ECB

明文分64的分组，用同一秘钥加密。相同明文得到相同密文。
适合数据少的情况。长消息并不安全。

CBC

用当前明文分组和前一密文分组异或，当做明文加密。同一秘钥。
需要初始向量（第0密文分组）

CFB?



相当于使用DES，转化明文流为密文流（用随机的加密流进行异或）。

OFB?

CTR?

ppt-8

单向函数和单向陷阱门函数

由任意x到y易得。
由y到x不可行。
陷阱门：获得暗门信息，则由y到x可行。

离散对数问题

离散对数的定义方式和对数类似。取有原根的正整数模数 m，设其一个原根为 g. 对满足 $(a,m)=1$ 的整数 a，我们知道必存在唯一的整数 $0\le k<\varphi(m)$ 使得

$$g^k\equiv a \pmod m$$

我们称这个 k 为以 g 为底，模 m 的离散对数，记作 $k=\operatorname{ind}_g a$，在不引起混淆的情况下可记作 $\operatorname{ind} a$.

显然 $\operatorname{ind}_g 1=0$，$\operatorname{ind}_g g=1$.

详细部分以后再加。

对于公钥交换：

如果a是素数p的一个原根(本原元素)，则

a mod p, a^2 mod p, ..., a^{p-1} mod p，生成模p的完全剩余集 {1, 2, ..., p-1}

对于所有素数，其原根必定存在，即

对于一个整数b和素数p的一个原根，可以找到唯一的指数i，使得 $b = a^i \bmod p$, 其中 $0 <= i <= p-1$

指数i称为b的以a为基数的模p的离散对数或者指数。

注意：b也须在模p的完全剩余集内。原根a有 `2<= a <= p-1`。

因数分解

给定大素数 p和q, 求n = p×q, 只要一次乘法

给定n, 求p和q, 即为因数分解问题(FAC), 最快方法需要 $T(n) = \exp\{c(\ln n \ln (\ln n))^{1/2}\}$ 次运算, 其中c为大于1的正整数。若p≈n, 解离散对数比因数分解难。

背包问题

给定有限个自然数序列集合B=(b1,b2,…bn)及二进制序列x=(x1,x2,…xn), xi∈(0,1), 求S=∑xibi最多只需n-1次加法；但若给定B和S, 求x则非常困难。

```
1,2,3,4,5,6,7
0,1,1,0,1,0,1
S = 2+3+5+7 =17
相对应，给出17和1,2,3,4,5,6,7，求二进制串则困难得多。
```

单向函数交换性

只有那些具有该性质的单向函数对密码学用处很大

定义8.3 交换性

令Z为一集合，F为将Z映射到Z本身的函数集合。

令z∈Z, $F_x(z)$表示此函数集合之第x函数,

若$F_x(F_y(z))=F_y(F_x(z))$，则称此函数集合具有交换性。

例：D(E(m))=E(D(m))

集合Z：首先，我们有一个集合Z，它包含了所有可能的输入或元素。这些元素可以是数字、字符、消息等，具体取决于上下文。
函数集合F：然后，我们有一个函数集合F，其中每个函数FxFx都是从集合Z映射到Z本身的映射。这意味着每个函数FxFx接收一个来自Z的元素作为输入，并返回Z中的另一个元素作为输出。这里，“第x函数”指的是集合F中的一个特定函数，用x来索引或标识它。
交换性定义：当我们说函数集合F具有交换性时，意味着对于集合F中的任意两个函数FxFx和FyFy，以及Z中的任意元素z，函数的复合操作满足交换律。即，先应用FxFx再应用FyFy的结果与先应用FyFy再应用FxFx的结果是相同的。用数学语言表达就是：$Fx(Fy(z))$=$Fy(Fx(z))$Fx(Fy(z))=Fy(Fx(z))
这意味着函数的顺序可以交换而不影响最终的输出结果。

费马定理

费马小定理： 若p为素数，a为正整数。a不能被p整除（p/a不为整数），则a^(p-1) mod p = 1。
例：a = 7，p = 19, 7^18 mod 19 = 1

等价形式：a^p 同余 a mod p，p为素数

证明

设一个质数为 p，我们取一个不为 p 倍数的数 a。

构造一个序列：$A = \{1, 2, 3 \ldots, p-1\}$，这个序列有着这样一个性质：

$$\prod_{i=1}^{p-1} A_i \equiv \prod_{i=1}^{p-1}(A_i \times a) \pmod p$$

证明：

$$\because (A_i, p) = 1, (A_i \times a, p) = 1$$

又因为每一个 $A_i \times a \pmod p$ 都是独一无二的，且 $A_i \times a \pmod p < p$

得证（每一个 $A_i \times a$ 都对应了一个 A_i）

设 $f = (p-1)!$, 则 $f \equiv a \times A_1 \times a \times A_2 \times a \times A_3 \cdots \times A_{p-1} \pmod p$

$$a^{p-1} \times f \equiv f \pmod p$$
$$a^{p-1} \equiv 1 \pmod p$$

证明

1. 构造序列：左侧Ai位于1到p-1内，右侧模后位于1到p-1内。对Ai倍数放大并进行对p模运算能够保持一一对应关系。构造是可能的。
2. 下方的三行就在证明构造序列的合法性。

