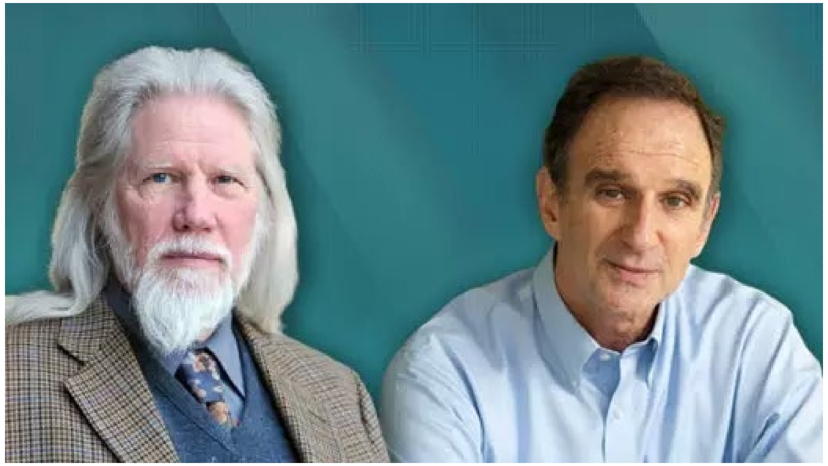
密 码 学

第五章 公钥密码算法

网络空间安全学院
朱丹 戚明平
zhudan/mpqi@nwpu.edu.cn

知识回顾-公钥密码的产生背景

- 20世纪70年代，斯坦福大学Diffie和Hellman研究了密钥分发问题，提出了一种通过公开信道共享密钥的方案，即Diffie-Hellman密钥交换协议
- 1976年，Diffie和Hellman发表了题为《密码学的新方向》（New Directions in Cryptography）的论文，论文提出了公钥密码的思想



美国计算机协会（ACM）将2015年的图灵奖授予Sun Microsystems的前首席安全官惠特菲尔德·迪菲（Whitfield Diffie）以及斯坦福大学电气工程系名誉教授马丁·赫尔曼（Martin Hellman），以表彰他们在现代密码学中所起的至关重要的作用。

知识回顾-公钥密码的理论模型

- 单向函数: 设函数$y = f(x)$，如果满足以下两个条件，则称为单向函数:
  - 如果对于给定的x，要计算出$y = f(x)$很容易
  - 而对于给定的y，要计算出$x = f^{-1}(y)$很难

单向函数是否
适于构造数据加密算法

- 利用单向函数构造加密函数
  - 用正变换作加密，加密效率高
  - 用逆变换作解密，安全，敌手不可破译

合法的消息接收者也无法解密

知识回顾-公钥密码的理论模型

- 单向陷门函数：设函数$y = f(\mathrm{x})$，且f具有陷门，若满足以下条件，则称为单向陷门函数：
  - 如果对于给定的x，要计算出$y = f(x)$很容易
  - 而对于给定的y，如果不掌握陷门要计算出$x = f^{-1}(y)$很难
  - 而如果掌握陷门要计算出$x = f^{-1}(y)$就很容易

- 利用单向陷门函数构造加密函数
  - 用正变换作加密，加密效率高
  - 用逆变换作解密，安全，敌手难以破译
  - 把陷门信息作为密钥，且只分配给合法用户。确保合法用户能够方便地解密，而非法用户不能破译

知识回顾-单向函数的研究现状

- 理论上：尚不能证明单向函数一定存在
- 实际上：密码学认为只要函数单向性足够应用即可

- ① 大合数的因子分解问题
- 大素数的乘积容易计算($p \times q = n$)，而大合数的因子分解困难($n = p \times q$)

- ② 有限域上的离散对数问题
- 有限域上大素数的幂乘容易计算($a^b = c$)，而对数计算困难($\log_a c = b$)

