第 2 章　密码技术

第 2 章　密码技术

- 2.1　古典密码
- 2.2　对称密码体制
- **2.3　公钥密码体制**

2.3 公钥密码体制

- 2.3.1 公钥密码概述
- 2.3.2 公钥密码体制数学基础
- 2.3.3 Diffie-Hellman密钥交换
- 2.3.4 RSA
- 2.3.5 其他算法

2.3.1 公钥密码思想

- Diffie和Hellman在1976年首次提出了公钥密码思想，是密码领域的一次真正革命性进步
  - 公钥加密算法基于数学函数（单向陷门函数），代替对称加密算法的比特模式简单操作（替换和置换）
  - 解决两个问题：密钥的分发与管理、数字签名及认证
  - 每个用户有一对选定的密钥
    - 公钥：可以公开，用于加密和验证签名
    - 私钥：自己保存，不能公开，用于解密和签名
  - 三类应用：加密/解密、数字签名、密钥交换

公钥密码体制与对称密码体制

- **公钥密码体制有两个不同的密钥，在公钥密码体制中，加密密钥(即公钥) PK 是公开信息，而解密密钥(即私钥或秘钥) SK 是需要保密的。**
- **加密算法 E 和解密算法 D 也都是公开的。**

公钥密码算法的表示

- **加密与解密分开：**密钥对产生器产生出接收者 B 的**一对密钥：**加密密钥 PK_B 和解密密钥 SK_B。
  - 加密密钥 PK_B 就是接收者B的**公钥，**它向公众公开。
  - 解密密钥 SK_B 就是接收者B的**私钥，**对其他人都保密。
- **保密通信：**发送者 A 用 B 的**公钥** PK_B 对明文 X **加密**（E 运算）后，在接收者 B 用自己的**私钥** SK_B **解密**（D 运算），即可恢复出明文：

$$D_{SK_B}(Y)=D_{SK_B}(E_{PK_B}(X))=X$$

公钥密码算法满足的需求

- 产生密钥对(公钥PK_B和私钥SK_B)在计算上是容易的
- 发送方A用接收方B的公钥对消息m加密产生密文c，即$c=E_{PK_B}(m)$，在计算上是容易的
- 接收方B用自己的私钥对密文c解密得到明文m，即$m=D_{SK_B}(c)$，在计算上是容易的
- 敌对方T由B的公钥PK_B求私钥SK_B在计算上是不可行的
- 敌对方T由密文c和B的公钥PK_B恢复明文m在计算上是不可行的
- 加密、解密次序可交换，$E_{PK_B}(D_{SK_B}(m))=D_{SK_B}(E_{PK_B}(m))$

公钥加密

- 基本要素
  - 明文Plaintext
  - 加密算法Encryption Algorithm
  - 公钥Public Keys和私钥Private key
    - 公钥环（多个用户的合法公钥）
    - 自己的私钥
  - 密文Ciphertext
  - 解密算法Decryption Algorithm