3. f右侧是a^p-1和Ai连乘（p-1!）。忽略最后两行，考虑如下格式：f = a^p-1 x f mod p，有 1 = a^p-1 mod p，第一形式证明完成。

费马大定理：
一个立方数分成两个立方数之和；
一个四次幂分成两个四次幂之和；
或者一般地将一个高于二次的幂分成两个同次幂之和；
这是不可能的。

欧拉定理

欧拉函数

φ(n) 是比n小，与n互素（没有除1以外的公因数的数）的正整数个数。对于9，有1,2,4,5,7,8共6个

计算欧拉函数的值

https://www.cnblogs.com/hiflora/p/3171775.html

1. n = 1，φ(n) = 1
2. n为质数，φ(n) = n -1
3. n位质数的一个次方，有

   $$\phi(p^k) = p^k - p^{k-1}$$

   如9：9 - 3 = 6
4. n可以分解为两个互质整数之积，则有φ(n) = φ(p1 x p2) = φ(p1) x φ(p2)。
5. 对于对普通的情况，有通用解法：由于整数的唯一分解定理，任何大于1的整数可以分解为质数之积，形式唯一有以下推导

因为任意一个大于 1 的正整数，都可以写成一系列质数的积。

$$n = p_1^{k_1} p_2^{k_2} ... p_r^{k_r}$$

根据第 4 条的结论，得到

$$\phi(n) = \phi(p_1^{k_1}) \phi(p_2^{k_2}) ... \phi(p_r^{k_r})$$

再根据第 3 条的结论，得到

$$\phi(n) = p_1^{k_1} p_2^{k_2} ... p_r^{k_r} (1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) ... (1 - \frac{1}{p_r})$$

也就等于

$$\phi(n) = n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) ... (1 - \frac{1}{p_r})$$

这就是欧拉函数的通用计算公式。比如，1323 的欧拉函数，计算过程如下：

$$\phi(1323) = \phi(3^3 \times 7^2) = 1323(1 - \frac{1}{3})(1 - \frac{1}{7}) = 756$$

欧拉定理

对于互素的a和n，有**a^φ(n) 同余 1 mod n**

证明

欧拉定理

在了解欧拉定理（Euler's theorem）之前，请先了解 欧拉函数。定理内容如下：

定义

若 $\gcd(a, m) = 1$，则 $a^{\varphi(m)} \equiv 1 \pmod{m}$。

证明

实际上这个证明过程跟上文费马小定理的证明过程是非常相似的：**构造一个与 m 互质的数列**，再进行操作。

设 $r_1, r_2, \cdots, r_{\varphi(m)}$ 为模 m 意义下的一个简化剩余系，则 $ar_1, ar_2, \cdots, ar_{\varphi(m)}$ 也为模 m 意义下的一个简化剩余系。所以 $r_1r_2\cdots r_{\varphi(m)} \equiv ar_1 \cdot ar_2 \cdots ar_{\varphi(m)} \equiv a^{\varphi(m)}r_1r_2\cdots r_{\varphi(m)} \pmod{m}$，可约去 $r_1r_2\cdots r_{\varphi(m)}$，即得 $a^{\varphi(m)} \equiv 1 \pmod{m}$。

当 m 为素数时，由于 $\varphi(m) = m - 1$，代入欧拉定理可立即得到费马小定理。

1. ?

中国剩余定理

中国剩余定理 (Chinese Remainder Theorem, CRT) 可求解如下形式的一元线性同余方程组（其中 $n_1, n_2, \cdots, n_k$ 两两互质）：

$$\begin{cases} x & \equiv a_1 \pmod{n_1} \\ x & \equiv a_2 \pmod{n_2} \\ & \vdots \\ x & \equiv a_k \pmod{n_k} \end{cases}$$

上面的「物不知数」问题就是一元线性同余方程组的一个实例。

过程

1. 计算所有模数的积 n；
2. 对于第 i 个方程：
    a. 计算 $m_i = \frac{n}{n_i}$；
    b. 计算 m_i 在模 n_i 意义下的 逆元 m_i^{-1}；
    c. 计算 $c_i = m_i m_i^{-1}$（**不要对 n_i 取模**）。
3. 方程组在模 n 意义下的唯一解为：$x = \sum_{i=1}^{k} a_i c_i \pmod{n}$。

