公开（非对称）密码算法之椭圆曲线密码体制

范明钰
信息安全研究中心

要点

- 非对称算法安全强度对比
- 改进：ECC体制
- ECC计算
- 例子

非对称算法安全强度

- RSA，和ElGamal密码系统最为人诟病的，就是在加解密，或是认证时候**庞大的运算量**
- 而计算能力的提高，又迫使**密钥长度不断增加**
- 椭圆曲线密码系统(*Elliptic Curve Cryptosystem, ECC*)，可以达到同样安全程度，但密钥位数要少得多

安全强度对比

- *RSA*算法是建立在大整数分解问题基础之上
- *ElGamal* 算法是基于有限域乘法群上的离散对数问题：
- 这两类问题的最好破解法是亚指数时间的：

$$O(\exp((c+o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}),$$

- *ECC*建立在椭圆曲线离散对数问题基础之上，一般情况下，只有**指数时间解法**。

椭圆曲线公钥密码体制——概述

- 1985年，*Miller*和*Koblitz*分别提出，用椭圆曲线加法群，来构造公钥密码体制
- 多年来，ECC一直是密码学研究的一个热点，并日渐成熟。已有多个信息处理标准推荐使用ECC，如*ANSI X9.63*，*P1363A*等
- 使用ECC可以实现数字签名与认证、密钥协商和信息加密三大功能

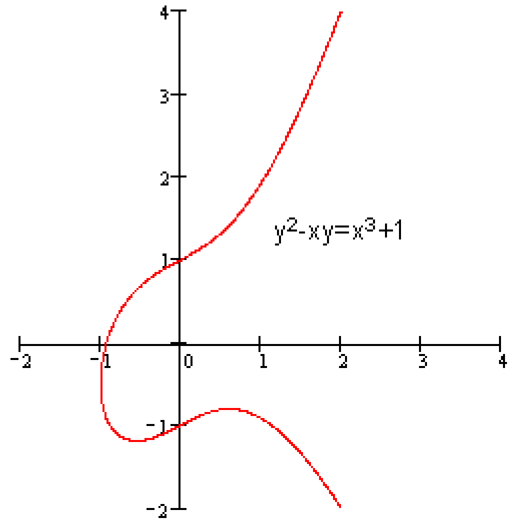
椭圆曲线

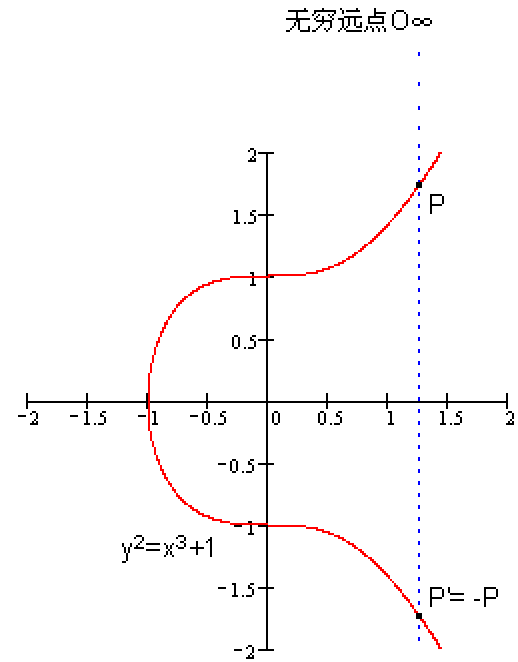
- 一般，椭圆曲线是满足

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

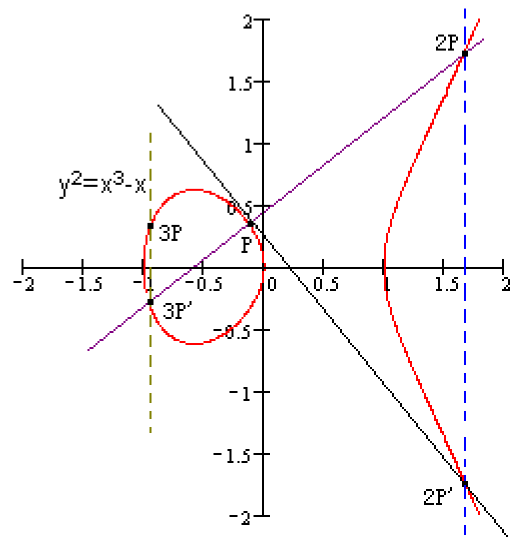
- 的坐标点的集合。
- 为了使用椭圆曲线进行运算，需要定义某种运算，这种运算对集合中两个点进行操作所得的结果，也在该集合中

实数域上的椭圆曲线





- 一般简化形式
  - $y^2=x^3+ax+b$
- $x^3+ax+b=0$ **没有重根**的条件
  - $4a^3+27b^2 \neq 0$



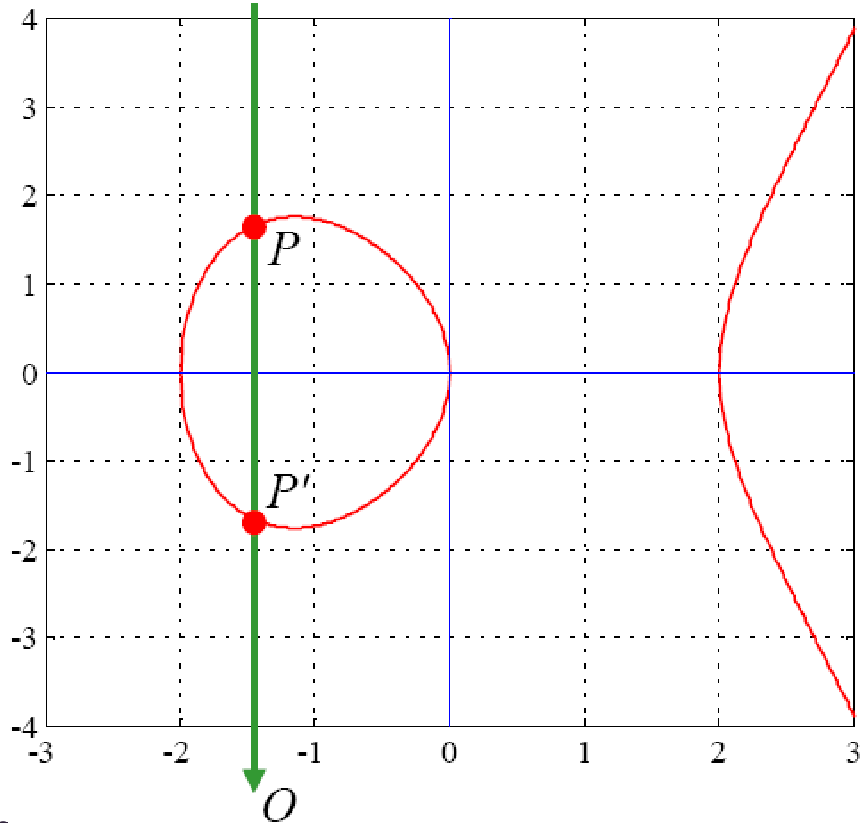
- 椭圆曲线的形状，并不是椭圆形状的
- 例子：

定义加法

- 椭圆曲线上的点集及其上的加法规则，构成一个群
  - 点集：椭圆曲线上的所有点，和无穷远点（单位元）
- 加法规则：若椭圆曲线上的三个点处于一条直线上，则它们的和为单位元O
  - 定义O是加法的单位元(additive identity)：O=-O；对于椭圆曲线上的任一点P，有P＋O ＝ P