公钥算法的特点

- 加密密钥是公开的，但不能用它来解密，即：

$$D_{PK_B}(E_{PK_B}(X)) \neq X$$

- 加密和解密运算可以对调，即加密和解密是互逆的：

$$E_{PK_B}(D_{SK_B}(X)) = D_{SK_B}(E_{PK_B}(X)) = X$$

- 无需事先分配密钥，减少了密钥数量

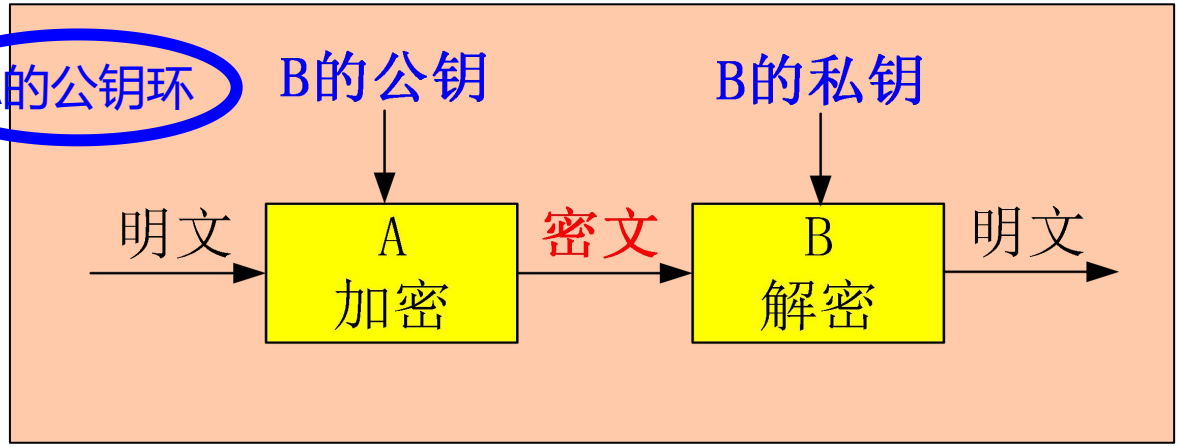
公开密钥与对称密钥的区别

对 称 密 码	公 钥 密 钥
一般要求 （1）加密和解密使用相同的密钥和相同的算法 （2）收发双方必须共享密钥	**一般要求** （1）同一算法应用于加密和解密，但使用不同的密钥 （2）收送方拥有加密或解密密钥，而接收方拥有另一密钥
安全性要求 （1）密钥必须是保密的 （2）若没有其他信息，则解密消息是不可能或至少是不可行的 （3）知道算法和若干密文不足以确定密钥	**安全性要求** （1）私钥必须是保密的 （2）若没有其他信息，则解密消息是不可能或至少是不可行的 （3）知道算法和公钥以及若干密文不足以确定私钥

两种基本模型

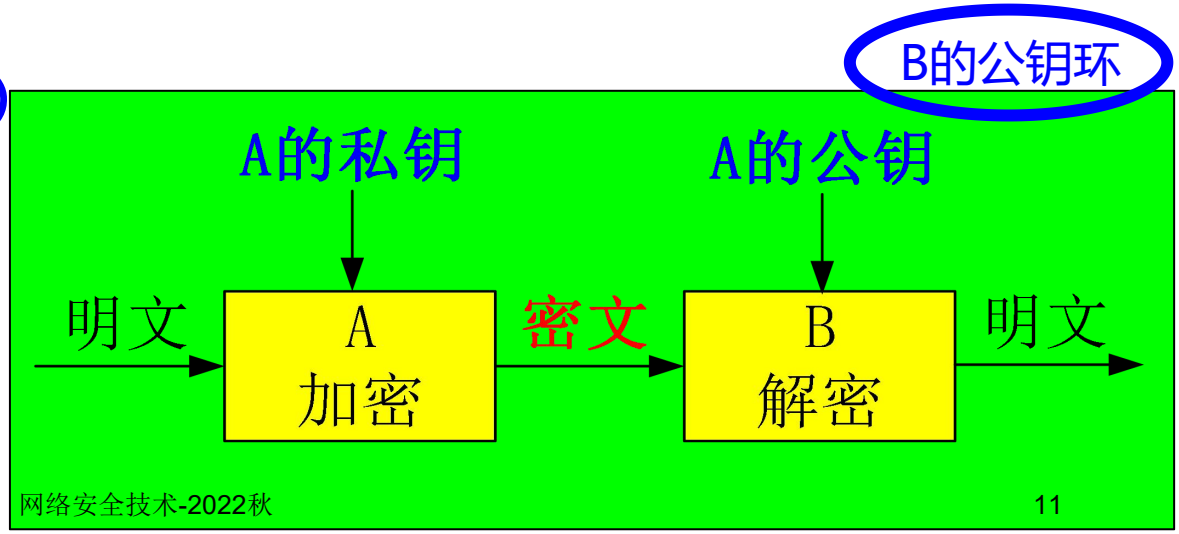
- **加密模型**

信息用**公开密钥加密，用私有密钥解密，**实现**保密**的目标



- **认证模型（数字签名）**

信息用**私有密钥加密，**必须用**公开密钥解密，**实现对发送方的认证



公钥密码系统的应用

- **加密/解密**：发送者用接收者的公钥加密消息
- **数字签名**：发送者用自己的私钥“签名”消息。签名可以通过对整条消息加密或者对消息的一个小的数据块(消息的函数)加密来产生。
- **密钥交换**：通信双方交换会话密钥。多种方法，且需要用到通信一方或者双方的私钥。

