软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(4):1151−1164 [doi: 10.13328/j.cnki.jos.006116]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

椭圆曲线同源的有效计算研究进展*

黄 艳[1,2], 张方国[1,2]

[1](中山大学 计算机学院,广东 广州 510006)

[2](广东省信息安全技术重点实验室,广东 广州 510006)

通讯作者: 张方国, E-mail: isszhfg@mail.sysu.edu.cn

摘 要: 由于Shor算法可以在多项式时间内解决大整数分解以及离散对数问题,使得基于这些问题设计的经典的密码体制不再安全.目前涌现出许多后量子密码体制的研究,如基于格、基于编码、基于多变量和基于椭圆曲线同源的密码系统.相比于其他后量子密码体制,基于椭圆曲线同源的密码系统具有密钥尺寸短的优势,然而其实现效率不占优势.以两类基于超奇异椭圆曲线同源的密钥交换协议为基准,根据经典的椭圆曲线标量乘和双线性对的优化技巧,并结合椭圆曲线同源自身的一些特殊性质,分析优化这两类协议的可能性.与此同时,分类回顾了目前椭圆曲线同源的有效计算方面的已有进展,提出了该方向可进一步开展的研究工作.

关键词: SIDH;CSIDH;同源的计算;Montgomery 曲线;Edwards 曲线

中图法分类号: TP309

中文引用格式: 黄艳,张方国.椭圆曲线同源的有效计算研究进展.软件学报,2021,32(4):1151−1164. http://www.jos.org.cn/1000-9825/6116.htm

英文引用格式: Huang Y, Zhang FG. Research development on efficient elliptic curve isogenous computations. Ruan Jian Xue Bao/Journal of Software, 2021,32(4):1151−1164 (in Chinese). http://www.jos.org.cn/1000-9825/6116.htm

Research Development on Efficient Elliptic Curve Isogenous Computations

HUANG Yan[1,2], ZHANG Fang-Guo[1,2]

[1](School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 510006, China)

[2](Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou 510006, China)

Abstract: It is well known that Shor's algorithm can solve the integer factorization problem and the discrete logarithm problem in polynomial time, which makes classical cryptosystems insecure. Hence, more and more post-quantum cryptosystems emerge at present such as lattice-based, code-based, hash-based, and isogeny-based cryptosystems. Compared with other cryptosystems, the isogeny-based cryptosystems have the advantages of short key size. Nevertheless, it does not outperform other cryptosystems in respect of implementation efficiency. Based on two types of key exchange protocols from supersingular elliptic curve isogeny, this paper analyzes the possibility of optimizing two key exchange protocols according to the classical optimizations of elliptic curve scalar multiplication and pairing as well as some characteristics of elliptic curve isogeny. Meanwhile, the paper categorizes and reviews the current progress on efficient isogenous computations, and puts forward the further researches in this direction.

Key words: SIDH; CSIDH; isogenous computation; Montgomery curve; Edwards curve

椭圆曲线同源是两条椭圆曲线之间的一个非平凡的代数映射,它是一个群同态.根据 Tate 定理[1],定义在有限域$\mathbb{F}_q$上的两条椭圆曲线 E_1 和 E_2 同源当且仅当$\#E_1(\mathbb{F}_q)=\#E_2(\mathbb{F}_q)$.然而,给定有限域上的两条椭圆曲线 E_1 和 E_2,

* 基金项目: 国家重点研发计划(2017YFB0802500); 国家自然科学基金(61672550, 61972429); 广东省基础与应用基础研究重大项目(2019B030302008)

Foundation item: National Key Research and Development Program of China (2017YFB0802500); National Natural Science Foundation of China (61672550, 61972429); Guangdong Major Project of Basic and Applied Basic Research (2019B030302008)

收稿时间: 2019-11-15; 修改时间: 2020-05-28; 采用时间: 2020-07-09; jos 在线出版时间: 2020-07-27

满足$\#E_1(\mathbb{F}_q)=\#E_2(\mathbb{F}_q)$,找到 E_1 与 E_2 之间的同源是困难的.我们称该问题为标准的同源计算问题.

