# 第8章 椭圆曲线算法

浙江大學
ZHEJIANG UNIVERSITY

## Diffie-Hellman密钥交换

- Diffie和Hellman在1976年提出公钥概念的阐述以及第一个公钥类型方案
  - 注：现在得知早在1970年的秘密研究中， Ellis、Williamson、Cocks等人就已提出了这个概念
- 是公开交换密钥的一种实用方法
- 用于许多商业产品中
- 基于有限（伽罗瓦）域（模素数或多项式）中的幂运算-简单
- 基于计算离散对数（类似于因子分解）的安全性-困难

## Diffie-Hellman 设置

- 用户都就全局参数达成一致：
  - 大素数整数或多项式，q
  - q的本原根（primitive root），a
- 每个用户（例如A）生成key：
  - 选择一个密钥（数字）：$x_A<q$
  - 计算公钥：$y_A=a^{x_A} \bmod q$
- 每个用户都会公开该密钥$y_A$

## Diffie-Hellman密钥交换

- 用户A和B的共享会话密钥为$K_{AB}$:
- $K_{AB} = a^{xA.\,xB} \bmod q$

  $= y_A^{xB} \bmod q$ （B可以计算）

  $= y_B^{xA} \bmod q$ （A可以计算）
- 在Alice和Bob之间的私钥加密方案中，$K_{AB}$被用作会话密钥(session key)
  - 如果Alice和Bob随后进行通信，除非他们选择新的公钥，否则他们将拥有与之前相同的密钥
  - 攻击者需要一个x，必须解决离散对数问题

## Diffie-Hellman 举例

希望交换密钥的用户Alice和Bob:

- 设置素数q=353和a=3
- 选择随机密钥:
  - A选择$x_A$ =97, B选择$x_B$ =233
- 计算各自的公钥:
  - $y_A = 3^{97} mod353=40$
  - $y_B = 3^{233} mod353=248$
- 计算共享会话密钥:
- $K_{AB} = y_B^{xA} \bmod q = 248^{97} \bmod 353 = 160$
- $K_{AB} = y_A^{xB} \bmod q = 40^{233} \bmod 353 = 160$

## 密钥交换协议

- 用户可以在每次通信时创建随机的私有/公共D-H密钥
- 用户可以创建一个已知的私有/公共D-H密钥并发布到一个目录中，然后每次访问并使用它与他们进行安全通信
- 这两者都容易受到中间相遇攻击（ Meet-in-the-middle attack ）
- 需要对密钥进行身份验证

## 椭圆曲线算法

- 大多数公钥密码（RSA，D-H）都使用整数或多项式算法来处理非常大的数字/多项式。
- 大多数使用公钥加密技术进行加密和数字签名的产品和标准都使用RSA。安全RSA使用的密钥长度近年来有所增加，在存储和处理密钥和消息时给应用程序带来了更大的处理负载。
- 与RSA相比，椭圆曲线算法的主要吸引力在于，它似乎能以更小的密钥大小提供同等的安全性，从而减少处理开销。

广义ECC

- ECDLP based cryptosystem(传统的基于离散对数困难的密码系统)

- Pairing based cryptosystem(基于双线性对的密码体制)

- Isogeny based cryptosystem(基于同源的密码体制)