- ③ 椭圆曲线离散对数问题
- 设d是正整数，G是解点群的基点，计算$dG = Q$是容易的，而由Q求出d是困难的

知识回顾-RSA公钥密码算法

- 加解密算法
  - 随机地选择两个大素数p和q，而且保密
  - 计算$n = p * q$，将n公开
  - 计算$\varphi(n) = (p\text{-}1) * (q\text{-}1)$，对$\varphi(n)$保密
  - 随机地选取一个正整数e，$1 < e < \varphi(n)$且$(e, \varphi(n)) = 1$，将 e公开
  - 根据 $ed = 1 \bmod \varphi(n)$，求出d，并对d保密
  - 加密运算：$C = M^e \bmod n$
  - 解密运算：$M = C^d \bmod n$
  - 公开密钥$K_e = <e, n>$，保密密钥$K_d = <p, q, d, \varphi(n)>$

知识回顾-RSA公钥密码算法

- 在计算上由公开的加密钥不能求出解密钥
  - 公开密钥$K_e = \langle e, n \rangle$
  - 保密密钥$K_d = \langle p, q, d, \varphi(n) \rangle$
  - 假设攻击者截获密文C，需恢复明文

大合数的因子分解问题

$$n = pq$$

$$\varphi(n) = (p - 1)(q - 1)$$

$$ed \equiv 1 \bmod \varphi(n)$$

$$M \equiv C^d \bmod n$$

$$C$$

知识回顾-RSA算法参数的选取

- (1) p和q要足够大：p和q应至少为512位，使n达1024位
- (2) p和q要是强素数： (p-1)、(p+1)、 (q-1)、(q+1)均有大素数因子
- (3) p和q位数相差不能太小，也不能太大
- (4) 在给定的条件下，找到的$d = e$，这样的密钥必须舍弃
- (5) 如果使得$e^i = 1 \bmod \varphi(n)$成立的i值很小，则很容易进行密文迭代攻击
- (6)(7) e和d的选择：兼顾安全性和实现效率

知识回顾-RSA算法的实现

- 加解密运算
  - 加密运算：$C = M^e \bmod n$
  - 解密运算：$M = C^d \bmod n$
- 模幂运算算法
  - 模幂运算基本定义
  - 重复平方算法
  - 滑动窗口算法
  - CRT算法
  - 蒙哥马利算法（了解）

章节安排

Outline

- Diffie-Hellman密钥交换协议
- ElGamal公钥密码算法
- ECC公钥密码算法
- SM2公钥密码算法

章节安排

Outline

- Diffie-Hellman密钥交换协议
- ElGamal公钥密码算法
- ECC公钥密码算法
- SM2公钥密码算法

5.8(1) 离散对数问题

- 离散对数问题（Discrete Logarithm Problem, DLP）
  - 设p为素数，则模p的剩余构成有限域

$$F_p = GF(p) = \{0,1,2,\dots\dots,p-1\}$$

  F_p的非零元素构成乘法循环群$F_p^* = \{1,2,\dots\dots,p-1\} = \{a,a^2,\dots,a^{p-1}\}$，

  则称a是F_p^*的生成元或模p的本原元。

  - 求a的模幂运算为$y = a^x \ mod \ p, 1 \le x \le p-1$，求对数$x$的运算为$x = \log_a y, 1 \le x \le p-1$，由于上述运算是定义在有限域$F_p$上的，所以称为离散对数运算。
  - 从x计算y是容易的。从y计算x就困难得多，利用目前最好的算法，对于小心选择的p将至少需用$O(p^{\frac{1}{2}})$次以上的运算，只要p足够大，求解离散对数问题是相当困难的。

离散对数问题（Discrete Logarithm Problem, DLP）

- 取$p = 13$，则$a = 2$是模p的本原元

	1	2	3	4	5	6	7	8	9	10	11	12
$y = a^x \bmod p$	2	4	8	3	6	12	11	9	5	10	7	1

	1	2	3	4	5	6	7	8	9	10	11	12
$x = \log_a y \bmod p$	12	1	4	2	9	5	11	3	8	10	7	6

5.8(1) 离散对数问题

- 离散对数问题（Discrete Logarithm Problem, DLP）
  - 离散对数具有良好的单向性，在公钥密码中得到了广泛的应用，如
    - Diffie-Hellman密钥交换协议
    - ElGamal公钥密码算法
    - 美国数字签名算法（DSA, Digital Signature Algorithm）
    - ElGamal公钥密码算法改进了Diffie-Hellman密钥交换协议，提出了基于离散对数的公钥密码和数字签名体制