算　法	加密/解密	数字签名	密钥交换
RSA	是	是	是
ElGamal	是	是	是
Diffie-Hellman	否	否	是
DSA	否	是	否

2.3.2 公钥密码体制数学基础

- 陷门单向函数是满足下列条件的函数f
  - (1) 给定x, 计算$y=f_k(x)$是容易的；
  - (2) 给定y, 计算$x=f_k^{-1}(y)$是不可行的(NP难)；
  - (3) 存在k, 已知k时，对给定的任何y,若相应的x存在，则计算x使$x=f_k^{-1}(y)$是容易的。
- 举例
  - 计算：7919*7927=62 773 913 容易
  - 分解：62 773 913 困难

单向函数(One way functions)

- **大整数分解困难问题**

已知大整数N，求素因子p和q（N=pq）是计算困难的

Challenge number	Difficulty of factoring to the two primes
15	Everyone can do this instantly
143	Doable with a little thought
6887	Should not take more than a few minutes
31897	A calculator is now useful
20-digit number	A computer is now required
600-digit number	This is impossible in practice
600-digit even number	One factor immediate, other easily computed
600-digit number with small factor	One factor easily found, other easily computed

公钥密码体制数学基础

- 研究公钥密钥算法就是找出合适的陷门单向函数
- 典型数学中的难解问题
  - 大整数因子分解问题：RSA公钥密码体制
  - 基于有限域上的离散对数问题： ElGamal公钥密码体制
  - 基于椭圆曲线上的离散对数问题：椭圆曲线公钥密码体制

2.3.4 Diffie-Hellman密钥交换

- 离散对数
  - 如果整数a是素数p的本原根，那么：

    $a \bmod p, a^2 \bmod p, a^3 \bmod p, \ldots, a^{n-1} \bmod p$

    由1到p-1的整数组成且各不相同
  - 设p为素数，a为模p的本原根，a的幂乘运算为

    $$Y=a^X \bmod p \quad (1 \le X \le p-1)$$

    则称X为Y的以a为底的模p的离散对数，记做：

    $$X=\mathrm{dlog}_{a,p}(Y)$$
  - 对给定的X,a,p，计算Y是可行的；对给定的Y,a,p，计算X(即求离散对数)计算上非常困难

举例1：离散对数计算

- 计算7是否是71的本原根，即 $7^i \bmod 71$

7	49	59	58	51	2	14	27	47	45	31	4	28	54	23	19
62	8	56	37	46	38	53	16	41	3	21	5	35	32	11	6
42	10	70	64	22	12	13	20	69	57	44	24	26	40	67	43
17	48	52	9	63	15	34	25	33	18	55	30	68	50	66	36
39	60	65	29	61	1										

- 计算8是否是71的本原根，即 $8^i \bmod 71$

8	64	15	49	37	12	25	58	38	20	18	2	16	57	30	27
3	24	50	45	5	40	36	4	32	43	60	54	6	48	29	19
10	9	1	8	64	15	49	37	12	25	58	38	20	18	2	16
57	30	27	3	24	50	45	5	40	36	4	32	43	60	54	6
48	29	19	10	9	1										

单选题 4分

设置

3是素数7的本原根吗?