基于椭圆曲线同源的密码系统早期的研究主要集中在一般椭圆曲线(ordinary elliptic curve)上[2−4].根据 Childs 等人[5]的研究结果,计算一般椭圆曲线同源的时间复杂度为量子亚指数时间;根据 Biasse 等人[6]的研究结果,计算超奇异椭圆曲线(supersingular elliptic curve)同源的时间复杂度为量子指数时间.另外,Costello 等人[7]发现:在 Montgomery 曲线上,超奇异椭圆曲线同源的计算比一般椭圆曲线同源的计算效率更高.

因此,从安全性和计算效率的角度来看,对于目前具有抗量子计算攻击的椭圆曲线同源的密码体制的研究主要集中在超奇异椭圆曲线同源上.Jao 等人[8]提出了扩域上的基于超奇异椭圆曲线同源的密钥交换协议(SIDH).之后,Azarderakhsh 等人[9]将基于 SIDH 的加密方案和密钥封装协议提交到美国 NIST,参与后量子密码方案的候选,并成功进入第 2 轮.然而,其实现的效率相比于基于纠错码[10,11]和基于格[12,13],均不占优势.Yoo 等人[14]和 Galbraith 等人[15]提出了基于超奇异椭圆曲线同源的签名方案,这两类签名方案均采用认证结构结合 FS[16]转换或者 U 转换[17]来加以构造,效率相比于基于格的[18,19]和基于哈希函数[20,21]的签名方案也不占优势,这主要是由于椭圆曲线同源的计算效率不高所致.

目前,对于 SIDH 的优化计算主要从以下 3 个角度来展开:探索适合的域的形式;借助不同曲线形式、不同坐标形式的优势;利用一些特殊技巧,如加法链、Weil 限制和双线性对等去优化.

以上基于椭圆曲线同源的密码方案均是在扩域上进行的,Castryck 等人[22]提出了基域上的基于超奇异椭圆曲线同源的密钥交换协议(CSIDH),其算法的实现效率相比在扩域上的 SIDH 提高了很多,然而其运行时间会随着密钥的变化而变化,因此不能抵抗侧信道攻击.

目前,对于 CSIDH 算法的优化主要从以下 3 个角度来实施:通过增加冗余,使其算法的运行时间为常数时间;借助不同曲线形式和坐标形式的优势;利用一些特殊技巧,如探索有效的基点生成算法、加法链和最优策略等优化计算.

本文第 1 节阐述椭圆曲线同源的计算公式、SIDH 协议、CSIDH 协议以及优化 SIDH 和 CSIDH 的可能性.第 2 节和第 3 节分别讨论目前所提出的优化 SIDH 和 CSIDH 的各种有效技巧.第 4 节探讨 SIDH 和 CSIDH 的其他可能优化的问题.

1 Vélu 公式、SIDH 协议以及 CSIDH 协议

本节主要描述有限域上计算椭圆曲线同源的 Vélu 公式、SIDH 协议以及 CSIDH 协议,并分析实现这两个协议的效率影响因素.

1.1 同源和Vélu公式

根据文献[23],两条椭圆曲线之间的同源具有有理函数表达式.特别地,对于一个可分的同源ϕ,其核的阶$\#Ker\phi$为同源ϕ的次数,ϕ的表达式由其核唯一决定.即:给定定义在有限域$\mathbb{F}_q$的椭圆曲线 E 上的一个子群,Vélu 给出变换如下:

$$\phi: E \to E'$$
$$(x_P, y_P) \to (x_P + \sum_{Q \in G \setminus O}(x_{P+Q} - x_Q), y_P + \sum_{Q \in G/O}(y_{P+Q} - y_Q))$$

其中,群 G 中的点在ϕ的作用下均映射到 E'中的单位元 O.根据这一变换,可以推导出ϕ在 Weierstrass 曲线形式下的有理函数表达式.对于其他曲线形式的 Vélu 公式,也可类似地给出或利用与 Weierstrass 曲线的同构进行推导.

