SEA 算法的有效实现*

祝跃飞[1,2], 顾纯祥[1], 裴定一[2]

[1](信息工程大学 网络工程系,河南 郑州 450002);
[2](中国科学院 研究生院 信息安全国家重点实验室,北京 100039)
E-mail: zyf0136@sina.com
http://www.home.is.ac.cn

摘要: 选取安全椭圆曲线的核心步骤是对椭圆曲线阶的计算.SEA(Schoof Elkies Atkin)算法是计算椭圆曲线阶的有效算法,同种圈(isogeny cycles)方法是 Morain 对 SEA 算法改善的一种重要局部优化技术.在实现了 F_p 上 SEA 算法的前提下,对同种圈方法作了进一步改进,就 SEA 算法中各方法的综合运用提出一种方案,并且对用 SEA 算法选取素数阶和拟素数阶椭圆曲线速度上的优化作了一些讨论,所获得的一些速度指标和国际公开资料上的指标有可比性.

关 键 词: 椭圆曲线;Frobenius 映射;SEA 算法;同种圈
中图法分类号: TP393 文献标识码: A

自从 1985 年 N. Koblitz 和 V.S. Miller 分别提出椭圆曲线公钥密码体制以来,在理论研究、标准化和产品化等要素的综合影响下,椭圆曲线公钥密码体制以其自身的优势,成为信息安全和密码学界关注的焦点之一.近年来,许多国际标准化组织已将椭圆曲线公钥密码体制作为其标准化(或草案)文件向全球颁布[1~5],同时,依照标准以椭圆曲线密码体制作为安全机制主体的一些安全产品也相继出笼.这些产品在算法上的安全性归结为对椭圆曲线参数的慎重选取,各种标准都在各级安全层次上提供了具体的参数,其核心步骤是寻找素数(或拟素数)阶的椭圆曲线.目前处理此问题的方法有两种:一种是复乘方法,即构造给定阶的椭圆曲线,另一种是随机选取椭圆曲线参数,计算它的阶,直到找到素数(或拟素数)阶椭圆曲线.复乘方法产生的椭圆曲线具有附带的结构特征,从安全性角度来说,这是一个潜在的威胁;而第 2 种随机选取的方法无疑是比较理想的,它完全依赖于椭圆曲线阶的计算.在这方面,R.Schoof[6]做了开创性工作,提出了著名的 Schoof 算法,后经 Atkin[7]和 Elkies[8]的改进,该算法更具有实用价值,因而被称作 SEA(Schoof Elkies Atkin)算法.后来,Morain,Lercier[9~12]等人又对它作了一些优化,现今 SEA 算法已成为计算椭圆曲线阶的有效算法.因而实现 SEA 算法是建立我国自身的椭圆曲线公钥密码标准或安全椭圆曲线密码体制的一个极其重要的环节.

本文在 F_p 上的 SEA 算法实现的前提下,对 SEA 算法中的同种圈(isogeny cycles)方法作了讨论,提出以可除多项式 $h(x)$的次数尽可能低的因子代替 $h(x)$,从而可对较大的 n 或 l 计算 $t \bmod l^n$,且对用 SEA 算法选取素数阶和拟素数阶椭圆曲线速度上的优化进行讨论,所获得的一些速度指标和国际公开资料上的指标有可比性.

* 收稿日期: 2000-08-02; 修改日期: 2001-12-05
基金项目: 国家自然科学基金资助项目(19931010);国家重大基础研究发展规划 973 资助项目(G1999035804);河南省杰出青年基金(0212001400)

作者简介: 祝跃飞(1962－),男,浙江杭州人,博士,教授,博士生导师,主要研究领域为密码学,信息安全,计算数论;顾纯祥(1976－),男,安徽霍山人,助教,主要研究领域为密码学,信息安全;裴定一(1941－),男,江苏常州人,教授,博士生导师,主要研究领域为数论,密码学,代数编码.

1 SEA 算法

设 p 是奇素数,本文考虑定义在有限域 F_p 上的椭圆曲线:

$$E: y^2 = x^3 + ax + b,$$

其中 $a,b \in F_p, a \cdot b \neq 0, \Delta = 4a^3 + 27b^2 \neq 0$.记

$$E(F_p) = \{(x,y) \in F_p{}^2 \mid y^2 = x^3 + ax + b\} \bigcup \{O\}$$

为 E 的 F_p 有理点群, $j(E)$为 E 的 j_不变量, $\Phi_l^c(x,y)$ 为 l_th 模多项式.