5.8(2) Diffie-Hellman密钥交换协议

参数选取：

选取大素数p，再选取Z_p的本原元a，并将p和a公开，全网公用。

密钥协商：

$1 \le x_A \le p-2$

$1 \le x_B \le p-2$

Alice

x_A

$$y_A = a^{x_A} \bmod p$$

$$y_B = a^{x_B} \bmod p$$

Bob

x_B

$$k = (y_B)^{x_A} \quad \boxed{= a^{x_A x_B} =} \quad k = (y_A)^{x_B}$$

将k作为Alice和Bob协商出来的密钥，同时不再保留x_A和x_B；攻击者如果想要获得x_A或者x_B，必须解决DLP问题

5.8(2) Diffie-Hellman密钥交换协议

安全性：

- 有限域上的离散对数问题是密码学中的困难问题，目前没有快速求解有限域上离散对数的数学方法。最快算法的复杂度为亚指数级别，当p为200位时，求解需要2~3天；而当p为664位时，求解约需2.7亿年。只要p足够大，D-H密钥交换协议就可以达到足够安全。

优点：

- 任何两人都可以协商出会话密钥
- 降低通信双方的密钥管理负担

缺点：

- 易遭受中间人攻击

5.8(3) 中间人攻击

Alice  x_A

$$y_A = a^{x_A} \bmod p$$

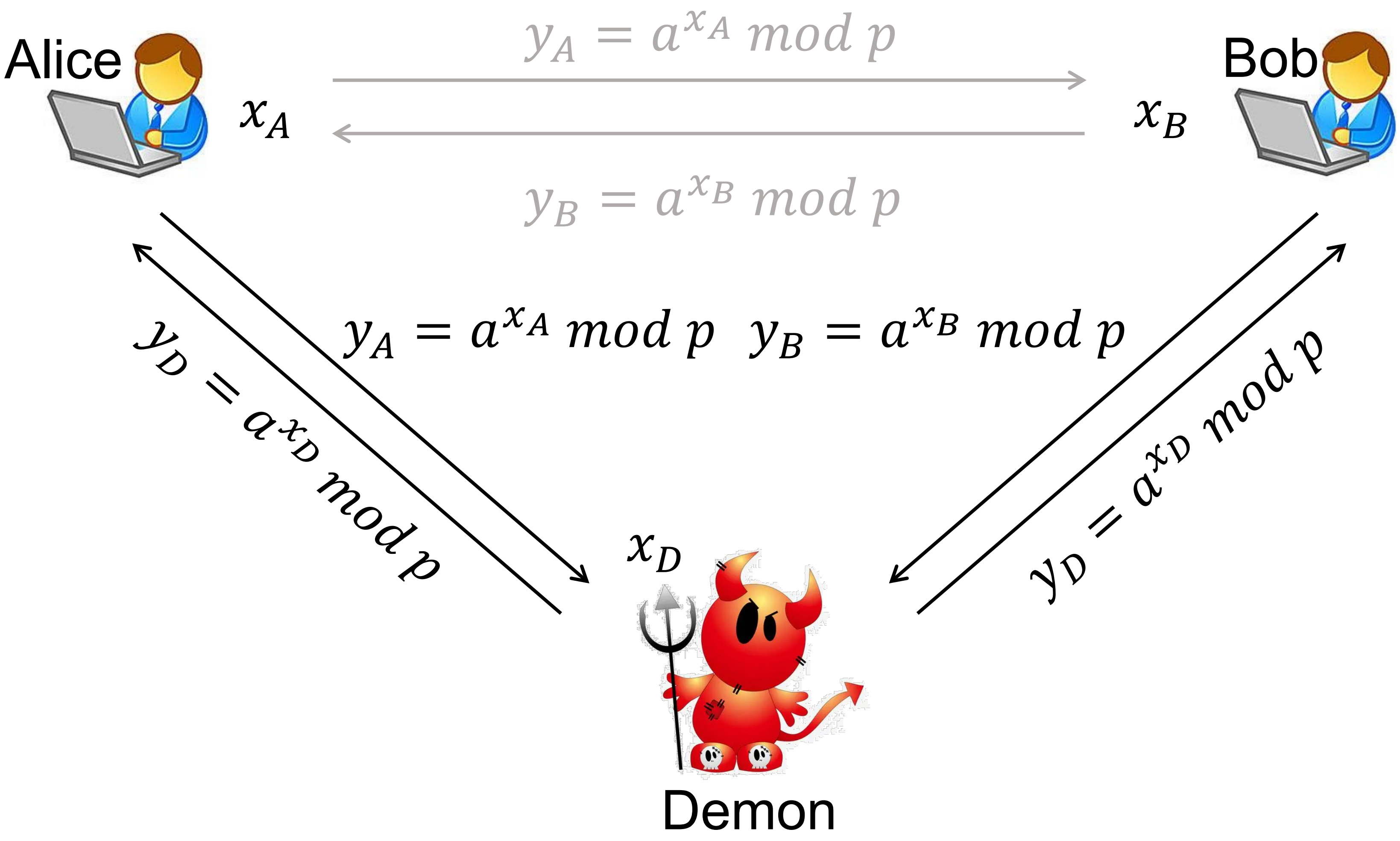
$$\longrightarrow$$

$$\longleftarrow$$

$$y_B = a^{x_B} \bmod p$$

Bob  x_B

5.8(3) 中间人攻击


Alice
Bob
x_A
x_B
$y_A = a^{x_A} \bmod p$
$y_B = a^{x_B} \bmod p$
$y_A = a^{x_A} \bmod p$
$y_B = a^{x_B} \bmod p$
$y_D = a^{x_D} \bmod p$
$y_D = a^{x_D} \bmod p$
x_D
Demon

