椭圆曲线密码系统(ECC)整体算法设计及优化研究

侯整风, 李 岚
(1. 合肥工业大学计算机与信息学院, 安徽合肥 230009)

摘 要: 在安全性差不多的情况下, ECC 使用的密钥要比 RSA 短得多. 然而, 虽然 ECC 的数学理论已比较成熟, 但其算法要比 RSA 更难理解, 算法实现也比 RSA 困难得多, 往往需要通过专门的硬件来加速算法实现. 本文设计了一种 ECC 整体算法, 并对其中的点积、平方剩余判定等关键算法进行优化, 提高了算法的效率, 使其能够完全通过软件实现.

关键词: 密码体制; 椭圆曲线; 点积; 平方剩余
中图分类号: TN393.08 文献标识码: A 文章编号: 0372-2112 (2004) 11-1904-03

The Research on Designing and Optimizing of the Algorithm for Elliptic Curve Cryptography (ECC)

HOU Zheng-feng, LI Lan
(1. *School of Computer and Information Hefei University of Technology, Hefei, Anhui 230009, China*)

Abstract: ECC uses much shorter key than RSA does in the case of the same security. Although the mathematical theory of ECC has become perfect, understanding and implementing its algorithms is much more difficult than those of RSA. It often needs special ECC chip to speed the algorithm implementing. This paper introduces a kind of entire ECC algorithms we have designed. To raise the efficiency of the ECC implementing, the algorithms for point multiplying and square overplus judging have been optimized, so that the whole ECC algorithms can be implemented through software.

Key words: cryptosystem; elliptic; point multyplying; square overplus

1 引言

随着计算机网络的发展和普及, 传统的密码系统(对称密钥系统)逐步暴露出它的两个严重缺陷: (1) 密钥管理和分配较困难; (2) 在数字签名和身份认证方面的应用非常困难. 自 1976 年斯坦福大学的 Diffie 和 Hellman 首次提出一种全新的密码体制 —— 公开密钥体制(又称为非对称密钥系统)[1], 公开密钥系统的研究已成为信息安全领域中一个引人瞩目的研究课题. 目前, 比较成熟的公钥系统有 Merkle 和 Hellman 的基于背包问题的公钥系统, Rivest, Shamir 和 Adleman 的基于数论的 RSA 系统[2], 还有具有纠错和加密双重功能的 McEliece 公钥系统. 其中 RSA 系统已被广泛地应用与计算机安全领域. 自 80 年代中期, 椭圆曲线理论被引入数据加密领域, 逐步形成一个挑战 RSA 系统的公开密钥系统 ——椭圆曲线密码系统(ECC)[3,4]. ECC 的安全性依赖于椭圆曲线离散对数问题的难解性. 到目前止, 取椭圆曲线离散对数的最快方法是 Pollard rho 方法. 若对 RSA 系统采用通用数域筛方法进行因数分解, 通过比较这两种方法的效率可以看到, 在安全性差不多的情况下, ECC 要比 RSA 使用小得多的密钥尺寸[5]. 此外, 在数字签名和身份认证应用方面, ECC 也明显优于 RSA. 然而, 尽管 ECC 的数学理论问题已基本解决, 但其算法的设计和优化及其实现仍存在很大难度, 甚至不得不设计专门的 ECC 芯片来提高执行速度, 这使得 ECC 离实际应用仍有一定的距离. 本文余部将介绍我们应用椭圆曲线理论和数论知识所设计的一种完全基于软件的 ECC.

2 有关 ECC 的数学定义和定理

定义 1 韦尔斯特拉斯(Weierstrass)方程:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

所确定的平面曲线称为椭圆曲线, 记为 E; 其中 $a, b, c, d, e, \in F_p$; F_p 为有限域. 满足式(1)的(x, y)称为 F_p 域上的点. 此外, 椭圆曲线还定义一个特殊的无穷点 O.