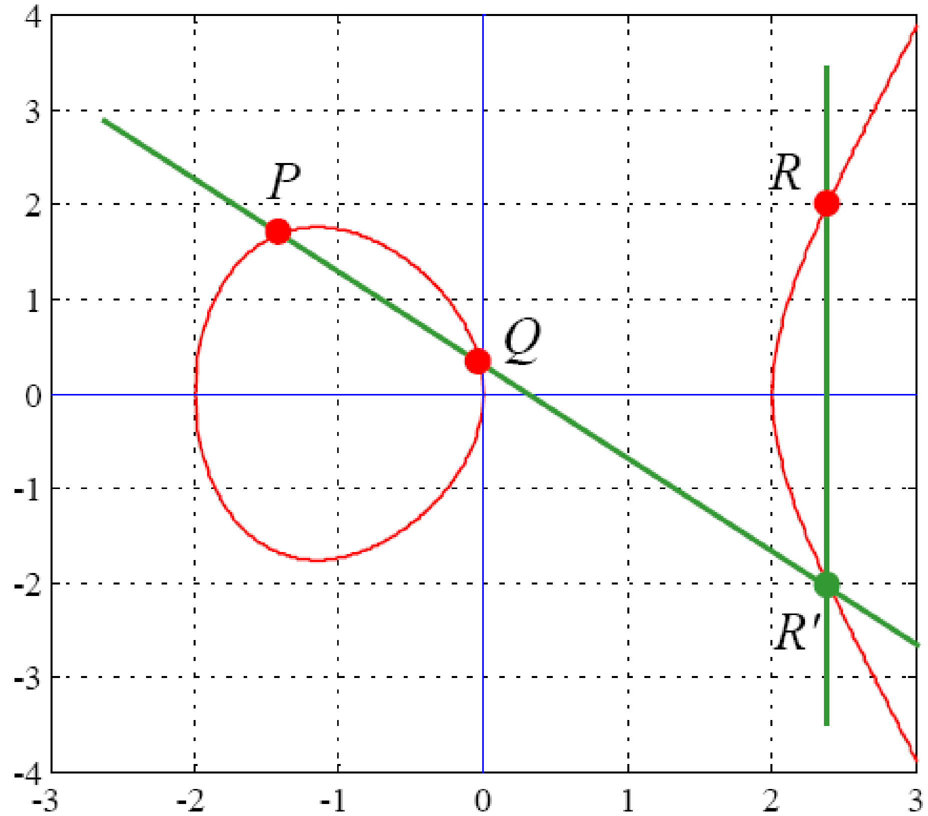
加法的逆元和单位元

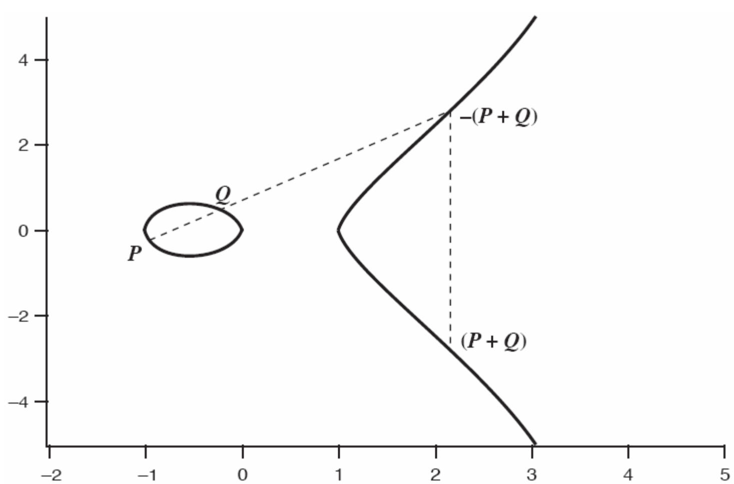
- 逆元：
  - 一条垂直线与曲线相交于P=(x,y)和P'=(x,-y)，也相交于无穷点O，有P+P'+O=O。即P= - P'
- 单位元：
  - P+O=P

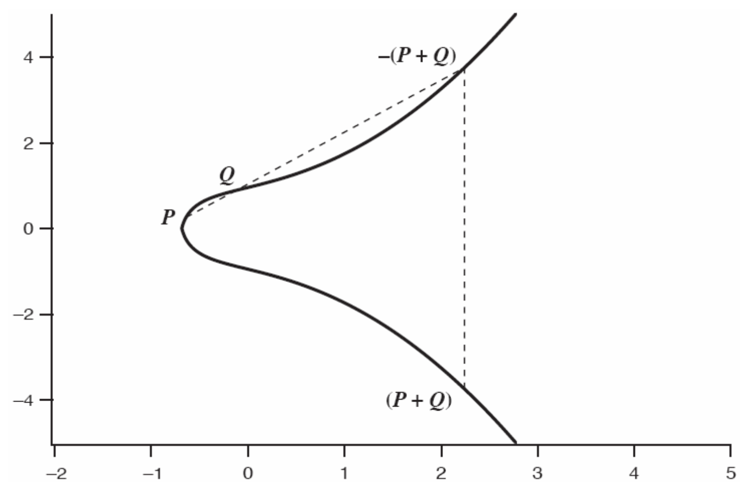


定义加法运算

◆ 连PQ做直线，得交点R'
  - P+Q+R'=O
  - P+Q=-R'