1.1 SEA算法回顾

对素数 l(l<<p),记 $E[l]$为 E 的 l_$torsion$ 点群,则 E 的 Frobenius 映射 $\phi:(x,y) \to (x^p, y^p)$ 诱导出 Tate 模 $T_l(E)$ 上的线性映射,且满足方程: $\phi^2 - t\phi + p = 0$,其中 t 是 Frobenius 映射的迹.由 Hasse 引理可知:

$$\#E(F_p) = p + 1 - t, \quad |t| < 2\sqrt{p}. \tag{1}$$

因此,如果找到 t_l 满足

$$\phi^2(P) + [p](P) = [t_l]\phi(P), \quad \forall P \in E[l], \tag{2}$$

就可得到 $t \equiv t_l \bmod l$.只要选取足够多的 l_i,使 $\prod_i l_i > 4\sqrt{p}$,那么由中国剩余定理(CRT),就可计算出 $\#E(F_p)$.

记 l_th 可除多相式为 $f_l(x)$ ($\deg f_l = (l^2-1)/2$),其根恰为 $E[l] \setminus \{O\}$ 中点的 x 坐标.因此,式(2)的计算是在环 $F_p[x,y]/(y^2 = x^3 + ax + b, f_l(x))$ 中进行的.这就是 Schoof 算法的原始思想.不难分析,算法的复杂度为 $O(\log^8 p)$.

当 $E[l]$上 Frobenius 映射在 F_l 中有特征值时,称 l 为 Elkies 素数;否则,称 l 为 Atkin 素数(这可由 $\Phi_l^c(x, j(E))$ 的分裂形式来判断[13]).当 l 为 Elkies 素数时,Elkies 提出下面的方法:通过计算 Isogeny 映射的核(它是 ϕ 的一个特征子空间),得到可除多项式的一个因子 h_l, $\deg h_l = (l-1)/2$,再寻找合适的 λ 使 $\phi(P) = [\lambda](P) \bmod (y^2 = x^3 + ax + b, h_l(x))$,此 λ 即为 ϕ 的特征值,由 $t \equiv \lambda + \lambda^{-1}p \bmod l$ 可以计算出 $t \bmod l$.当 l 为 Atkin 素数时,Frobenius 映射的迹 $t \in F_l$ 满足方程 $t^2 \equiv p(\zeta + 2 + \zeta^{-1}) \bmod l$,其中 $\zeta \in F_{l^2}$ 是 r 次本原单位根(r 由 $\Phi_l^c(x, j(E))$ 的分裂形式决定[13]).因此,可以计算出 $t \bmod l$ 的一个可能值集合 T_l.SEA 算法是 Elkies 方法和 Atkin 方法的结合.这样,对一个素数 l,我们或者能计算出 $t \bmod l$ 的确切值,或者能计算出 $t \bmod l$ 的一个可能值集合.用 CRT 综合这些信息,再用某一尝试方法,比如 BSGS(baby step and giant step)算法,即可确定椭圆曲线的阶.

SEA 算法在实现中得到了进一步的发展.Morain 在文献[9]中提出,对 Elkies 素数 l, E 有两个 l 次同种映射,记 $E \xrightarrow{I_1} E_1$和$E \xrightarrow{I_2} E_2$, E_1 也有两个 l 次同种映射,其中之一为 I_1 的对偶同种映射 I_1^*,另一个记作 I_{11},如此下去,形成两个同种映射链:

$$E \xrightarrow{I_1} E_1 \xrightarrow{I_{11}} E_{11} \xrightarrow{I_{111}} \ldots$$

$$E \xrightarrow{I_2} E_2 \xrightarrow{I_{22}} E_{22} \xrightarrow{I_{222}} \ldots$$

由此对某个确定的 k,我们可以计算 l^k _th 可除多项式 f_{l^k} 的一个因子 g_{l^k},进而计算 $t \bmod l^k$,这里, $\deg g_{l^k} = l^{(k-1)}(l-1)/2$.这一算法思想又被称为同种圈方法.此外,设 $t \equiv a \bmod l^k$,则存在 $b \in \{0,1,\ldots,l^j - 1\}$ 使得 $t \equiv a + bl^k \bmod l^{k+j}$.根据这一性质,文献[14]提出了 Virtual 方法:设 T_{l^k} 为 $t \bmod l^k$ 的可能值集合,则 $T_{l^{k+j}} = \{a + bl^k \mid a \in T_{l^k}, 0 \le b \le l^j - 1\}$.