对于有限域 F_p, 如果其特征值不是 2 或 3, 则可通过一系列变换将式(1)变换成一种较简单的形式:

$$y^2 = x^3 + ax + b \quad (2)$$

ECC 感兴趣的是所谓模 p 椭圆群, 即给定一个素数 p, 选择两个小于 p 的非负整数 a 和 b, 使得 $4a + 27b \pmod p \neq 0$. 那么这个群中的数偶(x, y)均满足如下方程且小于 p 的非负整数(外加无穷点 O): $y^2 \equiv x^3 + ax + b \pmod p$ (3)

因此, 对 ECC 来说, 椭圆曲线参数可以由一个多元偶$(a,$

收稿日期: 2002-11-04; 修回日期: 2004-07-16

$b, p)$描述, 其中 p 是一素数, 表示一个有限域 F_p; $a, b \in F_p$.

定义 **2**　设有限域 F_p 上椭圆曲线为 $y^2 \equiv x^3 + ax + b \pmod p$, P, Q 为 E 上的两点, L 是 P, Q 的连线, R 为 L 与 E 相交的另一点, L' 是 R 和无穷点 O 的连线(即 L' 是过 R 点与 Y 轴平行的直线), 则 L' 与 E 的另一交点 S 称为 P 与 Q 的点加, 记为 $S = P + Q$.

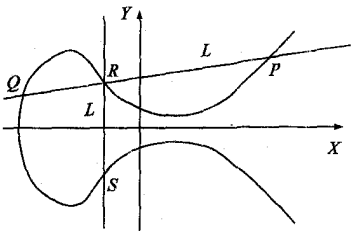

图 1

定理 **1**　设 P, Q 为 E 的任意两点, L 为 P, Q 连线, L 与 E 相交于另一点 R, 则:

(1) $P + O = P$

(2) $P + Q + R = O$

(3) $P + Q = Q + P$

(4) 若 $P + Q = O$, 则称 Q 为 P 的负点, 记为 $Q = -P$.

(5) $(P + Q) + R = P + (Q + R)$

定义 **3**　k 个 P 的点加称为 k 与 P 的点乘, 记为:

$$kP = P + P + P + \cdots + P \quad (k \text{ 个 } P)$$

定义 **4**　设 m 为大于 1 的整数, 若

$$x^2 \equiv n \pmod m$$

可解, 则称 n 为对模 m 的平方剩余; 否则称 n 为对模 m 的非平方剩余.

定义 **5**　设 p 是一素数, a 为一整数, a, p 互素, 则勒让德符号 (a/p) 定义为 $(a/p) = \pm 1$; (a 为模 p 的平方剩余则为 1, a 为模 p 的非平方剩余则为 -1.)

定理 **2**　$(1 \backslash P) = 1$

定理 **3**　$(2 \backslash P) = (-1)^{(p^2-1)\backslash 8}$

定理 **4**　当 p, q 为两个不相同的奇素数时, 则

$$(q/p)(p/q) = (-1)^{((p-1)/2)((q-1)/2)}$$

定理 **5**　若 p 为奇素数, 且 $p \nmid mn$, 则 $(mn/p) = (m/p)(n/p)$.

3　ECC 的整体算法

3.1　确定椭圆曲线参数

选取某一椭圆曲线 $E: y^2 \equiv x^3 + ax + b \pmod p$. 该椭圆曲线通过一组参数$(p, a, b, G)$描述; p 是一个足够大的素数, 表示一个有限域 F_p; $a, b \in F_p$ 且满足 $4a^3 + 27b^2 \pmod p \neq 0$; $G(x, y)$ 为 $E: y^2 = x^3 + ax + b$ 上的一个点, 称为基点. 选择基点 G 的基本准则是满足 $nG = 0$ 的最小 n 值是一个足够大的素数. 选定好的参数(p, a, b, G)保存在一个公用文件中, 可以被使用该 ECC 系统的任何用户使用.

ECC 是以椭圆曲线离散对数问题的难解性为基础的, 即在给定 P 和 kP 的条件下很难推导出 k. 当所选择的群的阶较小时, 离散对数问题容易求解从而使其易受攻击. 如 Baby-step Giant-step 方法、Phling-Hallman 方法都能够容易地攻破低阶 ECC. 因此, 为了确保 ECC 的安全性, 所选择的群的阶必须足够大, 即对有限域 F_P 来说, 素数 p 必须足够大. Menezes 等人证明了只要使椭圆曲线的阶含有至少 40 位十进制数的大素数因子, 也就是说椭圆曲线所在的域的大小至少为 2^{130}, 就可以有效地防止现有的一切攻击.