5.8(3) 中间人攻击

Alice

Bob

$y_A = a^{x_A} \, mod \, p$

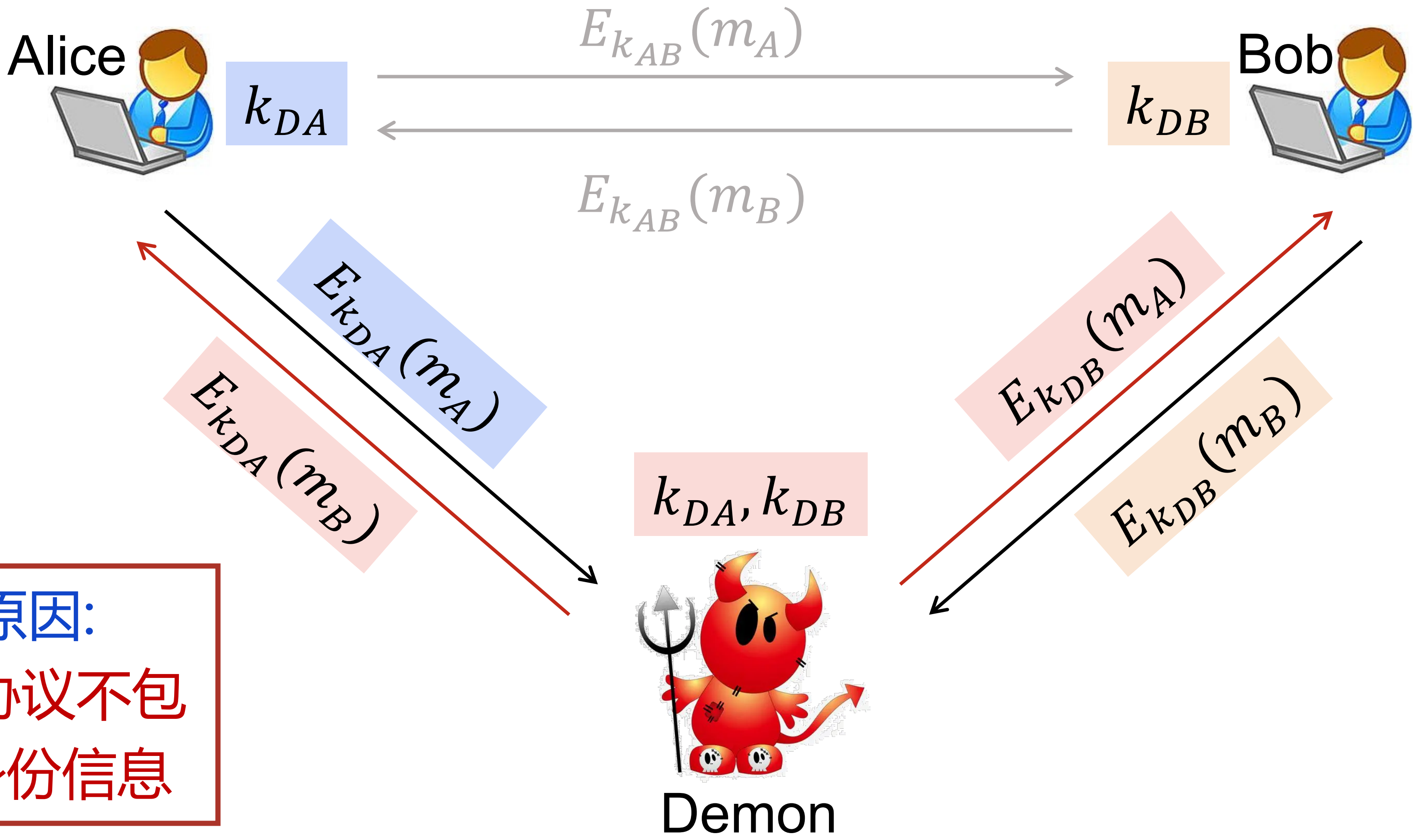
$y_B = a^{x_B} \, mod \, p$

$k_{DA} = (y_D)^{x_A} = (y_A)^{x_D} = a^{x_D x_A} \, mod \, p$

$k_{DB} = (y_D)^{x_B} = (y_B)^{x_D} = a^{x_D x_B} \, mod \, p$

Demon

5.8(3) 中间人攻击



攻击原因:
DH交换协议不包
含任何身份信息

身份认证:
公钥证书
数字签名

章节安排

Outline

- Diffie-Hellman密钥交换协议
- **ElGamal公钥密码算法**
- ECC公钥密码算法
- SM2公钥密码算法

5.9 ElGamal公钥密码算法

参数选取：

随机地选择一个大素数p，且要求$p-1$有大素数因子。再选择一个模p的本原元a。

将p和a公开作为算法的基础参数。

密钥生成：

- 用户随机地选择一个整数d作为自己保密的解密密钥k_d，$2 \le d \le p-2$;
- 用户计算$y = a^d \ mod\ p$，并取y作为自己公开的加密密钥k_e。

显然，由公开密钥y计算解密密钥d，必须计算离散对数，这是极困难的。

5.9 ElGamal公钥密码算法