我们需要证明上面算法计算所得的 x 对于任意 $i = 1, 2, \cdots, k$ 满足 $x \equiv a_i \pmod{n_i}$。

当 $i \neq j$ 时，有 $m_j \equiv 0 \pmod{n_i}$，故 $c_j \equiv m_j \equiv 0 \pmod{n_i}$。又有 $c_i \equiv m_i \cdot (m_i^{-1} \bmod n_i) \equiv 1 \pmod{n_i}$，所以我们有：

$$\begin{aligned} x &\equiv \sum_{j=1}^{k} a_j c_j && \pmod{n_i} \\ &\equiv a_i c_i && \pmod{n_i} \\ &\equiv a_i \cdot m_i \cdot (m_i^{-1} \bmod n_i) && \pmod{n_i} \\ &\equiv a_i && \pmod{n_i} \end{aligned}$$

即对于任意 $i = 1, 2, \cdots, k$，上面算法得到的 x 总是满足 $x \equiv a_i \pmod{n_i}$，即证明了解同余方程组的算法的正确性。

因为我们没有对输入的 a_i 作特殊限制，所以任何一组输入 $\{a_i\}$ 都对应一个解 x。

另外，若 $x \neq y$，则总存在 i 使得 x 和 y 在模 n_i 下不同余。

故系数列表 $\{a_i\}$ 与解 x 之间是一一映射关系，方程组总是有唯一解。

证明

1. i≠j时，mj整除以ni。原因是mj是除了nj以外所有ni的积，其余所有ni是mj的因数。可得cj = 0
2. mj是cj的因数，因而，同余。原因：cj = mj · mj对ni的逆
3. ci = mi · **mi对ni的逆**，则ci mod ni = 1
4. 考虑x的式子：$x = \sum_{i=1}^{k} a_i c_i \pmod{n}$。，由于n为ni们的积，x mod n mod ni = x mod ni。因而本式可以直接把mod n转为 mod ni
5. 对于任意i，和式x在mod ni时，和式中任何j≠i的cj = 0，该项消失。转为aici mod ni
6. 因而对任意ni，x = ai mod ni，正确性检验。
7. **一一对应关系，不同的x对ni不同余。（该项不知道意义）**
   重要过程：1,4,5

ppt-9

公开密钥密码体制

特点：加密与解密分离。密钥（公钥）分发简单，无需可靠信道。n个用户保存n个密钥（私钥）。可满足不相识的人之间保密通信。可以实现数字签名。

公钥密码体制的组成

1. 明文：算法的输入，可读信息或数据
2. 加密算法：对明文进行各种转换
3. 公钥和私钥：算法的输入，分别用于加密和解密
4. 密文：算法的输出，依赖于明文和密钥
5. 解密算法：根据密文和密钥，还原明文
   知道两个算法和任何一个密钥无法推导出另一个密钥。

体会一下公钥体制实现保密性和身份认证的巧妙：

1. 用户持有自己的私钥（身份证）和公钥。
2. A向B发消息：A用B的公钥加密，B用自己的私钥解密
3. A向B验证自己的身份：A用自己的私钥加密身份认证消息，B可以用A的公钥解密证明这是A的消息。

安全性

公钥密码易受穷举攻击——对一个消息穷举所有秘钥可能。尽管如此，如RSA的1024位秘钥，极广的秘钥范围还是让破解几近不可能。
从给定公钥计算私钥的攻击——数学上没尚未证明不可行。但能做到也就意味着公钥依托的数学难题被解决，至今没有例子。不过仍有一些实现漏洞和参数不当实现了计算私钥的事发生过。
穷举消息攻击——用公钥加密所有可能的消息，匹配传送密文。这更是个理论可行。

非对称加密还是很安全的。不过通常不会用于大量消息的加密，而是结合对称加密，用于对称加密的秘钥传输。以及身份认证（数字签名）。

RSA

基于大合数的质因子分解问题。
分组密码体制，明文密文

过程

1. 随机选择不等质数p = **61**，q = **53**
2. 计算乘积n = p x q = **3233**，n的二进制位数为秘钥长度。
3. 计算n的欧拉函数φ(n) = 3120
4. 随机选择e，小于φ(n)，大于1，与φ(n)互质，通常是65535。令**e = 17**
5. 计算e对于φ(n)的模反元素d，即e x d mod φ(n) = 1。d = 2753
6. 公钥为n，e，即3233,17；私钥为n，d，即3233,2753