## 椭圆曲线算法

- 椭圆曲线不是椭圆，之所以这样命名，是因为它们由三次方程式描述，类似于用于计算椭圆周长的方程式。
- 通常，椭圆曲线的三次方程称为Weierstrass方程：

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

- 就算法目的而言，局限于这种形式的方程就足够了：

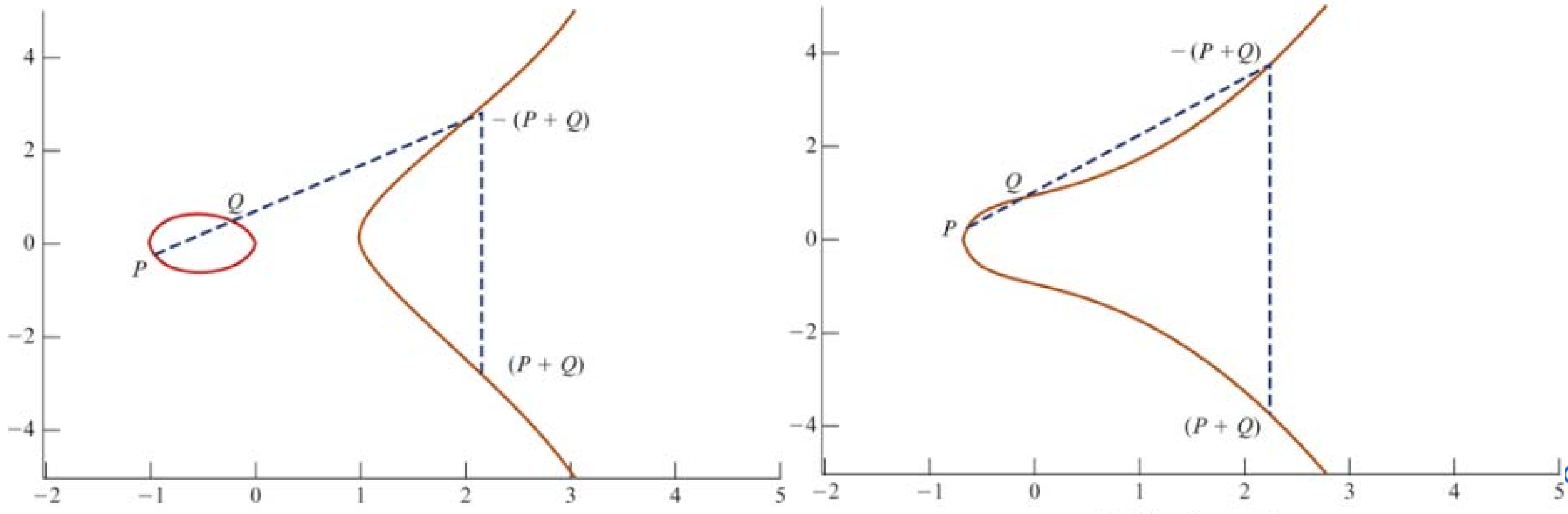
$$\boxed{y^2 = x^3 + ax + b}$$

## 有限椭圆曲线

- 椭圆曲线密码使用变量和系数为有限的曲线
- 有两个常用的族：
- $Z_p$上定义的素数曲线$E_p$（a，b）
  - 使用模素数的整数
  - 适用于软件
- GF（$2^m$）上定义的二元曲线$E_2m$（a，b）
  - 使用具有二进制系数的多项式
  - 适用于硬件

## 椭圆曲线算法（ECC算法）

- 椭圆曲线(Elliptic Curve) 可以定义成所有满足方程 E: $y^2 = x^3 + ax + b$ 的点(x,y)所构成的集合。
- 若 $x^3 + ax + b$ 没有重复的因式或 $4a^3 + 27b^2 \neq 0$ （称为判别式），则 E: $y^2 = x^3 + ax + b$ 能定义成为一个群。




(a) $y^2 = x^3 - x$

(b) $y^2 = x^3 + x + 1$

## 椭圆曲线算法（ECC算法）

- ECC加法(addition，点加)类似于模乘运算
- ECC重复加法(doubling，倍点)类似于模幂运算

- 椭圆曲线对数问题：
- Q=kP，其中Q，P属于素数曲线
  - 在给定k，P的情况下，计算Q比较“容易”
  - 但是在给定Q，P的情况下，很难找到k

## ECC算法的数学基础 - 椭圆曲线在素域$Z_p$上的运算规则

(1) P+0=0+P=P

(2) 如果$P=(x_1,y_1)$，$Q=(x_2, y_2)$，且有$x_1=x_2$及$y_1=y_2=0$，或有$x_1=x_2$及$y_1=-y_2\neq 0$，则P+Q=0；

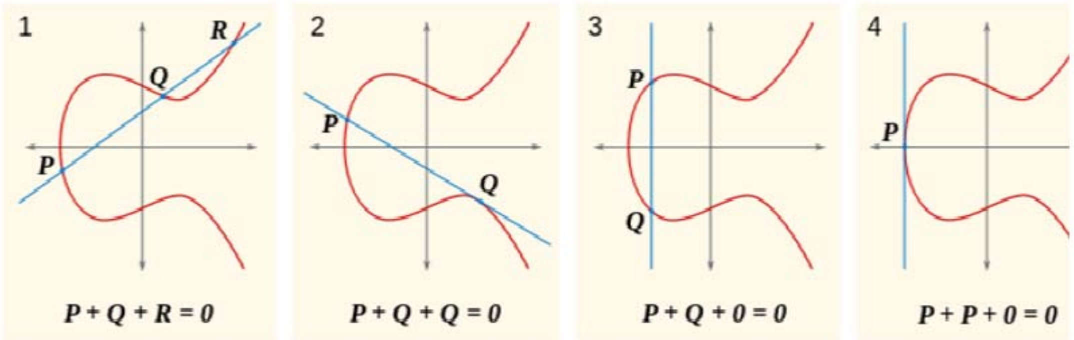
(3) 如果$P=(x_1,y_1)$，$Q=(x_2, y_2)$，且排除(1)(2)，则$P+Q=(x_3,y_3)$由下列规则决定：