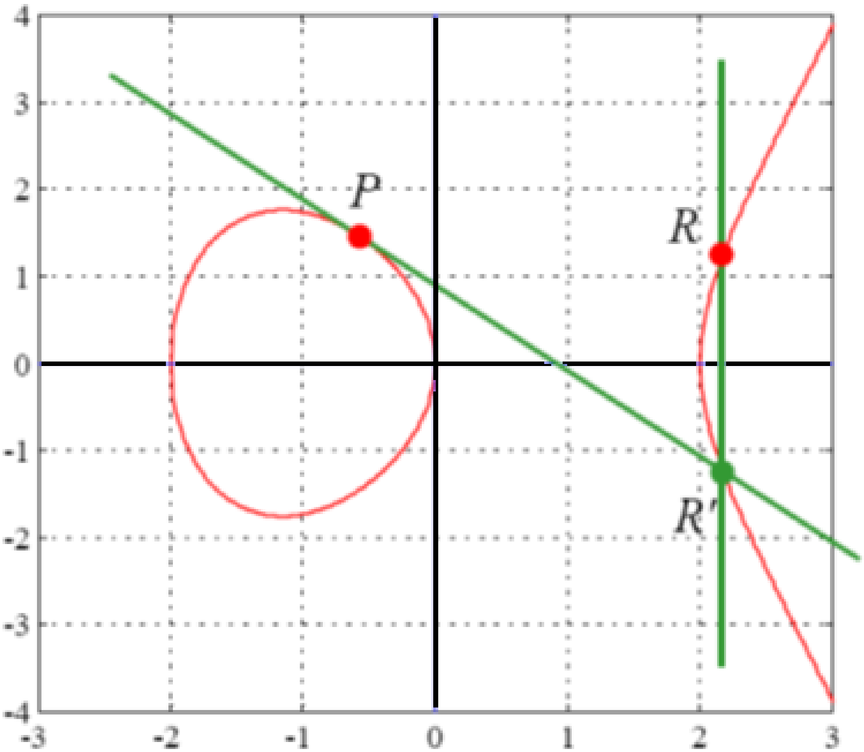




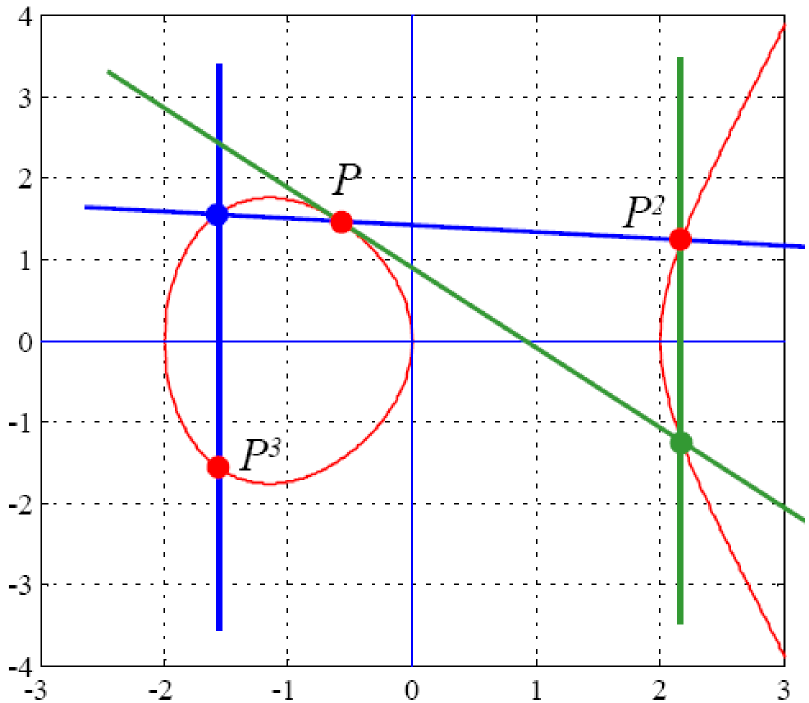


倍数

- 二倍：过点P(x, y)的切线
- P+P+R'=O
- P+P=2P=-R'



- 数乘，多次累加：
- kP=P+···+P



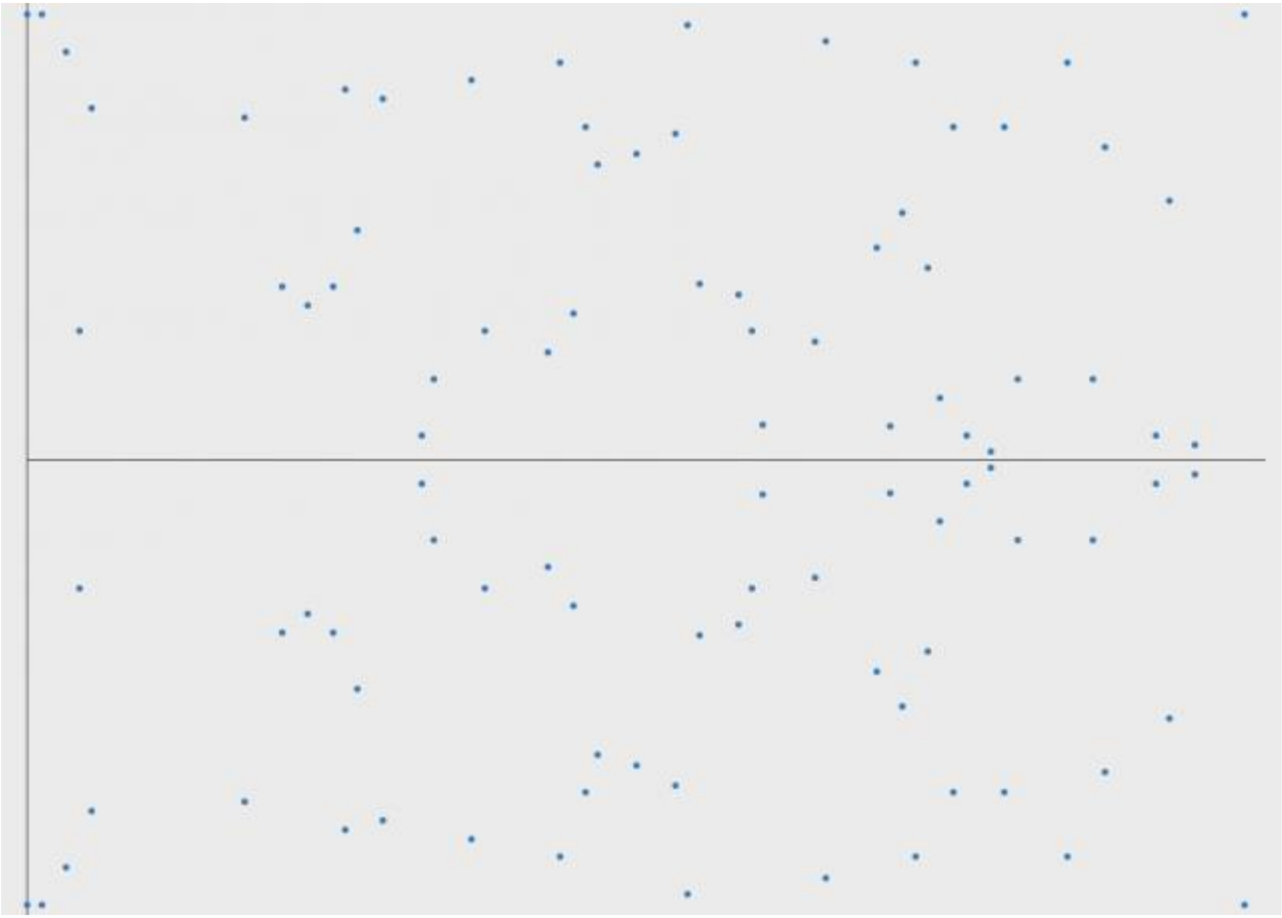
椭圆曲线上的有限加法群—描述

- 使用的椭圆曲线方程： $y^2=x^3+ax+b$
- 对称于$y=0$这条直线
- 参数a及b必需满足$4a^3+27b^2\neq 0$，才能确保没有重根，具有唯一解
- 加法单位元素O为一无穷远的点，并满足$O= -O$
- 此加法单位元素亦需满足：椭圆曲线在某三点共线其和为O