1.2 对同种圈方法的一点探讨

同种圈方法是 Morain 对 SEA 算法的进一步优化.首先,我们写出同种圈方法的具体步骤:

算法 **1**. 同种圈算法.

输入: (a,b), p,Elkies 素数 l,$n$max

输出: $t \bmod l^{n\max}$

1. 计算 E 的同种曲线 E_1 的方程 $\varepsilon_1(x,y)$ 及 *l_th* 可除多项式 f_l^E 的因子 $h_l(x)$.
2. 计算 Frobenius 映射的特征值 α_1.
3. for n=2 to $n$max
   3.1. 计算 E_{n-1} 的同种曲线 E_n 的方程 $\varepsilon_n(x,y)$ 及同种映射 $I_n: E_{n-1}\to E_n\ (I_n\neq I_{n-1}^*)$.
   3.2. 计算 E_{n-1} 的 *l_th* 可除多项式 $f_l^{E_{n-1}}$ 的因子 h_n.
   3.3. 计算 $h_n\cdot I_{n-1}\cdot\ldots\cdot I_2\cdot I_1$ 的分子,得到 $f_l^{E_n}$ 的因子 h.
   3.4. 寻找 $\lambda(0\le\lambda<l)$,使 $\alpha_n=\alpha_{n-1}+\lambda l^{n-1}$ 满足:

$$(x^p,y^p)=[\alpha_n](x,y)\bmod(y^2=x^3+ax+b,h(x)). \tag{*}$$

4. return: $t\equiv\alpha_n+p\alpha_n^{-1}\bmod l^n$.

分析上面的算法,其关键步骤在于第 3.4 步,在环 $F_p[x,y]/(y^2=x^3+ax+b,h(x))$ 中计算 x^p (或 y^p)的复杂度为 $O(\log p)$ 个此环中乘法,每个环中乘法要 $O(\deg^2 h(x))$ 个 F_p 中乘法,可见, $h(x)$ 的次数直接决定该算法的效率.

设 $g(x)\,|\,h(x)$, α_n 为满足式(*)的特征值,显然

$$(x^p,y^p)=[\alpha_n](x,y)\bmod(y^2=x^3+ax+b,g(x)) \tag{**}$$

仍成立.因此,若得到 $h(x)$ 的一个低次因子 $g(x)$,则通过式(**)检测的 α_n 全体构成 $t\bmod l^n$ 的一个可能值集合,可对它进行与 Atkin 素数相同的处理.在实际中,我们可以设法使惟一的 α_n 通过式(**)的检测.

由于 $h(x)$的次数为 $l^{n-1}(l-1)/2$,当 l 或 n 较大时,直接分解 $h(x)$是困难的,其关键步骤是要计算 $k(x)^{(p-1)/2}\bmod h(x)$ (其中 $k(x)$是一个随机多项式[15,16]),但在同种圈方法中我们可以稍作变换一下.从 $h(x)$的计算过程可以看出,若 g_n是 h_n的一个因子,则 $g_n\cdot I_{n-1}\cdot\ldots\cdot I_2\cdot I_1$ 的分子也是 $h(x)$的一个因子.同理,若 g_{n-1}是 $g_n\cdot I_{n-1}$ 分子的一个因子,则 $g_{n-1}\cdot I_{n-2}\cdot\ldots\cdot I_2\cdot I_1$ 的分子仍是 $h(x)$的一个因子.这样,对 $h(x)$求因子的问题就转化为对一系列次数较低的多项式求因子的问题.因为对较低次数的多项式(如次数小于 20)求小因子是容易的,若因子分解成功,对相同的 n, $\deg g(x)<\frac{1}{2}\deg h(x)$,故式(**)的计算时间应不大于(*)式计算时间的 1/4.当然,求低次数的多项式的因子也是要化时间的,只要在这一步的所化时间少于前面所省下来的时间,这一局部优化就有意义,在实际中对小于 40 的素数 l 采用这种技巧.综上所述,我们对同种圈算法作如下改进:

算法 2. 改进同种圈算法.