表 1 给出了目前已有的 Weierstrass 曲线、Montgomery 曲线、Edwards 曲线、Huff 曲线、Hessian 曲线和 Jacobian quartic 曲线上的ℓ-同源公式和相应的同源曲线公式.假设群 G 的生成元为点 P,其中,阶为ℓ.在 Montgomery 曲线上,注意到 $x_{[i]P} = x_{[\ell-i]P}$, 其中, $i = 1,2,...,\frac{\ell-1}{2}$.在同源计算中,可以利用这一性质简化公式的表达形式.ℓ-同源的像的 x 坐标只与原像中的 x 坐标和群 G 中的点的 x 坐标有关,且ℓ-同源曲线系数也只与群 G 中的 x 坐标和原像曲线系数有关.因此,在具体实现时,为了避免求逆,可以只利用坐标$(X:Z)$(其中,$x=X:Z$)就能够计算

Montgomerey 曲线上的同源和同源曲线.由 Edwards 曲线可以发现,ℓ-同源和ℓ-同源曲线的计算公式只与坐标 w 和曲线系数有关,因此,在具体实现时,为了避免求逆,利用坐标$(W_E:Z_E)$(其中, $w=\mathrm{d}x_E^2y_E^2$)即可完成这些计算.对于 Huff 曲线和 Hessian 曲线,目前只给了射影坐标$(X:Y:Z)$下的ℓ-同源和ℓ-同源曲线公式.对于其他坐标形式的优化,还需我们作更进一步的研究.

Table 1 Vélu formulae between different curve forms

表 1 不同曲线形式上的 Vélu 公式

曲线形式	ℓ-同源公式(ℓ=2s+1)
Weierstrass 曲线[23]: $y^2=x^3+ax^2+b$	$\phi(x,y)=\left(x+\sum_{i=1}^{s}\left(\frac{2(3x_{[iP]}^2+a)}{x-x_{[iP]}}-\frac{4y_{[iP]}^2}{(x-x_{[iP]})^2}\right),\ y+\sum_{i=1}^{s}\left(\frac{8y_{[iP]}^2y}{(x-x_{[iP]})^3}+2(3x_{[iP]}^2+a)\frac{(y-y_{[iP]}-(3x_{[iP]}{}^2+a)(-2y_{[iP]}))}{(x-x_{[iP]})^2}\right)\right)$
Montgomery[24] 曲线: $by^2=x^3+ax^2+x$	$\phi(x,y)=(f(x),yf'(x))$,其中, $f(x)=x\prod_{i=1}^{s}\left(\frac{xx_{[iP]-1}}{x-x_{[iP]}}\right)^2$
Edwards 曲线[25]: $x^2+y^2=1+dx^2y^2$	$\phi(w)=w\prod_{i=1}^{s}\frac{(w-w_i)^2}{(1-ww_i)^2}$
Huff 曲线[26]: $x(ay^2-1)=y(bx^2-1)$	$\phi(x,y)=\left(x\prod_{i=1}^{s}\frac{x^2-x_{[iP]}^2}{x_{[iP]}^2(1-b^2y_{[iP]}^2x^2x_{[iP]}^2)},y\prod_{i=1}^{s}\frac{y^2-y_{[iP]}^2}{y_{[iP]}^2(1-a^2y_{[iP]}^2y^2)}\right)$
Hessian 曲线[27]: $ax^3+y^3+1=dxy$	$\phi(x,y)=\left(x\prod_{i=1}^{s}\frac{(x-x_{[iP]}y_{[iP]}y^2)(y_{[iP]}^2x-x_{[iP]}y^2)}{(ax_{[iP]}^2xy-y_{[iP]})^2},y\prod_{i=1}^{s}\frac{(y_{[iP]}^2y-ax_{[iP]}x^2)(y-ax_{[iP]}y_{[iP]}x^2)}{(ax_{[iP]}^2xy-y_{[iP]})^2}\right)$
Jacobi quartic 曲线[28]: $y^2=dx^4+2ax^2+1$	$\phi(x,y)=\left((-1)^s\frac{x}{A^2}\prod_{i=1}^{s}\frac{y_{[iP]}^2x^2-x_{[iP]}^2y^2}{1-dx_{[iP]}^2x^2},\ \frac{y}{B^2}\cdot\prod_{i=1}^{s}\frac{y_{[iP]}^2y^2(1+dx_{[iP]}^2x^2)^2-((2ax_{[iP]}+2dx_{[iP]}^3)x+(2adx_{[iP]}^3+2dx_{[iP]})x^3)^2}{(1-dx_{[iP]}x^2)^4}\right)$ 其中, $A=\prod_{i=1}^{s}x_{[iP]},B=\prod_{i=1}^{s}y_{[iP]}$