难题

- 定义了基本的加法和乘法运算后，可以得到椭圆曲线加密依赖的数学难题：
  - k为正整数，P是椭圆曲线上的点（称为基点），已知$k*P$和P，计算k
- 改一种记法，把椭圆曲线上点的加法记作乘法，原来的乘法就变成了幂运算：
  - k为正整数，P是椭圆曲线上的点，已知P^k和P，计算$k=\log_P P^k$

◆ 椭圆曲线 $y^2=x^3-x+1$ 对素数97取模后的图像。原本连续光滑的曲线变成了离散的点，基本已经面目全非了，依然可以看到它是关于某条水平直线（y=97/2）对称的



两种有限域上的椭圆曲线

- 定义在Z_p上的曲线$E_p(a,b)$ → ***M-V* 公钥密码**
  - 整数运算对素数p取模
  - 适于软件实现
- 定义在$GF(2^m)$上的**二元曲线**$E_{2^m}(a,b)$
  - 二值系数的多项式运算
  - 适于硬件实现

素数域上的椭圆曲线

- 将椭圆曲线定义于有限域GF_p上：
  - $y^2=x^3+ax+b \mod p$
  - p是一个素数，并且
  - {0，1，…，p-1}是模p加的交换群(Abelian)；
  - {1，…，p-1}是模p乘的交换群
- 椭圆曲线密码，使用变量和参数都在有限域上的椭圆曲线

Z_p上的素曲线$E_p(A, B)$

- $E_p(a,b)$表示满足下列条件的模p椭圆群，群中元素(x,y)是满足方程$y^2 \equiv x^3+ax+b \pmod{p}$的小于p的非负整数对，另外加上无穷点O
  - 若$(x^3+ax+b) \bmod p$没有重复因子（没有重根），则基于集合$E_p(a,b)$可定义一个有限阿贝尔群，即要求：
  - $(4a^3+27b^2) \neq 0 \pmod{p}$
- 例如
  - $p=23$, $y^2=x^3+x+1$
  - $4\times 1^3+27\times 1^2 \bmod 23 = 8 \neq 0$，满足条件

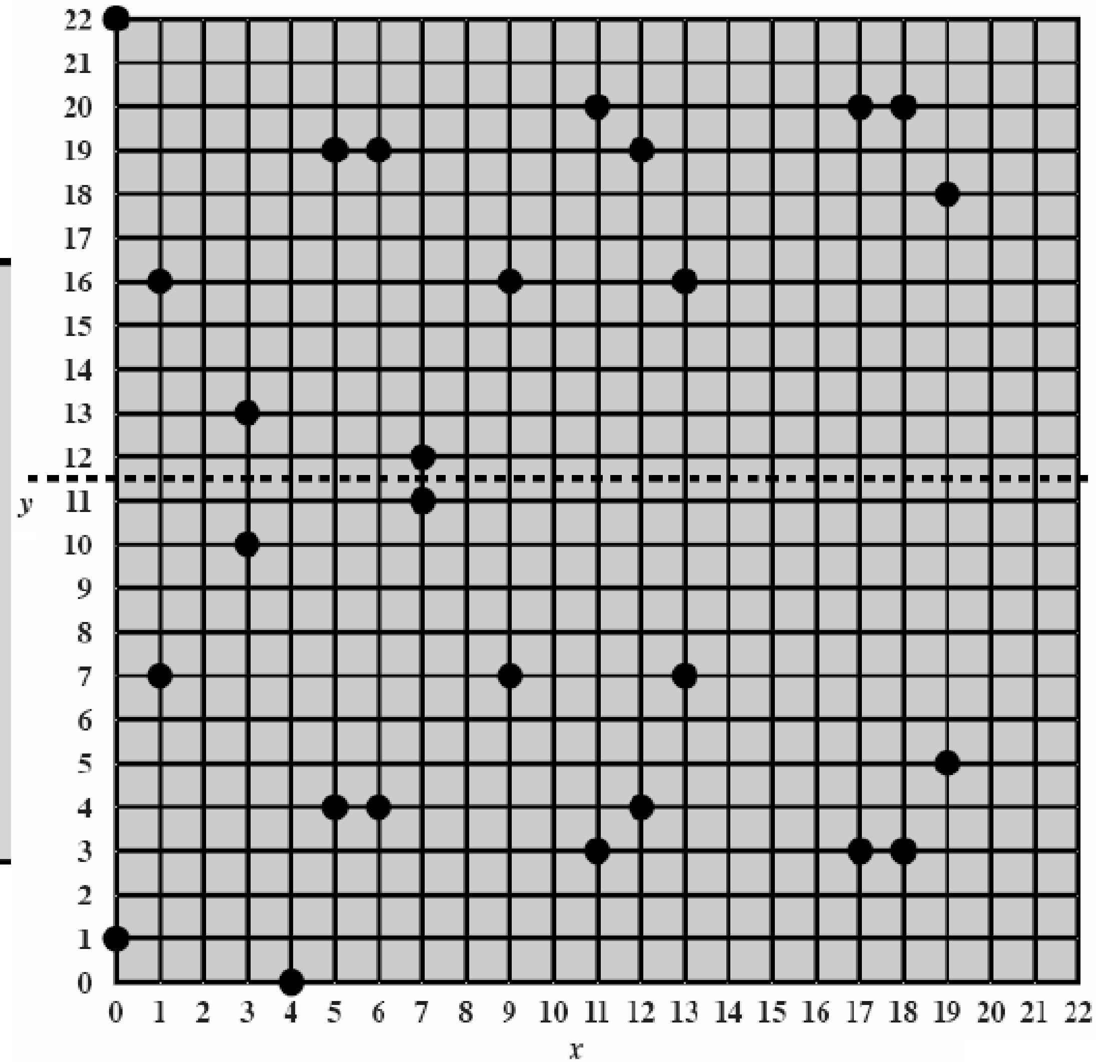
椭圆曲线上的点

- 1.对于每个满足$0 \leqslant x < p$的x，计算$x^3+ax+b \mod p$
- 2.对于上式的结果，确定它是否有一个模p的平方根
  - a)如果没有，在$E_p(a, b)$中就没有具有这个x值的点
  - b)如果有，就有两个满足平方根运算的y值(除非这个值是单个的y值0)。
- 3.所有满足条件2的(x, y)就是$E_p(a, b)$中的点

例子

- $E_{23}(1,1)$上的点（除O点外）
- $y^2=x^3+x+1 \mod 23$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)



椭圆曲线上的*M-V*公钥密码

- 有限域上的椭圆曲线
- Menezes-Vanstone公钥密码体制

有限域上的椭圆曲线

一般，椭圆曲线是方程 $y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_5$

所确定的平面曲线。经过坐标变换可转化为

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

有限域 Z_p（ $p\neq 2$ ）(p是素数)上的椭圆曲线是满足同余方程

$y^2\equiv(x^3+ax+b)\bmod p,\quad a,b\in Z_p,\qquad 4a^3+27b^2\not\equiv 0\bmod p$