A 是

B 否

提交

举例2：计算素数19的本原根

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1	18
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1	18
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1	9
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1	9
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1	9
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	3
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1	6
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1	9
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1	18

举例：模19的整数幂(续)

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}	
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	3
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1	6
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1	18
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1	18
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1	18
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1	9
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1	9
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	2

19的本原根为：2、3、10、13、14、15

Diffie-Hellman算法

- 全局共享参数
  - q:素数
  - a: a<q，a是q的本原根
- 用户共享key的计算

Alice	
Select a private X_A	$X_A < q$
Calculate public Y_A	$Y_A = a^{XA} \bmod q$
Calculate K	**$K=Y_B^{XA} \bmod q$**

Bob	
Select a private X_B	$X_B < q$
Calculate public Y_B	$Y_B = a^{XB} \bmod q$
Calculate K	**$K=Y_A^{XB} \bmod q$**

Diffie-Hellman密钥交换

- 目的：使得两个用户能够安全地交换密钥
- 算法的有效性：

计算离散对数很困难

$K=(Y_B)^{X_A} \bmod q$

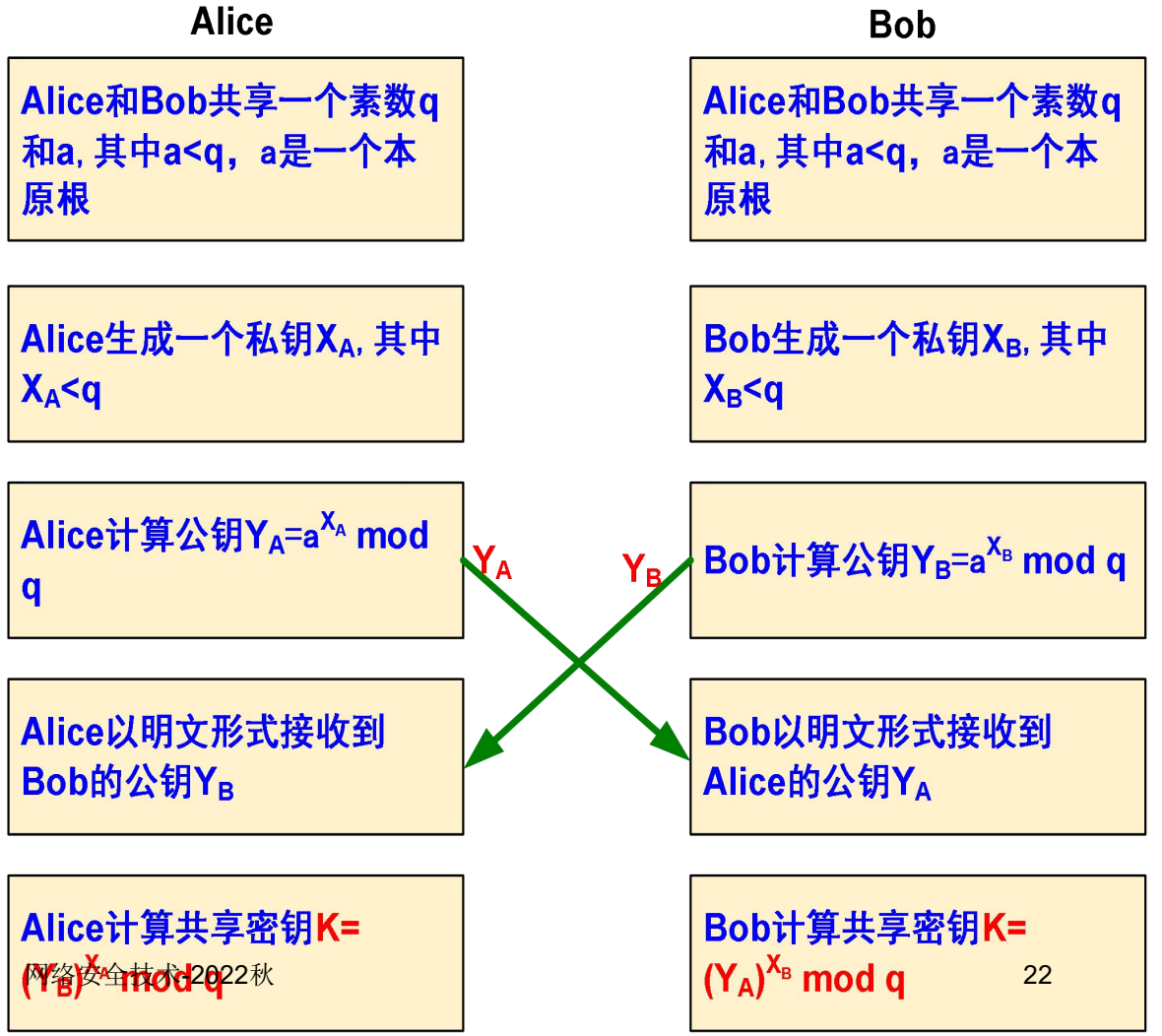
$=(a^{X_B} \bmod q)^{X_A} \bmod q$

$=(a^{X_B})^{X_A} \bmod q$

$=(a^{X_A})^{X_B} \bmod q$

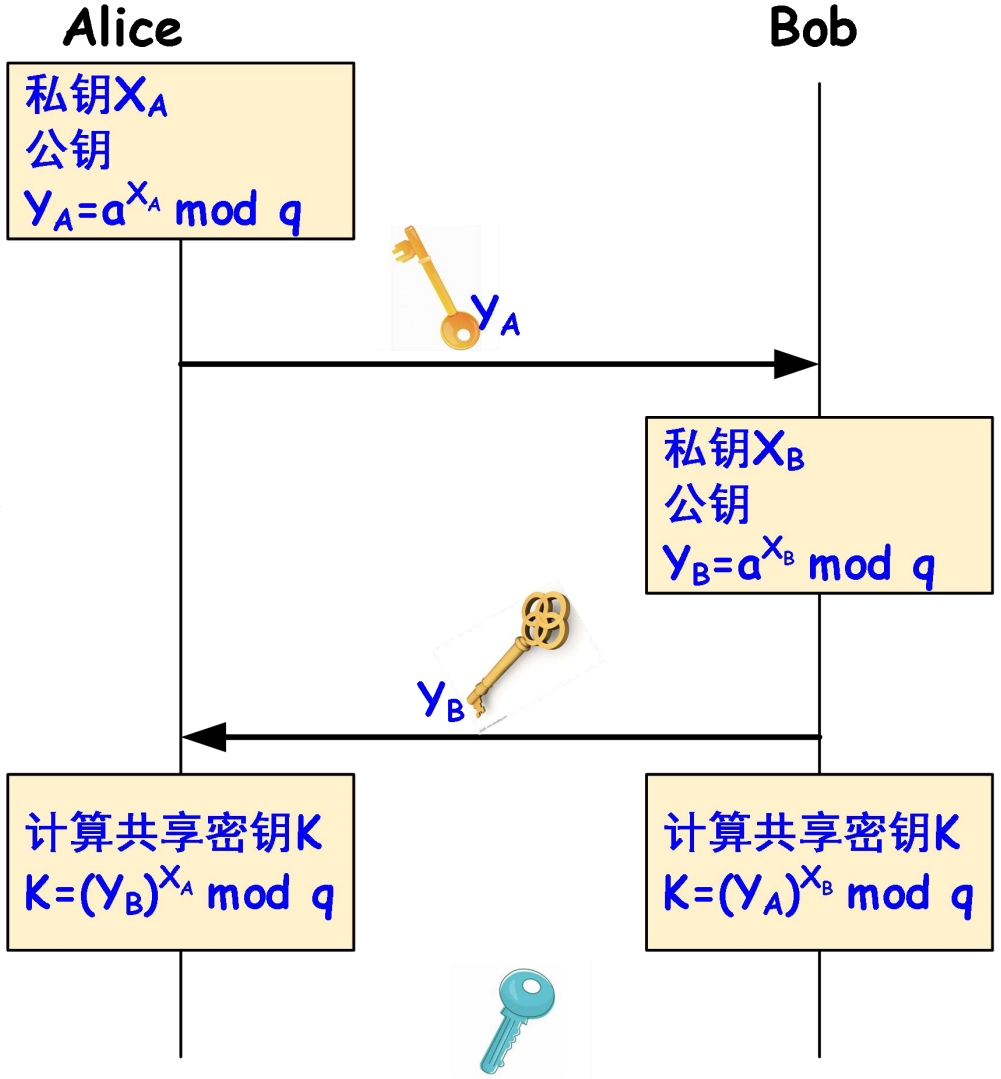
$=(a^{X_A} \bmod q)^{X_B} \bmod q$

$=(Y_A)^{X_B} \bmod q$



Diffie-Hellman

- **密钥交换协议**
  - 场景：用户Alice希望与用户Bob建立连接，并且在此连接上使用密钥加密消息。



Diffie-Hellman

- 举例
  - q=353, a=3
  - Alice X_A=97, Y_A=3^{97} mod 353=40
  - Bob X_B=233, Y_B=3^{233} mod 353=248
  - Key:
    - Alice：248^{97} mod 353=160
    - Bob: 40^{233} mod 353=160

填空题 3分

设置

此题未设置答案，请点击右侧设置按钮

用户A和B使用Diffie-Hellman算法交换密钥，已知共享素数q=31，本原根a=3.

(1) 如果用户A选择的私钥X_A=4,则Y_A=[填空1]

(2) 如果用户B选择的私钥X_B=2,则Y_B= [填空2]

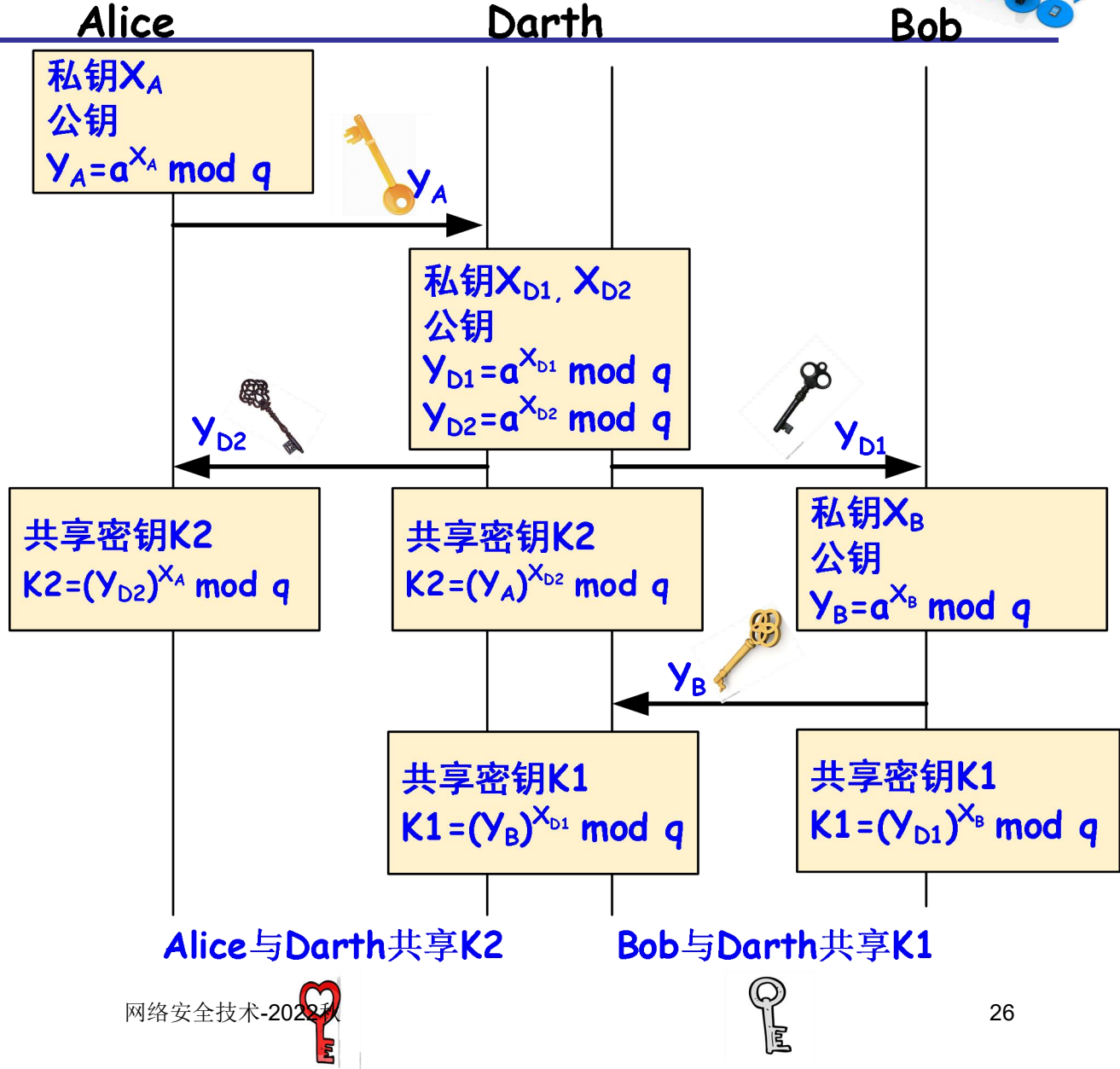
(3) A和B的共享密钥K= [填空3]