输入: (a,b),p,Elkies 素数 l,$d$max.

输出: (k, $t\bmod l^k$).

1. 计算 E_1 的方程 $\varepsilon_1(x,y)$ 及 $h_1(x)$;
2. 找特征值 α_1; $k\leftarrow 1$;
3. 计算 E_{k+1} 的方程 $\varepsilon_{k+1}(x,y)$ 及 $I_k: E_{k-1}\to E_k\ (I_k\neq I_{k-1}^*)$;
4. 计算可除多项式 $f_l^{E_{k+1}}$ 的因子 h_{k+1};
5. $h(x)\leftarrow h_{k+1}, i\leftarrow k+1$;
   if $\deg h(x)>d$max ,goto step 6;
   5.1. $g(x)\leftarrow h(x)$ 的最低次因子,if $i=0$, goto step 5.3;
   5.2. $h(x)\leftarrow g(x)\cdot I_i, i\leftarrow i-1$,goto step 5.1;
   5.3. 寻找 $\lambda(0\le\lambda<l)$,使 $\alpha_n=\alpha_{n-1}+\lambda l^{n-1}$ 满足
   $(x^p,y^p)=[\alpha_n](x,y)\bmod(y^2=x^3+ax+b,g(x))$;
   $k\leftarrow k+1$,goto step 3;
6. 计算 $T_{l^k}=\{\alpha_k+p\cdot\alpha_k^{-1}\bmod l^k\}$
7. Return: (k,T_{l^k}).

应用上述算法,我们可以根据数据的实际情况动态地调整 $t\bmod l^n$ 中 n 的取值,克服了原始算法中 $n$max 的主观确定性,从而更充分地利用小的 Elkies 素数的作用,提高了算法的效率.例如: $p=2^{160}+7, a=1, b=6, l=3$,在

Pentium11,450Mhz 微机上采用上述方法计算 $t \bmod l^5$ 总共仅需 3.14 秒($h(x)$有一个 27 次因子);而用原始方法,对 $l=3$,若取 $n$max=5,则 $\deg g_{l^5} = (l-1)*l^{(5-1)}/2 = 3^4 = 81$,因此,就整个算法的效率来说,计算 $t \bmod l^5$ 是不明智的.

1.3 SEA算法的实现

SEA 算法的实现过程分为两步,第 1 步利用 Elkies 方法、Atkin 方法、同种圈方法及 Virtual 方法,收集 Frobenius 映射迹 t 模一系列小素数的信息;第 2 步综合所得信息,得到 t 的一个候选值集合 T,利用 BSGS 算法从候选值集合 T 中选取 t 的确切值,从而计算出曲线的阶.第 2 步的时间消耗与$\#T$ 成正比,为了提高算法效率,必须控制$\#T$ 的大小,这就要求我们在第 1 步时恰当地处理好以下 3 个问题:

(1) 对 Elkies 素数,是否使用同种圈方法;

(2) 对 Atkin 素数 l,是否计算 $t \bmod l$ 的可能值集合,并应用于 BSGS 算法;

(3) 是否使用 Virtual 方法.

对此,文献[14]提出称为 Intelligent Choise System(简称 ICS)的决策系统.我们认为 ICS 使问题过于复杂化了,在实现 SEA 算法的过程中,我们总结了下面的方法.

我们首先简单分析一下上述 4 种方法的特点.前 3 种方法要利用一个新素数 l,因此首先要判断 l 是 Elkies 素数还是 Atkin 素数,其关键计算是 $x^p \bmod \Phi_l^c(x, j(E))$,$\deg \Phi_l^c(x, j(E))=l+1$.对 Elkies 素数,使用 Elkies 方法,我们可以得到 $t \bmod l$ 的确定值.若使用同种圈方法计算 $t \bmod l^k$,还有关键性计算 $(x^p, y^p) \bmod (y^2 = x^3 + ax + b, h(x))$;$\deg h(x) = l^{(k-1)}(l-1)/2$,即使使用改进的方法,还存在因子分解问题,因此,同种圈方法只适用于较小的 l.对 Atkin 素数,Atkin 方法可以较快地得到 $t \bmod l$ 的可能值集合 T_l,但当$\#T_l$ 较大时,利用 $t \bmod l$ 的信息会引起$\#T$ 迅速增大,因此,有时必须放弃这一信息.为此我们引入参量 $c_{l^k} = l^k / \#T_{k^k}$,不妨称为 c 参数.显然,应用于 BSGS 算法的 l 的 c 参数越大越好.Virtual 方法几乎可以忽略时间消耗,但 $c_{l^k} / c_{l^{k+j}} = 1$,因此会导致$\#T$ 迅速增大.