想要破解，则需根据3233,17，计算出2753。即寻找一个数，使其与17的积 模 3120的欧拉函数值为1.
所以，必须知道欧拉函数的值，才能迅速算出来（这是离散对数问题，由于e和φ(n)互质，扩展欧几里得算法能在O(log(min(y,c)))的时间复杂度内计算这个数，对数时间）。
然而，n大于1，n是质数积就是合数，两个不等质数之积也不会是质数的次方。不符合上面计算欧拉函数的前三个方法。
剩余方法为质因数分解，唯一分解。同时，**唯一分解的唯一性确保分解结果为所选两个质数的积**
因此，解决大整数的质因数分解就能解决RSA。
当然，想得美。

ppt-10

在公钥密码体系中，系统的安全性不仅仅依赖于加密算法本身的强度，还高度依赖于公钥的真实性和完整性。即使加密算法再漂亮，如果有办法替换公钥，则毫无安全可言。
确保公钥的真实性和来源的可信是非常关键的，这通常通过数字证书和证书权威机构（CA）来实现。
此外，私钥的安全性同样重要。但其关注点在于私钥的隐私保护，而不是公钥加密体系。通常使用物理安全措施、加密存储以及访问控制等手段。

公钥分配

自由公开发布

用户把公钥发给对方，或者直接在平台上广播，或者放到某些邮件列表中。
问题在于伪造公钥发布非常容易。

公开可访问的目录

一个可信实体或组织维护和分配这个公开目录。

- 目录包含{name, public-key}等项
- 每一通信方通过目录管理员以安全的方式注册一个公钥
- 通信方在任何时刻可以用新的密钥替代当前的密钥
- 目录定期更新
- 目录可通过电子方式访问