判别式

的点 $(x,y)\in Z_p\times Z_p$ 再加上无穷远点O所组成的集合，记为

$E_p(a,b)$ ，即

$$E_p(a,b)=\{O\cup(x,y)\,|\,y^2=(x^3+ax+b)\bmod p\}$$

有限域上的椭圆曲线的结构

可证明：椭圆曲线$E_p(a,b)$关于以上定义的加法构成一个交换群。

用|E|表示有限域上的椭圆曲线$E_p(a,b)$中点的数目，要精确计算该值是困难的，Hasse定理给出其上界和下界。

Hasse定理：$$p+1-2\sqrt{p} \le |E| \le p+1+2\sqrt{p}$$

例：

给定椭圆曲线

$E_{11}(1,6)$

Euler准则：如果p是一个奇素数，则z是模p的平方剩余 $x^2 \equiv z \pmod{p}$ 当且仅当 $z^{(p-1)/2} \equiv 1 \bmod p$

因为 $$(\pm z^{\frac{p+1}{4}})^2 = z^{\frac{p+1}{2}} \pmod{p}$$

$$= z \bullet z^{\frac{p-1}{2}} = z \bullet 1 \pmod{p} = z \pmod{p}$$

当$p \equiv 3 \pmod{4}$时，如果z是模p的平方剩余，

则$\pm z^{(p+1)/4}$就是z的两个模p的平方根。

有限域上的椭圆曲线——数量估计

例： 椭圆曲线$E_{11}(1,6)$中的点

x	$x^3+x+6 \bmod 11$	是否为模11的平方剩余	y
0	6	不是	
1	8	不是	
2	5	是	4 , 7
3	3	是	5 , 6
4	8	不是	
5	4	是	2 , 9
6	8	不是	
7	4	是	2 , 9
8	9	是	3 , 8
9	7	不是	
10	4	是	2 , 9

计算： $$p+1-2\sqrt{p} \le |E| \le p+1+2\sqrt{p}$$

问： 总共几个点？

例子

- GF_{11}上椭圆曲线方程$y^2=x^3+x+6 \bmod 11$的点
- 共有N=13个点，N称为椭圆曲线群的阶
- 问：P(x,y)与P’(x,y)的关系？

x	y^2	$y_{1,2}$	P(x,y)	P’(x,y)
0	6	-		
1	8	-		
2	5	4,7	(2,4)	(2,7)
3	3	5,6	(3,5)	(3,6)
4	8	-		
5	4	2,9	(5,2)	(5,9)
6	8	-		
7	4	2,9	(7,2)	(7,9)
8	9	3,8	(8,3)	(8,8)
9	7	-		
10	4	2,9	(10,2)	(10,9)
∞	∞	∞	O	

有限域上的椭圆曲线——计算

可以在椭圆曲线$E_p(a,b)$上定义**加法运算**：对于任意两点

$$P=(x_1,y_1)\in E,\quad Q=(x_2,y_2)\in E$$

$$P+O\overset{\text{定义}}{=}P$$

$$P+Q\overset{\text{定义}}{=}\begin{cases} O & \text{if} \quad x_1=x_2, y_1=-y_2 \\ (x_3,y_3) & \text{else} \end{cases}$$

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

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

$$\lambda=\begin{cases} \dfrac{y_2-y_1}{x_2-x_1} & \text{if} \quad P\neq Q \\ \dfrac{3x_1^2+a}{2y_1} & \text{if} \quad P=Q \end{cases}$$

有限域上的椭圆曲线——计算

设$\alpha=(2,7)$,计算$2\alpha=\alpha+\alpha=(2,7)+(2,7)$的过程如下

$$\lambda=\frac{3x_1^2+a}{2y_1}=\frac{3\bullet 2^2+1}{2\bullet 7}=2\bullet 3^{-1}=2\bullet 4=8$$

$$x_3=\lambda^2-x_1-x_2=8^2-2-2=-2-2-2=5$$

$$y_3=\lambda(x_1-x_3)-y_1=8(2-5)-7=-2-7=2$$

$$2\alpha=(5,2)$$

计算$3\alpha=2\alpha+\alpha=(5,2)+(2,7)$的过程如下：

$$\lambda=\frac{y_2-y_1}{x_2-x_1}=\frac{7-2}{2-5}=5\bullet 8^{-1}=5\bullet 7=2$$

$$x_3=\lambda^2-x_1-x_2=2^2-2-5=4-2-5=8$$

$$y_3=\lambda(x_1-x_3)-y_1=2(5-8)-2=5-2=3$$

$$3\alpha=(8,3)$$

有限域上的椭圆曲线—计算

类似，可以计算出 $n\alpha, n \geq 1$, 如下：

$$\alpha=(2,7),\quad 2\alpha=(5,2),\quad 3\alpha=(8,3),\quad 4\alpha=(10,2)$$
$$5\alpha=(3,6),\quad 6\alpha=(7,9),\quad 7\alpha=(7,2),\quad 8\alpha=(3,5)$$
$$9\alpha=(10,9),\quad 10\alpha=(8,8),\quad 11\alpha=(5,9),\quad 12\alpha=(2,4)$$
$$13\alpha=O$$

因此 $\alpha=(2,7)$ 是椭圆曲线 $E_{11}(1,6)$ 的本原根，$E_{11}(1,6)$ 是一个循环群

使用椭圆曲线建立密码算法

- 实际应用中，并不关心椭圆曲线的众多参数如何选取（要选对参数，对于普通使用者来说不现实）
- 从密码学家们精心挑选的一堆曲线中选择一个就行了。一般曲线Curve25519，prime256v1是比较常用的，比特币选择secp256k1则是有自己的考量
- 利用ECC实现加/解密的技术**有多种**

ECC 加/解密（ ECELGAMAL ）

- 结合Elgama的：
  - a)首先，将消息m编码为椭圆曲线上形如x-y的一点P_m
  - b)选择适当的椭圆曲线和基点G
  - c)每个用户选择私钥$n_A<n$，并计算公钥$P_A=n_AG$
- 加密P_m过程：$C_m=\{kG, P_m+kP_B\}$，其中k为随机正整数
- 解密C_m过程：$P_m+kP_B - n_B(kG) = P_m+k(n_BG) - n_B(kG) = P_m$

例

- $P=751$，$E_p(-1,188)$，$y^2=x^3-x+188$,，取$G=(0,376)$
- A发送给B的消息编码为点$P_m=(562,201)$
- A随机选择$k=386$，B的公钥为$P_B=(201,5)$
- 计算：
  - $kG=386(0,376)=(676,558)$
  - $P_m+kP_B=(562,201)+386(201,5)=(385,328)$
- 因此，密文为$C_m=\{kG, P_m+kP_B\}=\{(676,558),(385,328)\}$