设当前已对 $l_1, l_2, ..., l_s$ 计算得到相应 $t \bmod l_i^{k_i}$ 可能值集合 $T_{l_i^{k_i}}$ $(1 \le i \le s)$,对上面的 3 个问题,本文的决策方案由 4 个数据项组成:(Imax,*VirBound*,*AtkinBound*,z),其含义如下:

(1) 只对小于 Imax 的 Elkies 素数使用同种圈方法;

(2) 若存在 $l_v (0 \le v \le s)$ 满足:

(a) $\prod_{1 \le i \le s} l_i^{k_i} < 4\sqrt{p}$,

(b) $l_v \times \prod_{1 \le i \le s} l_i^{k_i} > 4\sqrt{p}$,

(c) $l_v \times \prod_{1 \le i \le s} \#T_{l_i^{k_i}} < VirBound$,

对 l_v 使用 Virtual 方法求 $T_{l_v^{k_v}}$.

(3) 若 $\prod_{1 \le i \le s} \#T_{l_i^{k_i}} > AtkinBound$,对 z 个新素数 l,计算 $t \bmod l$ 的信息,按 c 参数从 $(l_i, T_{l_i^{k_i}})$ 中选择子集 $(l_i', T'_{l_i^{k_i}})$ 使满足 $\prod_{1 \le i \le s} l_i'^{k_i} > 4\sqrt{p}$,且 $\prod_{1 \le i \le s} \#T'_{l_i^{k_i}} < AtkinBound$,若存在这样的子集,进入算法第 2 步;否则重复这一过程.

该方案各数据项的具体取值可根据基域尺寸 p 从实践中总结.我们对 $GF(2^{160}+7)$上的 300 条随机椭圆曲线计算其阶并测试其时间消耗,表 1 的实验数据可以反映出该方案的效率.机器配置为 Penium Ⅱ 64M 内存,450MHz,win98.该方案 A 的参数选取(Imax,*VirBound*,*AtkinBound*,z)=$(23, 10^7, 3\times10^8, 2)$.其中第 2 组数据来自文献[14].

Table 1 Comparison of the efficiencies

表 1 效率的比较

Field①	Configuration②	Scheme③	Average time④ (s)	Minimum time⑤ (s)	Maximum time⑥ (s)
$2^{160}+7$	Pentium Ⅱ,450MHz	A	37.9	20.0	169.0
$2^{160}+7$	Pentium Ⅱ,300MHz	ICS	66.5	34.7	334.7

①基域,②配置,③方案,④平均时间,⑤最短时间,⑥最长时间.

2 随机选取素数阶和拟素数阶椭圆曲线

为了得到一条随机性很好的素数阶椭圆曲线,我们可以采用以下方法:随机选取曲线参数,利用 SEA 算法计算出它的阶,再对阶进行素性测试,直到得到理想的椭圆曲线为止.在实际应用中,许多情况下我们在利用 SEA 算法计算出阶之前,利用已有信息就可断定阶为合数.

假设 l 是 $\#E(F_p)$ 的一个素因子,则 $\#E(F_p)$ 必有一个 l 阶子群,从而 E 有一个 l 次有理 Isogeny 曲线,因此 l 为 Elkies 素数.显然,利用 SEA 算法计算出 $t \bmod l$ 之后,可以计算 $\#E(F_p)=p+1-t \bmod l$,从而判断 l 是否是 $\#E(F_p)$ 的因子.实际上,我们还可以做得更好.

命题 **1**. E 是 Fp 上一条椭园曲线,对 Elkies 素数 l,若 l_th 可除多项式的因子 $h(x)$ 在 F_p 中有根 x_p,且 $\left(\frac{x_p^3+ax_p+b}{p}\right)=1$,则 $l\mid\#E(F_p)$.