正常使用填空题需3.0以上版本雨课堂

作答

Diffie-Hellman

- 中间人攻击
  - D-H协议不能认证参与者，对于中间人攻击脆弱
  - 改进：使用数字签名和公钥证书



2.3.4 RSA算法

- RSA是一种可逆公钥密码体制，也是目前理论上最为成熟完善的公钥密码体制
- 大整数分解困难问题

已知大整数N，求素因子p和q（N=pq）是计算困难的

RSA算法

密钥产生	
select p,q	p and q both prime, p≠q
n=p×q	
Φ(n)=(p-1)×(q-1)	
select integer e	gcd(Φ(n), e)=1; 1<e<Φ(n)
calculate d	de ≡ 1 (mod Φ(n)) ;d<Φ(n)
public key	PK={e, n}
private key	SK={d, n}

⇐ 正整数n的欧拉函数Φ(n)

⇐ Φ(n)和e互质

加密	
plaintext	M<n
ciphertext	$C=M^e \bmod n$

解密	
ciphertext	C
plaintext	$M=C^d \bmod n$

RSA算法举例

- 已知p=17,q=11,e=7,求密钥和明文m=88的加解密过程
- 求密钥：
  - n=17*11=187
  - Φ(n)=(p-1)*(q-1)=16*10=160
  - ed mod Φ(n)=1-->d=23
  - PK={e,n}={7，187}， SK={d,n}={23，187}
- 加密：
  - $c=88^7 \bmod 187=11$
- 解密：
  - $p=11^{23} \bmod 187=88$

RSA的安全性

- RSA的安全性基于分解大整数的困难性假设
- 对RSA算法的攻击可能的方式
  - 穷举攻击
  - 数学攻击
  - 计时攻击
  - 选择密文攻击：
- 密钥长度应该介于1024bit到2048bit之间
- 来自两方面的威胁
  - 人类计算能力的不断提高
  - 分解算法的进一步改进

公钥加密存在的问题

- 需要比较长的密钥(>1024bit)
- 计算成本：加密速度比DES慢1000倍！
- 长明文的安全问题

公钥应用

- 对称加密和公钥加密各有优缺点
  - 公钥密码体制密钥共享便捷
  - 对称密码体制加密简单、高效
- 因此，在实际应用中，你会如何使用？
  - 混合使用两种密码体制
- 应用举例：场景描述Alice向Bob加密传输一个大文件，应该如何设计加密方案（含密钥的分发）