加密算法：

将明文消息$M(0 \le M \le p-1)$加密成密文的过程如下：

- 随机地选择一个整数k，$2 \le k \le p-2$；
- 计算：

$$U = y^k \bmod p$$

$$C_1 = a^k \bmod p$$

$$C_2 = UM \bmod p$$

- 取$C = (C_1, C_2)$作为密文。

5.9 ElGamal公钥密码算法

解密算法：

将密文消息(C_1, C_2)解密成明文的过程如下：

- 计算：

$$V = C_1^d \ mod\ p$$

- 计算：

$$M = C_2V^{-1} \ mod\ p$$

- 获得明文M。

5.9 ElGamal公钥密码算法

- 解密的可还原性证明如下

$$
\begin{aligned}
C_2V^{-1}\ mod\ p &= (UM)V^{-1}\ mod\ p \\
&= UM(C_1^d)^{-1}\ mod\ p \\
&= UM(\left(a^k\right)^d)^{-1}\ mod\ p \\
&= UM(\left(a^k\right)^d)^{-1}\ mod\ p \\
&= UM(\left(a^d\right)^k)^{-1}\ mod\ p \\
&= UM((y)^k)^{-1}\ mod\ p \\
&= UM(U)^{-1}\ mod\ p \\
&= M\ mod\ p
\end{aligned}
$$