证明:因为 $h(x)$ 的所有根是 E 的 l 阶点的 x 坐标,且

$$\left(\frac{x_p^3+ax_p+b}{p}\right)=1,$$

所以 $(x_p,\sqrt{x_p^3+ax_p+b})\in E(F_p)$,即 $E(F_p)$ 有一个 l 阶点,故 $l\mid\#E(F_p)$. □

可以通过求公因子 $(x^p-x,h(x))$ 来判断 $h(x)$ 在 F_p 中是否有根,这里关键步骤是计算 $x^p \bmod h(x)$,而计算 $t \bmod l$ 的过程需要求解同余方程:

$$(x^p,y^p)=[\lambda](x,y) \bmod (y^2=x^3+ax+b,h(x)),$$

前者只是后者的中间步骤,其优越性是显然的.

算法 **3**. 随机选取素数阶椭圆曲线:

输入:有限域特征 p;

输出:椭圆曲线方程 $y^2=x^3+ax+b$,其阶为素数.

1. 随机选取曲线参数 $a,b\in F_p$, $ab\neq 0$, $\Delta=4a^3+27b^2\neq 0$.
2. 对 Atkin 素数,计算 $t \bmod l$ 的信息;
   对 Elkies 素数,判断除子多项式 $h(x)$ 在 F_p 中是否有根,
   若有根 x_p 且 $\left(\frac{x_p^3+ax_p+b}{p}\right)=1$,goto step 1;
   否则,计算 $t \bmod l$,若 $p+1\equiv t \bmod l$,goto step 1.
3. 计算 $\#E(F_p)$ 并进行素性测试,若 $\#E(F_p)$ 为合数,goto step 1;
   否则,输出椭园曲线方程 $y^2=x^3+ax+b$.

对算法 3 稍作修改,便可得到寻找拟素数阶椭圆曲线的算法.

算法 **4**. 随机选取拟素数阶椭圆曲线:

输入:有限域特征 p;

输出:椭圆曲线方程 $y^2=x^3+ax+b$,其阶为拟素数;

1. 随机选取曲线参数 $a,b\in F_p$, $ab\neq 0$, $\Delta=4a^3+27b^2\neq 0$.
2. 计算 $t \bmod l$ 的信息:
   (a) 若 l 为 Atkin 素数,计算 $t \bmod l$ 的信息;
   (b) 若 l 为 Elkies 素数且 l <4,计算 $t \bmod l$ 的值;
   (c) 若 l 为 Elkies 素数且 l >4,判断除子多项式 $h(x)$ 在 F_p 中是否有根,

若有根 x_p 且 $\left(\frac{x_p^3+ax_p+b}{p}\right)=1$,goto step 1;

否则计算 $t \bmod l$,若 $p+1\equiv t \bmod l$,goto step 1.

3. 计算 $\#E(Fp)$ 并判断 $\#E(Fp)$ 是否为拟素数,若 $\#E(F_p)$ 是拟素数,goto step 4;

否则 goto step 1;

4. 输出椭园曲线方程 $y^2=x^3+ax+b$.

比较算法 3 和算法 4 可以看出,算法 3 抛弃(a,b)对的速度比算法 4 要快得多,即在相同时间内,算法 3 要比算法 4 尝试的曲线要多得多,故算法 3 的效率要明显优于算法 4;再从资源的角度来看,也是素数阶椭圆曲线要好,因而在实际应用中建议使用算法 3.

我们在相同的机器配置下,对特征为 $p=2^{160}-47$ 的有限域 F_p,在 2.64 个小时内尝试了 2 617 条随机椭圆曲线,得到 10 条素数阶曲线,其中之一如下:

基域特征:$p=2^{160}-47$;

曲线参数:a=501099813759785732448264472652991642471458 5;

b=139107431847537640845966616831175078343078 2;

曲线的阶:$order$=146150163733090291820368512154002499497892591442 3;

随机点:(x,y)=(184977494696465582806354934130423153819697 1,

106085596239133023520814327459780822844835115629).

References:

[1] ANSI X9.62-1998. The Elliptic Curve Digital Signature Algorithm (ECDSA). Public Key Cryptography for the Financial Service Industry, American Bankers Association, 1998.