3.2　私钥和公钥以及共享密钥的产生

设有用户 A 和 B, 则它们的私钥和公钥以及共享密钥按如下步骤产生:

(1)用户 A 随机选取一整数 S_A 作为自己的私钥, 计算 $P_A = S_A G$(点积运算)作为自己的公钥.

(2)用户 B 用类似的方法产生自己的私钥和公钥 S_B 和 P_B.

(3)用户 A 产生共享密钥 $K_A = S_A P_B$, 用户 B 产生共享密钥 $K_B = S_B P_A$. 显然 $K_A = S_A P_B = S_A(S_B G) = S_B(S_A G) = S_B P_A = K_B$.

3.3　明文映射到椭圆曲线上

在对明文 M 加密之前, 需将 M 映射到椭圆曲线的有限域 F_p 上的一个点上; 如果 M 较长, 可分段处理. 值得注意的是不能简单地将 M 映射到椭圆曲线的任意点上, 而必须是 F_p 域中的点, 即椭圆曲线上满足平方剩余的点.

映射方法并非唯一, 本文采用的映射方法如下:

首先对 M 进行分段处理. 设 m 为 M 的一个分段, m 满足条件:

$$0 \leq m \leq [p/256] - 1$$

将 m 映射到点 $P_m(x, y)$ 上, 使得:

$$\begin{cases} 256m \leq x \leq 256(m+1) \\ P_m(x, y) \in F_p \end{cases}$$

找点 $P_m(x, y)$ 并不困难, 因为当 $256m \leq x \leq 256(m+1)$ 时, $y^2 \equiv x^3 + ax + b \pmod p$ 是一个非平方剩余的概率很小.

3.4　加密

明文分段 m 映射到点 $P_m(x, y)$ 不是加密, 仅仅是一种编码, 任何用户都可以通过解码将 $P_m(x, y)$ 恢复成 m. 因此, $P_m(x, y)$ 实际上可视为明文. 在发送 $P_m(x, y)$ 前, 需对其进行如下加密处理得到密文 $C_m = P_m + S_A P_B$.

3.5　解密

接收端收到密文 C_m 后, 进行如下处理恢复明文 P_m:

$$P_m = (P_m + S_A P_B) - S_B P_A$$

3.6　解码

接收端得到 $P_m(x, y)$ 后, 取出 P_m 的 X 坐标值 x, 再用下式得到明文 m:

$$m = [x/256]$$

4　ECC 关键算法优化

ECC 密码体制实现面临的挑战是算法运算量大. 下面将讨论 ECC 关键算法优化问题.

4.1　点积快速算法

公钥产生和加密/解密算法中需要大量的点积运算, 即计算

$$nP = P + P + \cdots + P \quad (n \text{ 个 } P)$$

本文采用的点积快速算法如下:

(1) 将 n 表示成二进制数形式, 即

$$n=(n_k n_{k-1}\cdots n_i\cdots n_1)$$

其中, $n_i=0$ 或 1, $k=[\log_2 n]+1$

(2) 去掉 $(n_k n_{k-1}\cdots n_i\cdots n_1)$ 的最高位 n_k, 得 $(n_{k-1}\cdots n_i\cdots n_1)$.

(3) 按照 $(n_{k-1}\cdots n_i\cdots n_1)$ 从高位到低位次序, 当 $n_i=0$, 计算 $2P$, 当 $n_i=1$, 计算 $2P+P$, 并将结果作为下次计算的初值, 即 $2P\Rightarrow P$ 或 $2P+P\Rightarrow P$.

例如, $n=27=(11011)_2$, 其运算次序为 1011, 迭代过程为:

$$2P+P\Rightarrow P$$
$$2P\Rightarrow P$$
$$2P+P\Rightarrow P$$
$$2P+P\Rightarrow P$$

采用常规方法, nP 需进行 n 次点加运算; 在本算法中, 平均只须 $3/2[\log_2 n]$ 次运算, 最多需要 $2[\log_2 n]$ 次运算.