5.9 ElGamal公钥密码算法

- ElGamal公钥密码算法的安全性
  - 由于ElGamal公钥密码的安全性建立在$GF(p)$离散对数的困难性之上，目前尚无求解$GF(p)$离散对数有效算法，所以在p足够大时Elgamal密码是安全的。
  - 为了安全p应为150位以上的十进制数，而且$p-1$应有大素因子。
  - d和k都不能太小，应为高质量的随机数。
  - 为了安全加密和签名所使用的k必须是一次性的。

ElGamal公钥密码算法的安全性

- 如果k不是一次性的，时间长了就可能被攻击者获得。又因y是公开密钥，攻击者自然知道。于是攻击者就可以根据$U = y^k \bmod p$ 计算出U，进而利用Euclid算法求出U^{-1}。又因为攻击者可以获得密文C_2，于是根据$C_2 = UM \bmod p$通过计算$U^{-1}C_2$得到明文M。

- 设用同一个k加密两个不同的密文M和M'，相应的密文为(C_1, C_2)和(C_1', C_2')。因为$C_2/C_2' = M/M'$，如果攻击者知道M，就很容易求出M'。

5.9 ElGamal公钥密码算法

- ElGamal公钥密码算法的应用
  - 由于ElGamal密码的安全性得到世界公认，所以得到了广泛应用。
    - 著名的美国数字签名标准DSS，采用了ElGamal密码的一种变形。
    - 电子邮件标准S/MIME采用了ElGamal密码。
    - 俄罗斯的数字签名标准也是ElGamal密码的一种变形，而且数据规模选得更大。
  - 为了适应不同的应用，人们在应用中总结出18种不同的ElGamal密码的变形。

5.9 ElGamal公钥密码算法

ElGamal公钥密码算法的应用

加解密速度快

- 由于实际应用时ElGamal密码运算的素数p比RSA要小，所以ElGamal密码的加解密速度比RSA快。

随机数源

- 由ElGamal密码的解密密钥d和随机数k都应是高质量的随机数。因此，应用ElGamal密码需要一个好的随机数源，也就是说能够快速地产生高质量的随机数。就是说能够快速地产生高质量的随机数。

大素数的选择

- 为了ElGamal密码的安全，p应为150位以上的十进制数，而且p−1应有大素因子。

章节安排

Outline

5.10(1) 椭圆曲线离散对数问题

素数域上的椭圆曲线问题

- 设p是大于3的素数，且$4a^3 + 27b^2 \neq 0 \ mod\ p$，称

$$y^2 = x^3 + ax + b$$

$a, b \in GF(p)$为$GF(p)$ 上的椭圆曲线。

- 由椭圆曲线可得到一个同余方程：

$$y^2 = x^3 + ax + b \ mod\ p$$

其解为一个二元组$< x, y >, x, y \in GF(p)$，将此二元组描画到椭圆曲线上便为一个点，称其为一个解点。

- 例如，(2,4)是椭圆曲线$y^2 = x^3 + x + 6 \ mod\ 11$的一个解点。

素数域上的椭圆曲线问题

为了利用解点构成交换群，需要引进一个0元素，并定义如下的加法运算：

- **定义单位元：** 引进一个无穷点$O(\infty,\infty)$，简记为O，作为0元素

$$O(\infty,\infty)+O(\infty,\infty)=0+0=0$$

并定义对于所有的解点$P(x,y)$，

$$P(x,y)+O=O+P(x,y)=P(x,y)$$

- **定义逆元素：** 设$P(x_1,y_1)$和$Q(x_2,y_2)$是解点，如果$x_1=x_2$且$y_1=-y_2$，令

$$P(x_1,y_1)+Q(x_2,y_2)=O$$

这说明**任何解点$R(x,y)$的逆就是$R(x,-y)$**。**规定无穷远点的逆是本身**：

$$O(\infty,\infty)\ =\ -O(\infty,\infty)$$

5.10(1) 椭圆曲线离散对数问题

素数域上的椭圆曲线问题

为了利用解点构成交换群，需要引进一个0元素，并定义如下的加法运算：

- 定义加法：设$P(x_1,y_1) \neq Q(x_2,y_2)$，且$P$和$Q$不互逆，则

$$P(x_1,y_1)+Q(x_2,y_2) = R(x_3,y_3)$$

其中

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \\ \lambda = \dfrac{(y_2 - y_1)}{(x_2 - x_1)} \end{cases}$$