问题：获得目录管理员的私钥，即可伪造公钥，甚至假冒任何通信方。

公钥授权

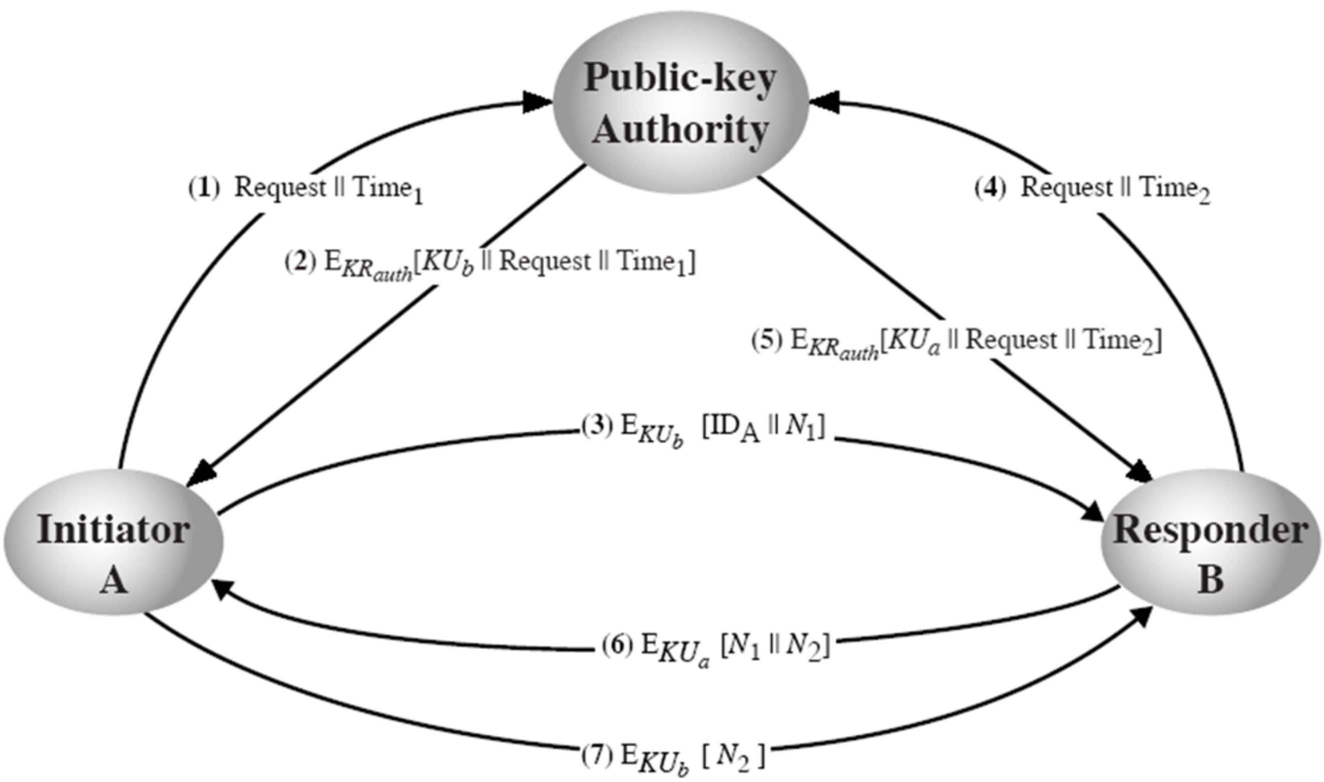


Figure 10.3 Public-Key Distribution Scenario

1. A发送有时间戳的消息给公钥管理员，请求B的公钥。
2. 管理员用自己的私钥加密（A发送的时间戳，A发送的消息，B的公钥），返回给A
3. A用管理员的公钥验证，确信来自管理员并获得了B的公钥。而后，向B发送用B的公钥加密的消息（A的身份标识，一次性随机数N1）

4. B接收到A的消息，进行解密获得A的身份。并根据解密成功反向确认A已经获得自己的公钥，具有可信度。重复上述向管理员申请A的公钥的过程。
5. B获得了A的公钥后，向A发送用A公钥加密的确认消息（A的一次性随机数N1，新一次性随机数N2）
6. A接收到消息，确认B已经获取自己的公钥。返回B公钥加密的随机数N2
7. 注解：很像tcp的三次握手。以及对方的公钥就是身份确认器。
   这种方法需要实时访问授权部门。