[2] ANSI X9.63-1999. Key Agreement and Key Transport Using Elliptic Curve Cryptography. Public Key Cryptography for the Financial Service Industry, American Bankers Association, 1999.

[3] IEEE P1363. Standards for Public-Key Cryptography. Institute of Electrical and Electronics Engineers, 1999.

[4] SEC1. Elliptic Curve Cryptography. Standards for Efficient Cryptography Group, 1999.

[5] FIPS 186-2. Digital Signature Standard. Federal Information Processing Standards, 2000.

[6] Schoof, R. Counting points on elliptic curves over finite fields. Journal of Theorie des Nombres de Bordeaux, 1995,7:219~254.

[7] Atkin, A.O. The number of points on an elliptic curve modulo a prime. Series of e-mail to the NMBRTHRY mailing list, 1992.

[8] Elkies, N.D. Elliptic and modular curves over finite fields and related computational issues. In: Buell, D.A., Teitelbaum, J.T., eds. Coputational Perspective on Number Theory. AMS/International Press, 1998. 21~76.

[9] Couveignes, J.-M., Morain, F. Schoof's algorithm and isogeny cycles. In: Adleman, L.M., Huang, M.D., eds. ANTS-I. LNCS 877, Springer-Verlag, 1994. 43~5.

[10] Lecier, R., Morain, F. Counting the number of points on elliptic curves over finite fields: strategy and performances. In: Guillou, L.C., Quisquater, J.J., eds. Proceedings of the EUROCRYPT'95. LNCS 921, Springer-Verlag, 1995. 79~94.

[11] Dewaghe, L. Remarks on the Schoof-Elkies-Atkin Algorithm. Mathematics of Computation 67, 1998. 1247~1252.

[12] Lercier, R. Finding good random elliptic curves for cryptosystems defined over F_2^n. In: Fumy, W., ed. Proceedings of the EURO-CRYPT'97. LNCS 1233, 1997. 379~392.

[13] Lehmann, F., Maurer, M., Müller, V., *et al*. Counting the number of point on elliptic curves over finite fields of characteristic greater then three. In: Adleman, L.M., Huang, M.D., eds. ANTS-I. LNCS 877, Springer-Verlag, 1994. 60~70.

[14] Tetsuya, Izu, Kogure, J., Noro, M., *et al*. Efficient Implementation of Schoof's Algorithm. In: Ohta, K., Pei, D.Y., eds. Proceedings of the ASIACRYPT'98. LNCS 1514, 1998. 66~79.

[15] Shoup, V. A new polynomial factorization algorithm and its implementation. Journal of Symbolic Computation, 1995,20:364~397.

[16] Cohen, H. A Course in Computational Algebraic Number Theory. Graduate Texts in Mathematics 138, New York: Springer-Verlag, 1996.

Efficient Implementation of SEA Algorithm*

ZHU Yue-fei[1,2], GU Chun-xiang[1], PEI Ding-yi[2]

[1](*Department of Network Engineering, Zhengzhou Information Engineering University, Zhengzhou* 450002, *China*);

[2](*State Key Laboratory of Information Security, Graduate School, The Chinese Academy of Sciences, Beijing* 100039, *China*)

E-mail: zyf0136@sina.com

http://www.home.is.ac.cn

Abstract: The core of choosing a secure elliptic curve for elliptic curve cryptosystems is the calculation of the order of a randomly selected elliptic curve. It is known that SEA (Schoof Elkies Atkin) algorithm is recently the most efficient method to calculate the orders of elliptic curves over F_p. Isogeny cycles method made by Morain is an important local optimized technique to improve SEA algorithm. In this paper, isogeny cycles method is enhanced, and a scheme of more optimal combination of the various techniques in SEA algorithm is provided. Furthermore, some discussions are made on how to speed up the selection of elliptic curves with prime order, and an efficient implementation of SEA algorithm over F_p is described.

Key words: elliptic curve; Frobenius endomorphism; SEA (Schoof Elkies Atkin) algorithm; isogeny cycle

* Received August 2, 2000; accepted December 5, 2001

Supported by the National Natural Science Foundation of China under Grant No.19931010; the National Grand Fundamental Research 973 Program of China under Grant No.G1999035804; the Foundation of He'nan Province of China for Distinguished Young Scholars under Grant No.0212001400