素数域上的椭圆曲线问题

为了利用解点构成交换群，需要引进一个0元素，并定义如下的加法运算：

- 定义加法：设$P(x_1,y_1)=Q(x_2,y_2)$，则

$$P(x_1,y_1)+Q(x_2,y_2)=2P(x_1,y_1)=R(x_3,y_3)$$

其中

$$\begin{cases} x_3=\lambda^2-2x_1 \\ y_3=\lambda(x_1-x_3)-y_1 \\ \lambda=\dfrac{(3x_1^2+a)}{2y_1} \end{cases}$$

5.10(1) 椭圆曲线离散对数问题

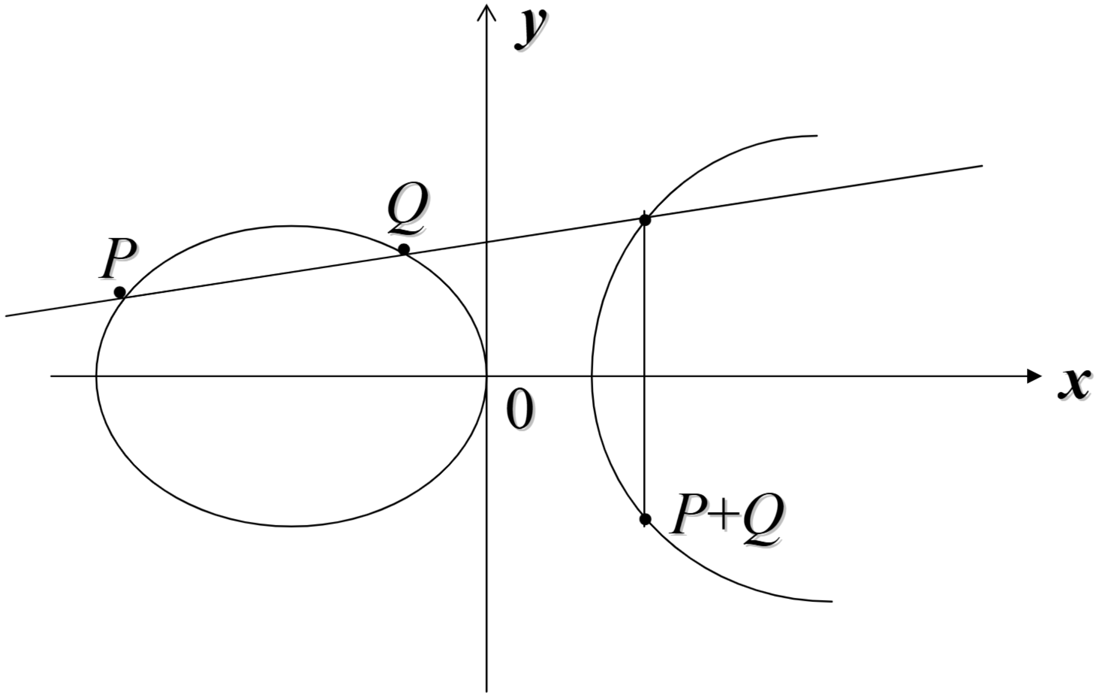
素数域上的椭圆曲线问题

- 作集合$E=\{$全体解点，无穷点$O\}$
- 可以验证，如上定义的集合E和加法运算构成加法交换群
- 复习：群G的定义
  - G是一个是一个非空集
  - 定义了一种运算，且运算是自封闭的，运算满足结合律
  - G中有单位元
  - G中的元素都有逆元
  - 若满足交换律：$a·b=b·a$，则构成阿贝尔群，也叫交换群

5.10(1) 椭圆曲线离散对数问题

椭圆曲线解点加法运算的几何意义

- 设$P(x_1,y_1)$和$Q(x_2,y_2)$是椭圆曲线上的两个点，则连接$P(x_1,y_1)$和$Q(x_2,y_2)$的直线与椭圆曲线的另一交点关于横轴的对称点即为$P(x_1,y_1)+Q(x_2,y_2)$点