M-V公钥密码算法—加解密（1）

椭圆曲线上的离散对数问题：

设p>3是一个素数，$E_p(a,b)$是有限域Z_p上的椭圆曲线，设G是$E_p(a,b)$的一个循环子群，α是G的一个本原根，$\beta \in G$

已知 α,β 求满足 $n\alpha=\beta$ 的唯一整数n$(0 \le n \le ord(\alpha)-1)$

即为椭圆曲线上的离散对数问题。

M-V 公钥密码体制：以下4步构造用户的公钥及私钥

（1） 设p>3是一个素数，$E_p(a,b)$是有限域 Z_p上的椭圆曲线，

$\alpha \in$ $E_p(a,b)$ 是椭圆曲线上的一个点，并且阶足够大，

使得由 α生成的循环子群中离散对数问题是难解的。

p和$E_p(a,b)$以及α对全系统都是公开的数据。

M-V公钥密码——加解密（2）

（2）随机选取整数d，$1 < d < ord(\alpha)$，计算 $\beta = d\alpha$

β 是用户B的公钥，d是其私钥。

（3）明文空间为 $Z_p^* \times Z_p^*$，密文空间为 $E \times Z_p^* \times Z_p^*$

加密：对于任意明文 $x = (x_1, x_2) \in Z_p^* \times Z_p^*$

秘密随机选取一个整数k，$1 < k < ord(\alpha)$

密文为 $y = (y_0, y_1, y_2)$，其中

$$y_0 = k\alpha, (c_1, c_2) = k\beta, y_1 = c_1 x_1 \bmod p, y_2 = c_2 x_2 \bmod p$$

若c_1或c_2之一为0时，重新选取k，最终使c_1和c_2均不为0。

（4）解密：计算 $dy_0 = (c_1, c_2)$,

则 $x_1 = y_1 c_1^{-1} \bmod p$, $x_2 = y_2 c_2^{-1} \bmod p$

M-V公钥密码——加解密（3）

解密过程的正确性证明：

$$dy_0 = dk\alpha = kd\alpha = k\beta = (c_1, c_2)$$

$$y_1 c_1^{-1} \bmod p = c_1 x_1 c_1^{-1} \bmod p = x_1$$

$$y_2 c_2^{-1} \bmod p = c_2 x_2 c_2^{-1} \bmod p = x_2$$

3. 椭圆曲线上公钥密码的特点

优点： 同样安全强度下密钥短；处理速度快。
可以应用于计算和存储能力小的智能卡等场合。

不足： 相对DES等单钥密码体制，加解密速度还是太慢，实际应用时，一般只应用于密钥的加密和解密。

设计困难，实现复杂

如果序列号设计过短，那么安全性不够完善

ECDH算法 —密钥交换

- 小红和小明约定使用某条椭圆曲线（包括曲线参数，有限域参数以及基点等）
- 小红生成私钥x，计算x*P作为公钥公布出去
- 小明生成私钥y，计算y*P作为公钥公布出去
- 小红得知y*P后，计算s=x*(y*P)=xy*P
- 小明得到x*P后，计算s=y*(x*P)=yx*P
- 双方都得到了相同的密钥的s，交换完毕
- 由于计算椭圆曲线上的离散对数是很难的，所以第三方没办法在只知道x*P和y*P的情况下计算出x或y的值。

消息映射算法

- 有多种方法，一例：
- 编码：
  - 将域p划分为长256的小段
  - 对明文进行分组：使得每个分组$0 \leqslant m \leqslant \lfloor p/256 \rfloor$
  - 对明文分组进行编码，使之成为由域参数给出的椭圆曲线上的点P_m
    - ✓ 在$256m \leqslant x < 256(m+1)$中找到一个x，使得椭圆曲线方程有解
    - ✓ 一般地，对所有的满足$256m \leqslant x < 256(m+l)$的x，椭圆曲线方程都无解的概率是很小的。从而可以完成编码。
- 解码：
- 若解密横坐标落在$256m \leqslant x < 256(m+1)$中，则解码为m

椭圆曲线密码系统 —小结

- 定义
  - 域标识：定义椭圆曲线采用的有限域
  - 椭圆曲线：系数a和b, $E_p(a,b)$
  - 基准点(base)：指定的椭圆曲线上的点α
  - 阶(order)：α点的阶n，使得：nα=O
- 建立公钥系统
  - E(a, b), GF(p)
  - 基点α(x, y)
  - 选择正整数d作为私有密钥
  - 公开密钥为 β=d α

- 描述一条F_p上的椭圆曲线，用到六个参量：$T=(p,a,b,\alpha,n,h)$。
- （p、a、b 用来确定一条椭圆曲线，α为基点，n为点α的阶，h 是椭圆曲线上所有点的个数m与n相除的整数部分）

- 这几个参量取值的选择，直接影响了加密的安全性。参量值一般要求满足以下几个条件：

- 1、p 当然越大越安全，但越大，计算速度会变慢，200位左右可以满足一般安全要求；
- 2、$p \neq n \times h$；
- 3、$p^t \neq 1 \pmod n$，$1 \leqslant t < 20$；
- 4、$4a^3+27b^2 \neq 0 \pmod p$；
- 5、n 为素数；
- 6、$h \leqslant 4$。

ElGamal密码体制与M-V公钥密码的比较

	ElGamal密码体制	M-V公钥密码
本元根 α	α 是$(Z_p^*,\bullet)$ 中的本元根	α是$(E_P(a,b),+)$中阶非常大的点
公开密钥 保密密钥	公开密钥：$p,\alpha,\beta(=\alpha^d)$ 保密密钥：d	公开密钥：$p,\alpha,\beta(=d\alpha)$ 保密密钥：d
明文空间 密文空间	$Z_p^* \to Z_p^* \times Z_p^*$	$Z_p^* \times Z_p^* \to E_P(a,b) \times Z_p^* \times Z_p^*$
加密算法	对明文$m \in Z_P^*$, 随机选$k(1<k<p-1)$ $c_1=\alpha^k, c_2=m\beta^k$	对明文$m=(m_1,m_2) \in Z_P^* \times Z_P^*$ 随机选$k(1<k<ord(\alpha))$ $y_0=k\alpha, (c_1,c_2)=k\beta$ $y_1=c_1x_1$, $y_2=c_2x_2$
解密算法	对密文$(c_1,c_2) \in Z_P^* \times Z_P^*$ 明文 $m=c_2(c_1^d)^{-1}$	对密文$(y_0,y_1,y_2) \in E_P(a,b) \times Z_P^* \times Z_P^*$ $dy_0=(c_1,c_2)$ $x_1=y_1c_1^{-1}(mod\ p), x_2=y_2c_2^{-1}(mod\ p)$

回顾：椭圆曲线 —运算

定义　　设$a, b \in R$是满足 $4a^2 + 27b^3 \neq 0$ 的实数。那么方程

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

的所有解$(x, y) \in R \times R$集合E，加上一个无穷远点O组成了一个非奇异椭圆曲线。

可以证明，条件 $4a^3 + 27b^2 \neq 0$ 是保证方程 $y^2 = x^3 + ax + b$有三个不同解的充要条件。如果 $4a^2 + 27b^3 = 0$，则对应的椭圆曲线称为奇异椭圆曲线。