4.2 平方剩余快速判定算法

在明文到椭圆曲线映射过程中, 需要判别一个数是否为模 p 下的平方剩余, 即平方剩余判定. 目前现有的平方剩余判定算法仅仅简单地根据平方剩余定义来判别, 涉及到大数的平方运算和取模运算, 算法效率非常低. 针对这种情况, 本文设计了一种快速平方剩余判定算法.

设明文分段 m 映射到点 $P_m(x,y)$ 上, 使其满足

$$\begin{cases}256m\le x\le 256(m+1)\\ P_m(x,y)\in F_p\end{cases}$$

下面要解决的问题是在 256 和 $256(256+1)$ 之间给定一个 x, 判定 $A=x^3+ax+b$ 是否是模 p 下的平方剩余. 即判定 (A/q) 是否为 1. 本文提供的快速平方剩余判定算法如下:

(1) 设 J 为平方剩余判定变量, 初始时, $J=1$.

(2) 如果 A 为偶数, 则根据定理 2.5, (A/p) 可分解为

$$(A/p)=(2/p)((A/2)/p)$$

根据定理 3, 计算 $(2/p)$, 然后执行

$$J(2/P)\Rightarrow J$$
$$A/2\Rightarrow A$$

如果 A 为奇素数, 则根据定理 4 得

$$(A/p)(p/A)=(A/p)((p \bmod A)/p)=(-1)^{((A-1)/2)((p-1)/2)}$$
$$(A/p)=(-1)^{((A-1)/2)((p-1)/2)}((p \bmod A)/A)$$

这样, 对 (A/p) 的判定等价于对 $((p \bmod A)/A)$ 的判定, 即执行如下操作:

$$J(-1)^{((A-1)/2)((p-1)/2)}\Rightarrow J$$
$$A\Rightarrow q$$
$$P \bmod A\Rightarrow A$$
$$q\Rightarrow p$$

如果 A 为奇数但不为素数, 可将 A 分解为 $\prod A_i$, 其中 A_i 为奇素数, 根据定理 2.5 得:

$$(A/p)=(A_1/p)(A_2/p)\cdots(A_i/p)\cdots(A_k/p)$$

再分别计算 (A_i/p).

(3) 当 $A\neq 1$ 时, 返回 (2); 否则, 算法结束. 此时根据 J 值判定 x^3+ax+b 是否是模 p 下的平方剩余; 若 $J=1$, 则为平方剩余, 若 $J=-1$, 则为非平方剩余.

5 结束语

我们以数论中的椭圆曲线离散对数问题为基础, 对 ECC 公开密钥体制进行了长期研究, 设计了一种 ECC 实现的整体算法, 并对传统的点积算法和平方剩余判定算法进行了优化, 大大地提高了算法的效率, 使其能够在 PC 机环境下完全通过软件实现. 该整体算法现已在校园网中两台 PC 机之间用 Visual C++ 6.0 予以实现. 结果表明该 ECC 整体算法是正确和切实可行的. 同时, 该 ECC 整体算法在实现过程中还存在一些问题: (1) 大整数的模逆和模幂算法效率较低; (2) 我们在实验中所选择的 p 值大约为 2^{64} 左右, 其安全性相对降低. 这些问题有待于今后进一步研究.

参考文献:

[1] Diffie W, Hellman W E. New direction in cryptography[J]. IEEE Trans On Information Theory, 1976, IT-22(11): 644-654.

[2] Rivest R L, Shamir A, Adleman L. On a method for obtaining digital signatures and public key cryptosystem[J]. Commun of ACM, 1978, 21(2): 120-126.

[3] Kumanduri R, Romero C. Number Theory with Computer Applications [M]. Upper Saddle River, NJ: Prentice Hall, 1998.

[4] Menezes A. Elliptic Curve Public Key Cryptosystems [M]. Boston: Kluwer Academic Publishers, 1993.

[5] Jurisic A, Menezes A. Elliptic curves and cryptography[J]. Dr Dobb's Journal, April 1997, 22: 26-37.

作者简介:

侯整风 男, 1958 年出生, 副教授, 1982 年毕业于合肥工业大学, 获学士学位, 1988 年研究生毕业于合肥工业大学, 获硕士学位, 1992 年元月至 1993 年 8 月瑞士日内瓦大学访问学者, 主要研究方向: 计算机网络, 数据库.