Table 1 Vélu formulae between different curve forms (Continued)

表 1 不同曲线形式上的 Vélu 公式(续)

曲线形式	ℓ-同源曲线公式	ℓ-同源曲线/ℓ-同源计算量
Weierstrass[23]: $y^2=x^3+ax^2+b$	$a'=a-5\left(\sum_{i=1}^{s}2(3x_{[iP]}^2+a)\right),b'=b-7\left(\sum_{i=1}^{s}(4y_{[iP]}^2+2x_{[iP]}(3x_{[iP]}^2+a))\right)$	射影坐标$(X:Y:Z)$ $(s^2+20s)M+4sS$/ $(s^2+5s)M+2sS$
Montgomery[24] 曲线: $by^2=x^3+ax^2+x$	$a'=(6A-6B+a)C,b'=bC$,其中, $A=\sum_{i=1}^{s}x_{[iP]},B=\sum_{i=1}^{s}\frac{1}{x_{[iP]}},C=\prod_{i=1}^{s}x_{[iP]}$	射影坐标$(X:Z)$ $5\ell M+\ell S/4sM+2S$
Edwards 曲线[25]: $x^2+y^2=1+dx^2y^2$	$d'=d^{\ell}\prod_{i=1}^{s}\frac{(w_i+1)^s}{4^4}$	射影坐标$(W:Z)$ $(2s+2)M+6S/4sM+2S$
Huff 曲线[26]: $x(ay^2-1)=y(bx^2-1)$	$a'=a^{\ell}\left(\prod_{i=1}^{s}y_{[iP]}\right)^4,b'=b^{\ell}\left(\prod_{i=1}^{s}x_{[iP]}\right)^4$	射影坐标$(X:Y:Z)$ $(4s+4)M+8S$/ $(8s+3)M+3S$
Hessian 曲线[27]: $ax^3+y^3+1=dxy$	$a'=a^{\ell},d'=\frac{(1-4s)d+6A+6B}{C}$,其中, $A=\sum_{i=1}^{s}\frac{1}{x_{[iP]}y_{[iP]}},B=\sum_{i=1}^{s}\frac{y_{[iP]}^2}{x_{[iP]}},C=\prod_{i=1}^{s}\frac{x_{[iP]}^2}{y_{[iP]}}$	射影坐标$(X:Y:Z)$ $5\ell M+2\ell S/15sM+4S$
Jacobi quartic 曲线[28]: $y^2=dx^4+2ax^2+1$	$a'=a+\sum_{i=1}^{s}\lambda_i,d'=\mu-4\left(a+\sum_{i=1}^{s}\lambda_i\right)\gamma$,其中, $\lambda_i=2dx_{[iP]}^2+2a-\left(\frac{2ax_{[iP]}+2dx_{[iP]}^3}{y_{[iP]}}\right)^2+4dx_{[iP]},\mu=d+\sum_{i=1}^{s}(4a\lambda_i+\lambda_i^2)+\sum_{j=1,j\neq i}^{s}\sum_{i=1}^{s}4\lambda_i\lambda_j,\gamma=\sum_{i=1}^{s}\left(dx_{[iP]}^2+2a-\frac{y_{[iP]}}{x_{[iP]}^2}\right)$	射影坐标$(X:Y:Z)$ $(s^4+s^2)M+4sS$/ $25sM+10sS$