在密码学中，更多地关心非奇异椭圆曲线。假设E是一个非奇异椭圆曲线。

在E上定义一个二元运算，通常用加法来表示这个二元运算，使其成为一个阿贝尔（交换）群。　　首先定义无穷远点O位单位元，即有$P + O = O + P = P$，对于任意$P \in E$。现假设，P，$Q \in E$，$P = (x_1, y_1)$，$Q = (x_2, y_2)$是E中任意两点。为了定义$P + Q$，分三种情况讨论。　$(1) x_1 \neq x_2$

$(2) x_1 = x_2, y_1 = -y_2$

$(3) x_1 = x_2, y_1 = y$

情形1

情形 1，当$x_1 \neq x_2$时。定义L是通过P和Q两点的直线。L交E于P和Q，从几何上可以看出，L还交E与第三点，记作R'。记R是点R'关于x轴的反射点。

定义$P+Q=R$。

下面给出计算 R 的代数公式。假设直线 L 的方程为 $y=\lambda x+v$，其中 L 的斜率为：

$$\lambda=\frac{y_2-y_1}{x_2-x_1} \quad 且 \quad v=y_1-\lambda x_1=y_2-\lambda x_2$$

为了计算 $E\cap L$ 中的点，将 $y=\lambda x+v$ 代入 E 的方程中，得到 $(\lambda x+v)^2=x^3+ax+b$

等价于 $\boxed{x^3-\lambda^2x^2+(a-2\lambda v)x+b-v^2=0}$

则方程的根就是$E\cap L$中的点的x坐标值。因为P和Q是$E\cap L$中两个点，所以x_1和x_2是方程的两个根。

方程是实数域上的三次方程，且有两个实根，故第三个根也是实根，记为x_3。根据根与系数之关系，三根之和是二次项系数λ^2的相反数。即

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

x_3是点R'的x坐标。R' 的y坐标记作就是$-y_3$，则R的y坐标就是y_3。求y_3的一个简单的办法是利用L的斜率是由L 上的两点确定这个事实。用点(x_1, y_1)和$(x_3, -y_3)$计算这个斜率，得到

$$\lambda=\frac{-y_3-y_1}{x_3-x_1} \quad 即 \quad y_3=\lambda(x_1-x_3)-y_1$$

于是对情形 1，我们得到$P+Q$的一个计算公式：若$x_1\neq x_2$，则

$$(x_1, y_1)+(x_2, y_2)=(x_3, y_3) \quad 其中 \quad x_3=\lambda^2-x_1-x_2$$

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

$$\lambda=\frac{y_2-y_1}{x_2-x_1}$$

情形2

情形 2; $x_1 = x_2$, $y_1 = -y_2$时，定义$(x , y) + (x , -y) = O$，$(x , y)\in E$。因此，(x , y)和$(x , -y)$是关于E的加法运算互逆的。

情形 3 是定义一个点$p = (x_1 , y_1)$与自己如何相加。当$y_1 = 0$ 时，情形 3 变为情形 2。不妨设$y_1 \neq 0$。这种情形 3 与情形 1 非常相似，只是定义L是E在P 点的切线。计算直线L的斜率要利用对E的微分：

$$2y\frac{dy}{dx} = 3x^2 + a$$

替换$x = x_1$, $y = y_1$，得到切线的斜率：

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

剩下的分析与情形 1 完全相同。得到的公式也是一样，只是斜率的计算不同。

综合上述讨论，我们定义了 E 上的加法运算，它具有下列性质：

1）加法在集合 E 上封闭；

2）加法是交换的；

3）O 是加法的单位元；

4）E 上每一点关于加法存在逆元素。

要证明$(E, +)$是一个阿贝尔群，还须验证加法是结合的。

设$p>3$ 是素数。$\mathbf{Z}_p$上的椭圆曲线E与实数域$\mathbf{R}$上一样定义；只是$\mathbf{R}$上运算用$\mathbf{Z}_p$中类似运算代替即可。

定义　设$p>3$ 是素数，$\mathbf{Z}_p$ 上的同余方程　$y^2=x^3+ax+b(\bmod p)$ 的所有解$(x, y)\in \mathbf{Z}_p\times\mathbf{Z}_p$，连同一个特殊的无穷远点$O$，组成$\mathbf{Z}_p$上的椭圆曲线$E$：$y^2=x^3+ax+b$，其中，$a, b\in \mathbf{Z}_p$是满足 $4a^2+27b^3\equiv 0(\bmod p)$ 的常量。

$\mathbf{Z}_p$上的椭圆曲线 $y^2=x^3+ax+b$ 是由p与a，b共同完全确定的，一般可将其记为 $E_p(a, b)$,或在不指明p与a和b的情况下，简记为E 。

$\mathbf{Z}_p$上的椭圆曲线$E_p(a, b)$的加法定义如下：假设$P(x_1, y_1)$, $Q(x_2, y_2)$ $E_p(a, b)$，

定义
$$\boxed{P+Q=\begin{cases}0, & \text{当}x_1=x_2\text{ 且}y_2=-y_1\text{时}\\(x_3, y_3), & \text{其他}\end{cases}\qquad \text{其中}\quad \begin{aligned}x_3&=\lambda^2-x_1-x_2\\y_3&=\lambda(x_1-x_3)-y_1\end{aligned}}$$

且
$$\boxed{\lambda=\begin{cases}(y_2-y_1)(x_2-x_1)^{-1} & p\neq Q\\(3x_1^2+a)(2y_1)^{-1}, & p=Q\end{cases}}$$

