椭圆曲线探究报告

2111454 李潇逸 信息安全、法学

一、数学问题：

RSA密码体制在我们现在还是非常常用的，但是当今，计算机运算速度之快给该加密方式带来了一定的威胁，为了保证安全性，它的密钥长度需要一再地增大，使其运算负担也随之增大。相比下，椭圆密码体制ECC（elliptic curve cryptography）可以用短得多的密钥获得同样的安全性，因而具有广泛的前景。

在密码中，比较普遍的是采用有限域的椭圆曲线，有限域的椭圆曲线指的是曲线方程y2+axy+by=x3+cx2+dx+e（一般形式）中，所有系数都是某一有限域GF（p）中的元素（其中p为一个大素数）。（GF（p）是定义在整数集合{0, 1, 2, …, p-1}上的域，GF（p）上的加法和乘法分别是模加法和模乘法）。其中最常用的曲线方程是y2≡x3+ax+b(mod p)（其中a，b属于GF（p），4a3+27b2≠0）。

假定P点为(x1, y1)，Q点为(x2, y2)，P+Q为(x3, y3)，因此P+Q由以下规则确定：

x3≡k2-x1-x2(mod p)

y3≡k(x1-x3)-y1(mod p)

k有两种情况：（1）P=Q，（2）P≠Q

P=Q情况下：k=(3x12+a)/(2*y1)

P≠Q的情况下：k=(y2-y1)/(x2-x1)

二、算法思想和特点：

1.扩展欧几里得算法：求一个数的模逆

```
long long exgcd(long long a,long long b,long long &x,long long &y)
{
    if(b==0)
    {
        x=1,y=0;
        return a;
    }
    long long ret=exgcd(b,a%b,y,x);
    y-=a/b*x;
    return ret;
}

long long getInv(long long a,long long mod)
{
    long long x,y;
    long long d=exgcd(a,mod,x,y);
    return d==1 ? (x % mod + mod) % mod : -1;
}
```

2. 点加：在上方已说过

```
Point ecc::Pointadd(Point point1,Point point2)
{
    long long t;
    Point ans;
    if(point1.isInfinity)
    {
        return point2;
    }
    if(point1.operator==(point2))
    {
        long long t1 = getInv(2*point1.y,p);
        t = ((3 * (point1.x * point1.x) + a) * t1) % p;
    }
    else
    {
        long long t2 = point2.x-point1.x;
        if(t2 == 0)
        {
            ans.isInfinity = true;
            return ans;
        }
        if(t2 < 0)
        {
            t2 += p;
        }
        long long t1 = getInv(t2,p);
        t = ((point2.y-point1.y) * t1) % p;
        if(t < 0)
```

三、在密码学中的应用：

密钥交换：最著名的例子是椭圆曲线迪菲-赫尔曼密钥交换（ECDH）。在ECDH中，两个通信方各自选择一个私钥（一个随机数）并计算相应的公钥（私钥点倍加的结果）。然后，它们交换公钥，并使用对方的公钥与自己的私钥生成共享密钥。由于ECDLP的困难性，即使攻击者知道公钥，也很难计算出共享密钥。

数字签名：椭圆曲线数字签名算法（ECDSA）是一种广泛使用的签名方案。在ECDSA中，发送方使用其私钥生成消息的签名，并与消息一起发送。接收方使用发送方的公钥验证签名的有效性。由于ECDLP的复杂性，伪造有效的签名非常困难，除非你知道私钥。

加密：虽然椭圆曲线本身不直接用于加密，但它可以与其他加密技术（如ElGamal）结合使用来创建椭圆曲线加密方案。