椭圆曲线离散对数问题

- 例，求出椭圆曲线$y^2 = x^3 + x + 6 \; mod \; 11$的解点
- 由于p较小，使$GF(p)$也较小，故利用穷举的方法根据$y^2 = x^3 + x + 6 \; mod \; 11$可以求出所有解点
- 复习：平方剩余

设p为素数，如果存在一个正整数y，使得

$$y^2 = a \; mod \; p$$

则称a是模p的平方剩余。

椭圆曲线离散对数问题

- 根据表可知解点(x, y)集为：(2, 4)，(2, 7)，(3, 5)，(3, 6)，(5, 2)，(5, 9)，(7, 2)， (7, 9)，(8, 3)， (8, 8), (10, 2)，(10, 9)。再加上无穷远点O，共13个点构成一个加法交换群

- 由于群的元素个数为13，为素数，此群是循环群，而且任何一个非0元素都是生成元

x	$x^3+x+6 \bmod 11$	是否是模11平方剩余	y
0	6	No	-
1	8	No	-
2	5	Yes	4, 7
3	3	Yes	5, 6
4	8	No	-
5	4	Yes	2, 9
6	8	No	-
7	4	Yes	2, 9
8	9	Yes	3, 8
9	7	No	-
10	4	Yes	2, 9

椭圆曲线离散对数问题

- 由于是加法群，n个元素G相加表示为：

$$G + G + \cdots + G = nG$$

称为倍点运算

- 我们取$G = (2,7)$为生成元，2倍点计算如下：

$$2G = (2,7) + (2,7) = (5,2)$$

因为$\lambda = \left(3 \times 2^2 + 1\right)(2 \times 7)^{-1}\, mod\ 11 = 2 \times 3^{-1}\, mod\ 11 = 2 \times 4\, mod\ 11 = 8$

于是，$x_3 = 8^2 - 2 \times 2\ mod\ 11 = 5$，$y_3 = 8 \times (2 - 5) - 7\ mod\ 11 = 2$

椭圆曲线离散对数问题

G	$2G$	$3G$	$4G$	$5G$	$6G$	$7G$
$(2,7)$	$(5,2)$	$(8,3)$	$(10,2)$	$(3,6)$	$(7,9)$	$(7,2)$

$8G$	$9G$	$10G$	$11G$	$12G$	$13G$
$(3,5)$	$(10,9)$	$(8,8)$	$(5,9)$	$(2,4)$	$O(\infty,\infty)$

- 上例中，p较小，使得$GF(p)$也较小，故可以利用穷举法求出所有解点。但是，对于一般情况要计算椭圆曲线解点数N的准确值比较困难

- N满足不等式 $p+1-2p^{1/2} \leq N \leq p+1+2p^{1/2}$

椭圆曲线离散对数问题

- 在上例中椭圆曲线上的解点所构成的交换群恰好是循环群，但是一般并不一定。于是我们希望从中找出一个循环子群E_1
- 现有研究已经证明：当循环子群E_1的阶n是足够大的素数时，这个循环子群中的离散对数问题是困难的
- 除了$GF(p)$上的椭圆曲线，还有定义在$GF(2^m)$上的椭圆曲线。基于这两种椭圆曲线都可以设计出安全的椭圆曲线密码

椭圆曲线离散对数问题

设P和Q是椭圆曲线上的两个解点，t为一正整数，且$1 \le t < n$

- 对于给定的P和t，计算$tP = Q$是容易的
- 但若已知P和Q点，要计算出t则是极困难的。这便是椭圆曲线群上的离散对数问题，简记为ECDLP (Elliptic Curve Discrete Logarithm Problem)
- 除了几类特殊的椭圆曲线外，对于一般ECDLP目前尚没有找到有效的求解方法。因子分解和DLP问题都有亚指数求解算法，而ECDLP尚没有发现亚指数求解算法
- 于是可以在这个循环子群E_1中建立任何基于离散对数困难性的密码，并称这个密码为椭圆曲线密码


感谢聆听!
THANK YOU FOR YOUR ATTENTION!