特别地，定义　$P+O=O+P=P$　对任意 $P\in E_p(a, b)$

虽说$\mathbf{Z}_p$上的椭圆曲线E没有实数域上椭圆曲线的直观几何解释，然而，同样的公式可以用来定义加法运算模式$(E, +)$也是一个阿贝尔群。

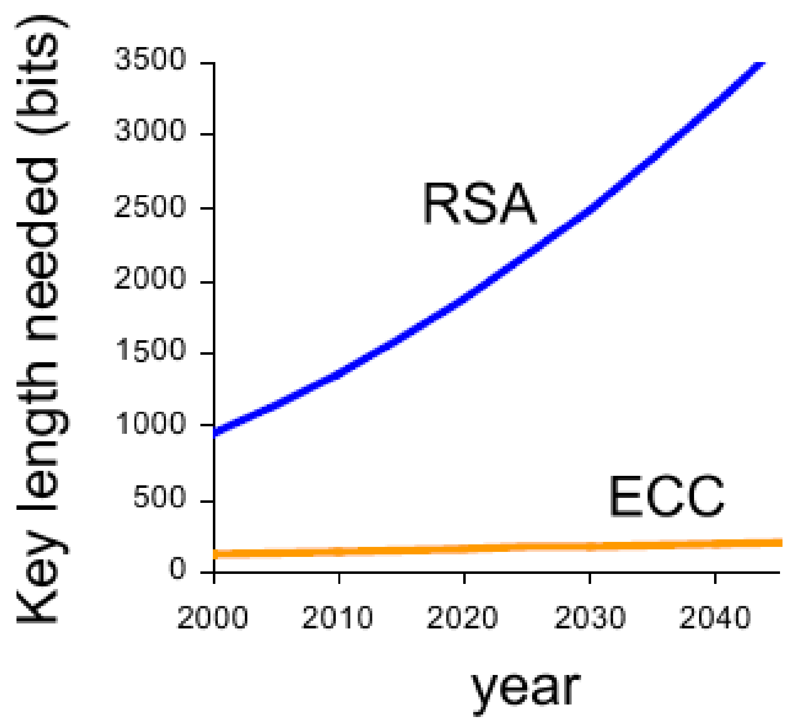
算法分析

- 椭圆曲线密码体制的安全性取决于椭圆曲线离散对数问题的难度
  - "Pollard rho方法"是目前最快的破解方法
- 同等密码强度下，所需密钥量和计算量都小于RSA算法
  - 同等密钥长度下，计算量与RSA相当
  - 密钥长度与密码强度的关系

Symmetric	56	80	112	128	192	256
RSA n ECC p	512 112	1024 161	2048 224	3072 256	7680 384	15360 512
Key size ratio	5:1	6:1	9:1	12:1	20:1	30:1

关于安全方面的比较

- Arjen Lenstra und Eric Verheul: http://cryptosavvy.com/table.htm



有关ECC的研究课题

理论

- 拓展曲线的选择范围
- 算法强度的分析

应用

- 确定椭圆曲线标准：国家、部门、行业
- 应用协议研究
- 专用芯片研究

椭圆曲线公钥密码体制的研究

三个研究方向

（1）椭圆曲线公钥密码体制的构造

（2）椭圆曲线公钥密码体制的分析

（3）椭圆曲线公钥密码体制的快速实现

椭圆曲线公钥密码体制的构造

设计构造椭圆曲线公钥密码体制主要考虑两个问题

（1）选择什么样的有限域作为椭圆曲线的**基域**

（2）用哪类**曲线**来构造公钥密码体制

椭圆曲线快速实现

- 如果模数是Mersenne数$p=2^k-1$， $A<p^2$,将A析分成由高k个比特和低k个比特组成的两个整数t和u，则

$$A \bmod p = 2^k t+u \bmod p$$

$$= (2^k-1)t+t+u \bmod (2^k-1)$$

$$= (t+u) \bmod p$$

ECC相关标准(I)

- ANSI X9.62 (ECDSA)
- ANSI X9.63 (ECIES,ECDH,ECMQV)
- FIPS 186-2 (ECDSA)
- IEEE P1363 (ECDSA,ECDH,ECMQV)
- IEEE P1363 A (ECIES)
- IPSec (ECDH,ECDSA)

ECC相关标准(II)

- ISO/IEC 14888-3 (ECDSA)
- ISO/IEC 15946 (ECDSA,ECDH,ECMQV)
- SEC1/SEC2 (the Standards for Efficient Cryptography)
  (ECDSA,ECDH,ECMQV,ECIES)
- WAP-WTLS (ECDH,ECDSA)
- ГОСТ Р 34.10-2001 （俄联邦电子数字签名标准-RECDSA）
- 美国国防部PKI标准 (ECDSA)

ECC相关产品浏览

- 硬件产品

Motorola：MPC180E

Siemens：SLE 66C160P

/X160P/X320P/x640P

- 软件产品

Certicom：WTLS Plus

ECC发展现状

- 得到广泛关注，逐步占领主要地位，走上了实用

使用前要解决的问题

- 数学选择
- 基域运算
- 曲线运算
- 协议的实现

作业

在$E_{11}(1,6)$上取本原根 α =(2,7)，其所有点列出如下，利用该椭圆曲线实现M-V公钥密码体制。

α=(2,7)，2α =(5,2)，3α =(8,3)，4α =(10,2)，5α =(3,6)，

6α =(7,9)，7α =(7,2)，8α =(3,5)，9α =(10,9)，

10α=(8,8)，11 α=(5,9)，12α=(2,4)，13α=O。

(1) 写出椭圆曲线方程，设用户A的私钥d_A=9，通过两个点相加的方法验证其公开密钥为β_A=(10，9)。

(2) 用户B将明文x=（x_1，x_2）=（3，8）发送给用户A，选择随机数 k=7,求密文 y=（y_0，y_1，y_2）。

(3) 写出用户A恢复明文x=（x_1，x_2）的过程。

参考答案：

（1）椭圆曲线方程为： $y^2=x^3+x+6$

设用户A的私钥 $d_A=9$ 则其公钥 $\beta_A=(10,9)$ 的验证过程如下：

$$\beta_A=9\alpha=5\alpha+4\alpha=6\alpha+3\alpha=\cdots=(7,9)+(8,3)=(10,9)$$

$(7,9)+(8,3)=(x_3,y_3)=(10,9)$ 的计算过程如下：

$$\lambda=\frac{9-3}{7-8}=\frac{6}{-1}=\frac{6}{10}=6\times 10 \bmod 11=5$$

$$x_3=\lambda^2-x_1-x_2=(5^2-7-8) \bmod 11=-1 \bmod 11=10$$

$$y_3=\lambda(x_1-x_3)-y_1=[5(7-10)-9] \bmod 11=9$$

（2）用户B加密明文 $x=(x_1,x_2)=(3,8)$ 的过程如下：

$$y_0=k\alpha=7\alpha=(2\alpha+2\alpha)+((\alpha+\alpha)+\alpha)=(7,2),\beta_A=(10,9)$$

$$(c_1,c_2)=k\beta=7\beta=(2\beta+2\beta)+((\beta+\beta)+\beta)=(5,9)$$

$$y_1 = c_1 x_1 = 5\times 3 = 15 \bmod 11 = 4$$

$$y_2 = c_2 x_2 = 9\times 8 = 72 \bmod 11 = 6$$

所以，密文为 $y = (y_0, y_1, y_2) = ((7,2),4,6)$

（3）用户A恢复明文的过程如下：

$$dy_0 = 9y_0 = 9(7,2) = 9\bullet 7\alpha = 63\alpha = 11\alpha = (5,9) = (c_1, c_2)$$

$$x_1 = y_1 c_1^{-1} = 4\times 5^{-1} = 4\times 9 = 36 \bmod 11 = 3$$

$$x_2 = y_2 c_2^{-1} = 6\times 9^{-1} = 6\times 5 = 30 \bmod 11 = 8$$

$$x = (x_1, x_2) = (3,8)$$

实际中，$9(7,2)$ 的计算过程如下：

$$9(7,2) = (7,2) + \{((7,2)+(7,2)) + ((7,2)+(7,2))\}$$
$$+ \{((7,2)+(7,2)) + ((7,2)+(7,2))\}$$

下次内容

- Hash函数