公钥证书

公钥证书可以不通过实时访问授权部门。
公钥证书绑定一个通信方和他的公开密钥

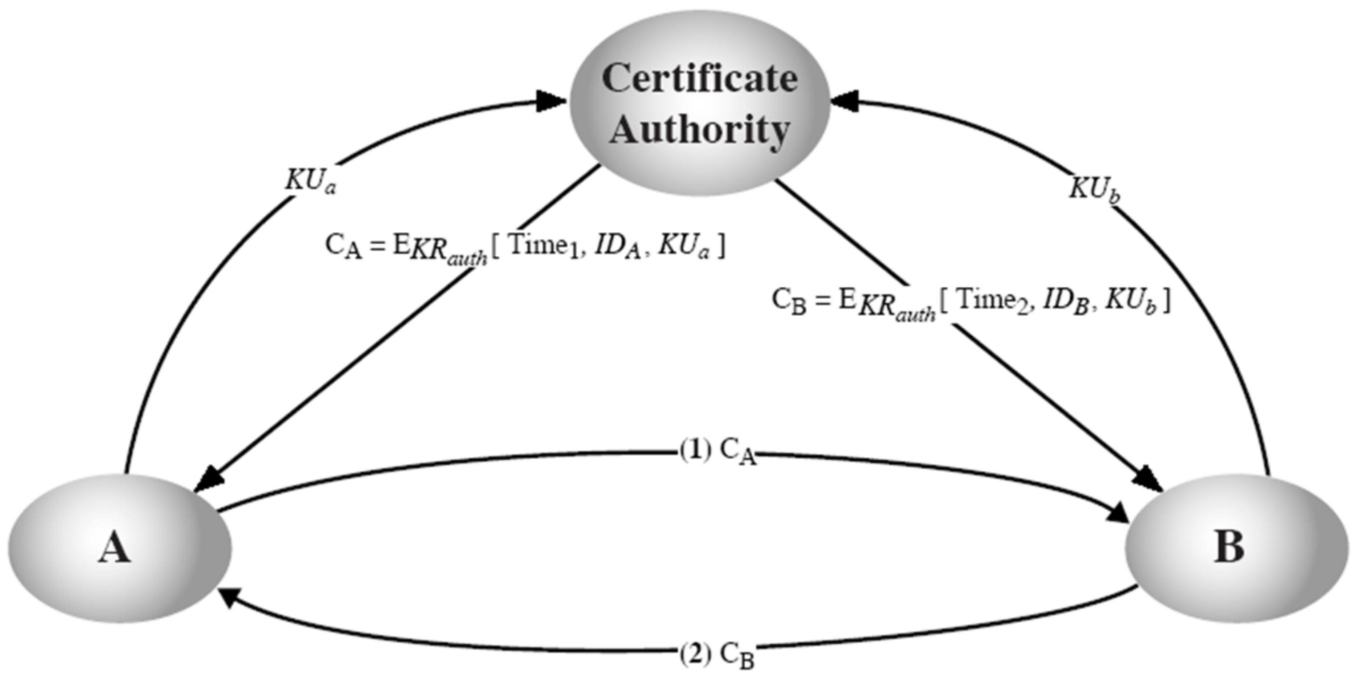


Figure 10.4 Exchange of Public-Key Certificates

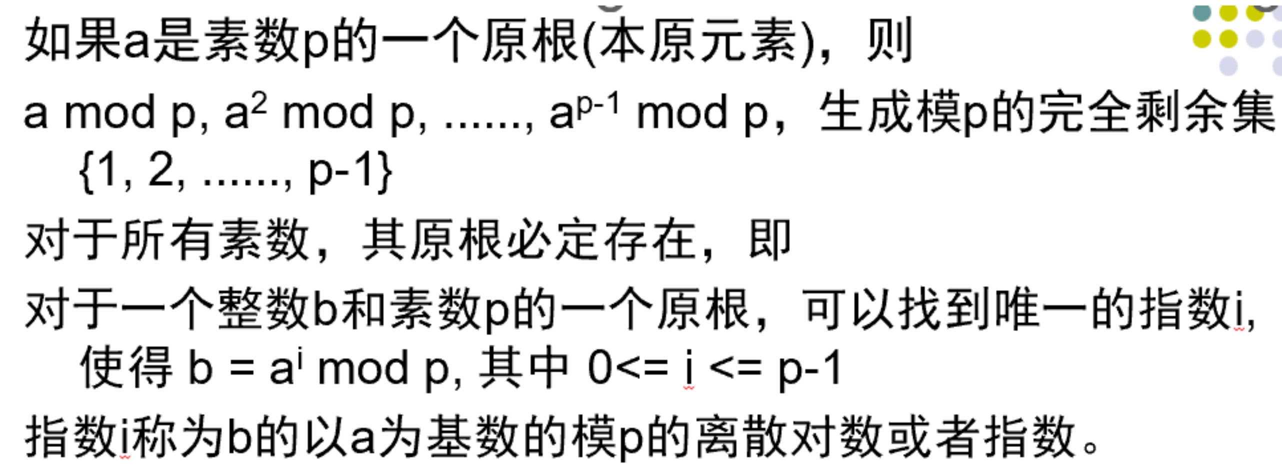
CA证书：认证者用自己的私钥加密三个内容（A提供的A的公钥，提供的时间，A的身份），生成证书。
B得到这个证书，用认证者的公钥解密，确认这是未篡改的证书（没人拥有认证者的私钥）。B获得A的公钥，并在建立与A的联系后能够确认对方是A（只有A有A的私钥）

分配传统密码体制的秘钥

证书体系能够认证通信方，能够保密。
通常公钥体系用于传统密码秘钥的分配，需要产生会话密码来加密通信。

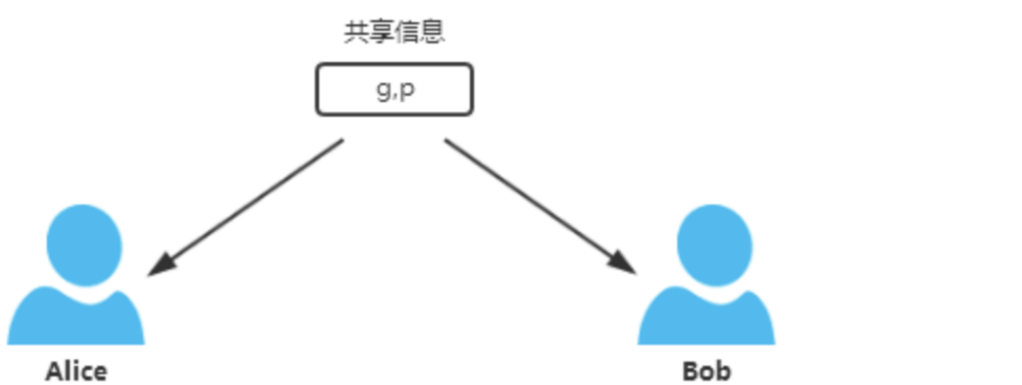
Diffie-Hellman密钥交换

参考
双方通过一系列计算步骤，利用各自的私钥、对方的公钥以及一些公开的参数（如大素数p和一个原根g），能够神奇地达成一个相同的秘密值，这个秘密值就是所谓的共享密钥或会话密钥。
注意：私钥和公钥只是一个自己拿着和公开出去的概念，并不特指某些文件或数。
基于有限域GF中的指数运算的(模一素数或多项式)，安全性依赖于离散对数问题。