公钥应用

- 很少用于大量数据加密
- 数字签名
- 密钥的管理与认证

2.3.5 ElGamal密码体制

- 1984年由T.Elgamal提出
- ElGamal密码体制是一种基于离散对数的公钥密码体制
- 与Diffie-Hellman密钥分配体制密切相关
- ElGamal的安全性是基于计算离散对数的困难性之上的
  - 为了恢复Alice的私钥，攻击者需要计算$X_A=d\ \log_{a,q}(Y_A)$
  - 为了恢复K，攻击者需要选择随机数*k*，然后计算离散对数$k=d\ \log_{a,q}(C_1)$

ElGamal密码体制

- 数字签名标准DSS
  - 专门为数字签名功能而设计的算法
  - 不能用来加密或者进行密钥交换
  - 采用DSA算法，基于ElGamal密码算法

椭圆曲线公钥密码体制ECC

- **椭圆曲线密码（ECC）**由Neal Koblitz和Victor Miller于1985年发明。
- **椭圆曲线密码**为公钥密码机制，可以看作是椭圆曲线对先前基于离散对数问题的密码系统的模拟，只是群元素由元素数换为有限域上的椭圆曲线上的点。
- 安全性基于**椭圆曲线群上计算离散对数困难问题**
  - 给定椭圆曲线上的一个点P，一个整数*k*，求解Q=*k*P容易；
  - **给定点P、Q，知道Q=*k*P，求整数*k*困难**

椭圆曲线公钥密码体制ECC

- ECC特点
  - 可以用更短的密钥获得更高的安全性
  - 加密速度比RSA快
- 典型算法
  - ECDH: ECC算法和DH结合使用，用于密钥磋商
  - ECDSA: ECC算法和DSA结合使用，用于数字签名（比特币中使用）
- 应用场合：适用于处理能力、存储空间、带宽及功耗受限的场合，如移动互联网、无线手机加密等。

小结

	RSA	ElGamal	ECC
数论基础	欧拉定理	离散对数	离散对数
安全性基础	整数分解问题的困难性	有限域上离散对数问题的困难性	椭圆曲线离散对数问题的困难性
当前安全密钥长度(bit)	1024	1024	160
用途	加密、数字签名	加密、数字签名	加密、数字签名
是否申请专利	是	否	否
复杂性			设计困难，实现复杂

关于密码算法的一些事

1994年，全世界范围内同时使用了1600个工作站、耗时8个月时间才完成了129位数的分解。

如果要用同样的计算能力来分解250位数则要耗时

80万年

利用万亿次传统计算机分解300位的整数需要**150万年**

量子计算机则只要1秒钟即可解决。

关于密码算法的一些事

2019年9月23日晚，谷歌发表的一篇关于实现“**量子霸权**”（**量子优越性**）的论文Quantum supremacy using a programmable superconducting processor。这篇关于“量子霸权”的论文声称其开发出来拥有54个量子比特数组（其中53个功能正常）的量子计算机达到了量子优越性，他们用3分20秒运行一系列操作，传统超级计算机（世界上第一超算）大约需要花费10000年才能完成计。

量子霸权后给我们的加密算法造成多大的影响？

关于密码算法的一些事

- 量子计算机的出现不会给现代密码学带来毁灭性的打击
  - 对于公钥密码体制：只要数学难解问题在数学上未取得突破，依然可用
  - 对于分组密码和安全散列算法：需要重新设计算法，但依然是可以实现的

附1：模19的整数幂

- 通过19的整数幂，可观察到：
  - 所有序列均以1结束
  - 每一行均有整数个幂序列长
  - 有些序列长为18。这时称a生成了模19的非零整数集，并称这样的整数a为模19的本原根
- 对于素数p，正整数a若是p的本原根，则a的幂能够生成1~p-1的所有整数，即

$$a, a^2, a^3, \ldots, a^{p-1}$$

(模p)各不相同。

附1：模19的整数幂(续)

- 只有当a为p的本原根时，对于给定的Y，才存在唯一的以a为底模p的离散对数

以3为底，模为19的离散对数

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

以5为底，模为19的离散对数

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{5,19}(a)$	9 18			8 17	1 10	2 11	6 15		5 14		3 12						4 13	

仅当a为p的本原根时，以a为底模p的离散对数才唯一