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1-x_3) - y_1$$

当$P\neq Q$时，$\lambda=(y_2-y_1)/(x_2-x_1)$；

当P=Q时，$\lambda=(3x_1^2+a)/(2y_1)$；



## ECC算法的数学基础 - 点加法运算的性质

- (封闭性) 对任意$P \in E$和 $Q \in E$, $P + Q \in E$
- (结合律) 对任意$P \in E$, $Q \in E$以及$R \in E$, $(P + Q) + R = P + (Q + R)$
- (单位元) 对任意$P \in E$, $P + O = O + P = P$
- (负元素) 对任意$P \in E$, 存在$Q \in E$, 满足$P + Q = Q + P = O$
- (交换律) 对任意$P \in E$ 和$Q \in E$, $P + Q = Q + P$

## ECC算法的数学基础 - Euler准则

- $y^2 = x \bmod p$
- 设p>2是一个素数，x是一个整数，gcd(x,p)=1，则
- (1) x是模p的平方剩余当且仅当
  - $x^{(p-1)/2} \equiv 1 \pmod p$
- (2) x是模p的平方非剩余当且仅当
  - $x^{(p-1)/2} \equiv -1 \pmod p$

## ECC算法的数学基础 - 点加运算

- 椭圆曲线$y^2 = x^3 + x + 6 \pmod{11}$上的点。

  a=1　b=6　p=11

  基点G　G的阶　余因子

- 以上6项决定一条椭圆曲线
- 其中余因子=曲线的阶即曲线上点的个数/G的阶，此值通常=1

- 可以把α称为生成元(generator)，也称作基点(base point)，假定nα=0，则n称为α的阶(order)。
- 曲线的阶是曲线上点的个数。
- 曲线上的点(x,y)一定满足条件0<= x,y <p，并且x,y一定是整数。

## ECC算法的数学基础 - 点加运算举例

- Q = k*P　其中k<n
- 已知P及Q的情况下，求k很困难。
- 设α=(2,7)，计算2α=α+α：
- $\lambda=(3x_1^2+a)/(2y_1) = (3*2^2+1)/(2*7) =13/14 =13*14^{-1} =2*3^{-1}=2*4=8 \bmod 11$
- $x_3 = \lambda^2 - x_1 - x_2 = 8^2-2-2 = 60 = 5 \bmod 11$
- $y_3 = \lambda(x_1-x_3) - y_1 = 8*(2-5)-7 = 8*8-7=64+4=2 \bmod 11$
- 因此2α=(5,2)

## ECC算法的数学基础 - 点加运算举例

- No.01=(02, 07)
- No.02=(05, 02)
- No.03=(08, 03)
- No.04=(0A, 02)
- No.05=(03, 06)
- No.06=(07, 09)
- No.07=(07, 02)
- No.08=(03, 05)
- No.09=(0A, 09)
- No.10=(08, 08)
- No.11=(05, 09)
- No.12=(02, 04)
- No.13=infinity

## 用ECC算法加密解密 -公钥及私钥

- 公钥点R=d*G
- 私钥d是一个随机数，且d<n，其中n是G的阶

## 用ECC算法加密解密 - 加密

- r = (k*G).x ；其中k是一个随机数且k<n，r不可以mod n
- s = m*(k*R).x mod n；其中m是明文
- 密文包括两部分:r,s

## 用ECC算法加密解密 – 解密

- 红色的r是一个点，r=k*G
- m = s/(dr).x = m*(k*R).x/(d*(k*G)).x = m*(k*d*G).x/(k*d*G).x = m

## 用ECC算法加密解密 – 举例

- 设曲线方程是$y^2=x^3+x+6$ (mod 11)
- 基点G=(2,7)，G的阶n=13
- 设私钥d=7，则公钥R=dG=7*(2,7)=(7,2)
- 设随机数k=6，明文m=9，则
- 密文第1部分r=kG=6*(2,7)=(7,9)
- 密文第2部分s=m*(k*R).x=9*(6*(7,2)).x=9*(8,3).x=9*8 mod 13 = 7
- m=s/(dr).x= 7/(7*(7,9)).x =7/(8,3).x = 7/8 = $7*8^{-1}$ mod 13 = 7*5 mod 13 = 9