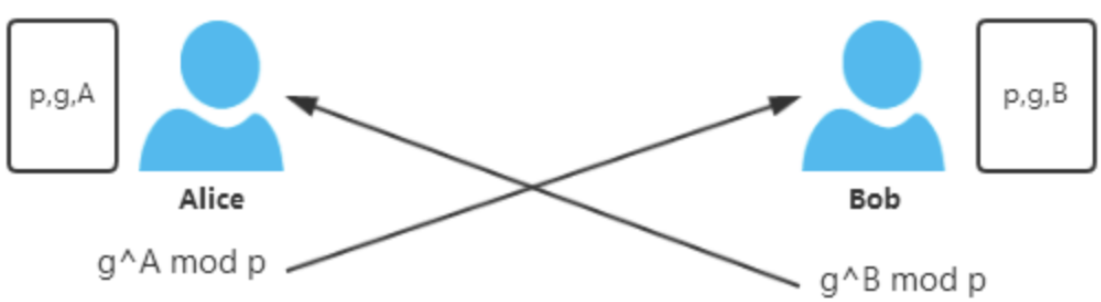
> 
>
>
> 注意：b也须在模p的完全剩余集内。原根a有 `2<= a <= p-1`。

过程

1. 素数必然存在原根，双方一起选取素数p，原根g。该项在不安全的网络中交换。
   原根大于2，小于p-1



2. s双方产生私有随机数A，B（大于1，小于p-1），计算Ya = g^A mod p，Yb，发送给对方



3. 协商结束。Alice获得p，g，A，Yb；Bob获得p，g，B，Ya。
4. 相同的会话秘钥：Alice执行Ka = Yb^A mod p；Bob执行Kb = Ya^B mod p。Ka必然等于Kb（见证明）

证明

代入并展开即可。

$$K_a = (Y_b)^A\ mod\ p = (g^B\ mod\ p)^A\ mod\ p = g^{B\times A}\ mod\ p$$

对于Bob有：

$$K_b = (Y_a)^B\ mod\ p = (g^A\ mod\ p)^B\ mod\ p = g^{A\times B}\ mod\ p$$

中间人攻击

架桥。
A和伪造的Bk通信，B和伪造的Ak通信，有两个会话秘钥ka和kb。中间人解密信息再加密送给对方。

ElGamal

公钥加密而非秘钥交换。与DH相近，同样需要共享素数p和原根g，选取自己的私钥。但是多出一个消息m

过程

1. 选取大素数p和原根g，消息m，共享。现在，Bob需要给Alice发消息
2. Alice选取私有随机数Xa（私钥），计算Ya = g^Xa mod p，将Ya（公钥）给Bob。这是公钥交换，一次就可以。
3. Bob选取私有随机数Xb，计算公共秘钥K = Ya^Xb mod p，公共秘钥获取器c1 = g^Xb mod p，密文c2 = m x K mod p。c1和c2发送给Alice
4. Alice接收到c1c2。此时公共秘钥K = Ya^Xb mod p = g^(XaXb) mod p。Alice计算K = c1 ^ Xa mod p，明文m = c2 x （K在mod p下的逆元）
   注意：每次通信发起方的随机数不可以重复使用。
   证明：Bob进行两次c1c2计算，并发现同随机数下两次明文的比m1/m2 为两次密文c21c22的比

证明

暂无

ppt-11

消息认证

注意其主要目的：

1. 验证消息与发送时一样
2. 发送方声称的身份是真实的
3. 消息源的不可否认（发起方无法否认自己发送过该消息）
   上述的公钥加密方案确实可以提供一种形式的身份认证：消息发送方的私钥加密消息，即可基于私钥私有验证身份。（但消息认证不局限于此）

加密：更难被攻击

消息加密

消息加密本身就是一种身份认证手段。

对称加密的身份认证

由于秘钥只有两方有，可以直接认证对方。

公钥加密的身份认证

如上所述，发送方用私钥加密，接收方用发送方公钥解密，即可认证。（攻击者用发送方公钥也可以看到明文）
如果需要认证和保密，就用私钥加密，对方公钥再加密。

消息认证码

参考
最简单的：用秘钥k哈希消息M获得MAC，将其贴到M后发送出去原根。
显然，消息未被更改（接收方MAC计算相同），身份认证（秘钥只有两者知道）。
通常，将M+MAC进行加密后再发送（2.0）。

MAC无法提供数字签名，因为双方共享秘钥（对称结构）。

安全散列函数

ppt-13

数字签名

一个起签名作用的码字：计算消息M的哈希，并用私钥进行加密，生成签名。
哈希确保消息完整性，私钥确保消息的来源。

消息认证需要双方共享密钥，但数字签名不用。接收方只需要验证就行。

- 数字签名的基本形式
  - 对消息签名的两种方法
    - 对消息整体的签字，将被签消息整体经过密码变换得到签字；
    - 对消息摘要的签字，附在被签消息之后，或嵌在某一特定位置上作一段签字图样。
  - 两类数字签名
    - 确定性数字签名，明文与签名一一对应；
    - 概率性数字签名，一个明文可以有多个合法签名，每次都不一样。

直接数字签名

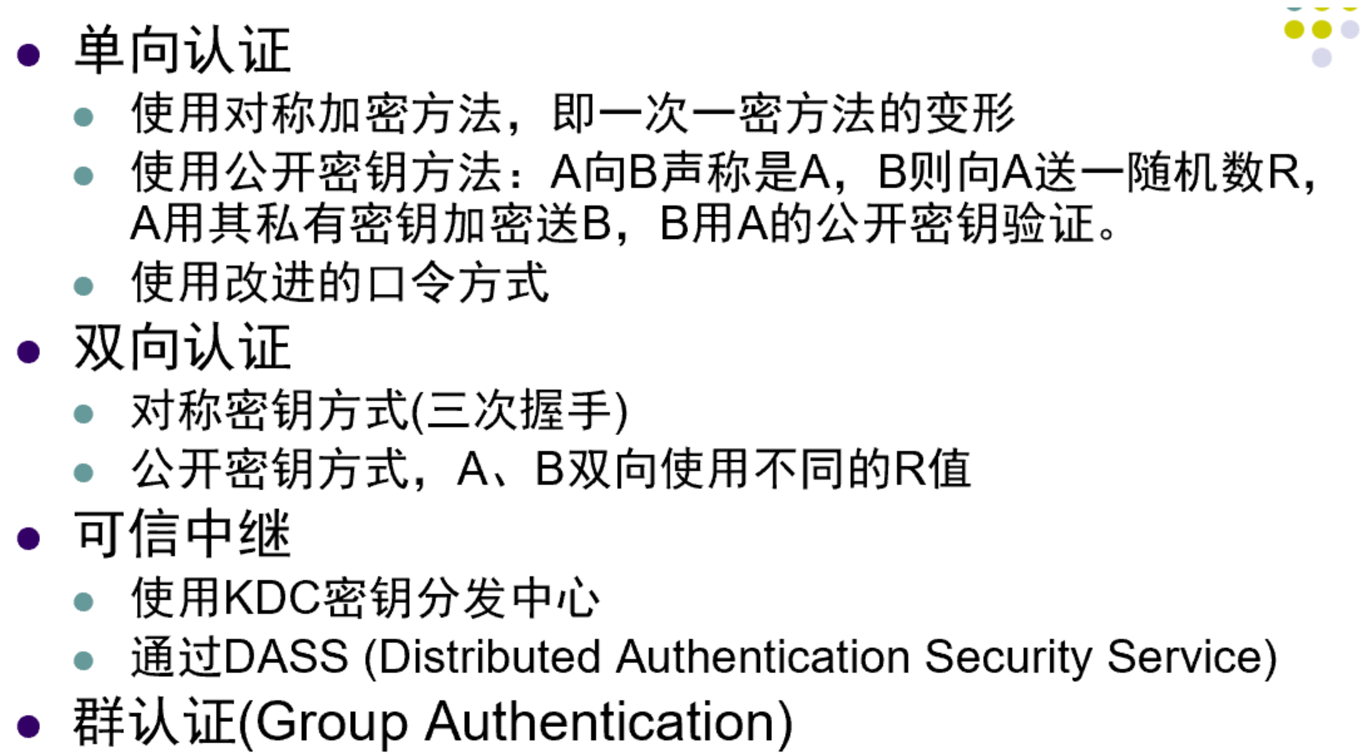
仅涉及通信方，并假定接受者知道发送者的公钥。
对报文或摘要进行哈希，再用私钥加密报文和签名。
安全性完全依赖于私钥保存的安全性。

仲裁数字签名

有一个仲裁方：发送方把签名报文发给仲裁者，仲裁者对其测试并注上日期和仲裁说明后发给接收方。
感觉没啥提高，要求仲裁方可信。

认证协议

消息认证的超集。上述消息认证的方法都在单向认证中。



ElGamal数字签名

A和B共享素数p，原根g，消息m。各自的私钥Xa，Xb，公钥Ya = g^Xa mod p，Yb。
另外，消息`m∈[0, p-1]`（不知为何）

现在A为B签署m（向B发送消息），有：

1. A选择随机数k（不是私钥那个随机数），满足k与p-1互素
2. 计算第一部分签名r = g^k mod p
3. 计算K = k逆 mod p-1
4. 计算第二部分签名`s = (m - Xa*r) * K`
5. 签名（r，s）
   B进行验证：
6. 第一签名：V1 = g^m mod p
7. 第二签名：V2 = Ya^r 乘 r^s mod p
8. 两签名必须相等。