表 1 中,通过比较不同曲线形式上的ℓ-同源和ℓ-同源曲线的计算量可知:在 Montgomery 曲线和 Edwards 曲

线的同源计算量相同,且均比在 Huff 曲线、Hessian 曲线和 Jacobi quartic 曲线的计算更具优势.对于同源曲线的计算,在 Edwards 曲线上比在 Montgomery 曲线、Huff 曲线、Hessian 曲线和 Jacobi quartic 曲线上的计算都更有优势.

1.2 SIDH协议

Jao 等人[9]首次提出了基于超奇异椭圆曲线同源的密钥交换协议.该协议的具体描述如下.

假设 $p=\ell_A^{e_A}\ell_B^{e_B}f\mp1$ 是一个大素数,其中,ℓ_A 和 ℓ_B 均为小素数;e_A 和 e_B 满足 $\ell_A^{e_A}\approx\ell_B^{e_B}$;$f$ 为一个小因子,使得 p 为素数.在有限域 $\mathbb{F}_{p^2}$ 中选择一个超奇异椭圆曲线 E_0 作为起始曲线,其基数为 $\#E_0(\mathbb{F}_{p^2})=(\ell_A^{e_A}\ell_B^{e_B})^2$.

假设 Alice 和 Bob 想进行密钥交换获得一个共同的密钥,首先,Alice 和 Bob 分别产生阶为 $\ell_A^{e_A}$ 的独立点$\{P_A, Q_A\}$和阶为 $\ell_B^{e_B}$ 的独立点$\{P_B,Q_B\}$.Alice 选择 $m_A\in\{1,2,...,\ell_A^{e_A-1}\}$,计算在核$\langle P_A+m_AQ_A\rangle$下的同源$\phi_A$:$E_0\to E_A$ 以及点 P_B 和 Q_B 的同源值$\phi_A(P_B)$和$\phi_A(Q_B)$,将$\phi_A(P_B)$、$\phi_A(Q_B)$、E_A 发送给 Bob.同时,Bob 进行类似的操作,计算在核$\langle P_B+m_BQ_B\rangle$下的同源$\phi_B$:$E_0\to E_B$ 以及点 P_A 和 Q_A 的同源值$\phi_B(P_A)$和$\phi_B(Q_A)$,将$\phi_B(P_A)$、$\phi_B(Q_A)$、E_B 发送给 Alice.

在密钥确立阶段,Alice 计算在核$\langle\phi_B(P_A)+m_A\phi_B(Q_A)\rangle$下的同源$\phi_{A'}$:$E_B\to E_{AB}$,获得曲线 E_{AB}.Bob 进行类似的操作,计算在核$\langle\phi_A(P_B)+m_B\phi_A(Q_B)\rangle$下的同源$\phi_{B'}$:$E_A\to E_{BA}$,获得曲线 E_{AB}.最后,Alice 和 Bob 获得共同的 j 不变量,即 $j(E_{AB})=j(E_{BA})$.协议过程如图 1 所示.定义 A 和 B 为 Alice 和 Bob 的标识,sID 为唯一的会话标识.

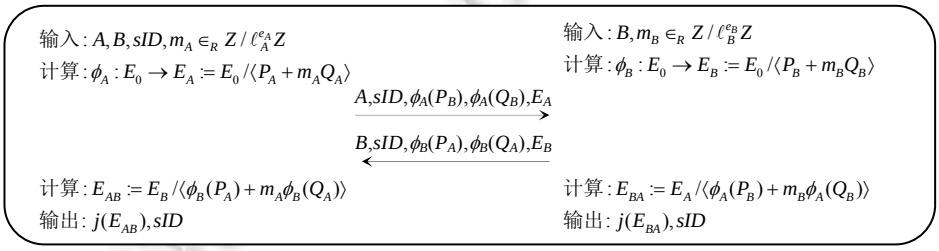

Fig.1 Key-exchange protocol based on supersingular elliptic curve isogeny

图 1 基于超奇异椭圆曲线同源的密钥交换协议

通过对上述协议的描述,我们分析影响 SIDH 有效实现的因素主要有:

(1) 有限域的类型以及基本代数运算.在保证安全度的前提下,可以选择在基域$\mathbb{F}_p$ 或者在扩域 $\mathbb{F}_{p^2}$ 中实现.一般来说,尽可能多地选择在基域中进行优化计算.另外,任何加速底层的有限域的基本算术方法都能加快 SIDH 的实现;

(2) 椭圆曲线中标量乘的计算.注意到:在 SIDH 中,Alice 和 Bob 在初始阶段生成同源的核生成点时都需要进行标量乘的计算.同时,在同源的复合计算过程中,也涉及到核生成点的计算,因此也用到了标量乘的计算.从而,有效的标量乘计算可加速 SIDH 的计算;

(3) 曲线以及坐标的类型.不同的曲线模型以及相应的不同坐标形式,其上的点加、倍点、同源和同源曲线的代数操作的计算量是不同的,因此,选择一条合适的曲线形式和坐标形式,使其具备有效的代数操作,将能够在很大程度上加快 SIDH 的有效实现;

(4) 同源和同源曲线的计算.在 SIDH 中,Alice 和 Bob 均需要计算ℓ^e-同源和同源曲线.对于ℓ^e-同源的计算,需要进行 e 次ℓ-同源的复合.显然,复合次数越少,计算速度越快.对于同源曲线的计算,当ℓ较大时,直接利用同源曲线公式计算,效率较低.因此,探索有效的ℓ^e-同源和同源曲线的计算也会促进 SIDH 的有效实现;

(5) 压缩公钥和恢复公钥的计算量.Alice 和 Bob 在利用 SIDH 进行通信时,彼此传递的信息有同源曲线和两个独立点的同源值,需要 $12\log_2 p$ 的通信量.而这些通信量可以通过一些压缩算法更进一步地加以减少,从而降低公钥的尺寸规模.同时,其压缩算法和解压算法的效率也会在一定程度上影响到 SIDH

的有效实现.

1.3 CSIDH

Castryck 等人[22]提出了定义在基域$\mathbb{F}_p$上的可交换的密钥交换协议 CSIDH,其具体过程如下.

令 $p=4\ell_1\ell_2\ldots\ell_n-1$ 为一个素数,其中,$\ell_i(i=1,\ldots,74)$为各自不同的素数.E 为定义在$\mathbb{F}_p$上的具有自同态环$\mathcal{O}$的超奇异椭圆曲线,其中,$\mathcal{O}$为一个虚二次域的 Order,$\pi\in\mathcal{O}$为一个 Frobenius 自同态映射,$\mathcal{E}\ell\ell_p(\mathcal{O},\pi)$为定义在$\mathbb{F}_p$上满足当$\pi$对应于曲线的$\mathbb{F}_p$-Frobenius 时具有与$\mathcal{O}$同构的自同态环的曲线的集合.任何一个类群 $cl(\mathcal{O})$中的元素$[\boldsymbol{a}]$通过同源作用在$\mathcal{E}\ell\ell_p(\mathcal{O},\pi)$中的曲线 E,即$[\boldsymbol{a}]E$.假设 Alice 和 Bob 想交换一个密钥:在密钥生成段,Alice 选择一个理想类$[\boldsymbol{a}]$,计算 $E_A=[\boldsymbol{a}]E$,将 E_A 发给 Bob.Bob 选择一个理想类$[\boldsymbol{b}]$,计算 $E_B=[\boldsymbol{b}]E$,将 E_B 发给 Alice;在密钥确立阶段,一旦收到对方的公钥,Alice 计算$[\boldsymbol{a}]E_B$,Bob 计算$[\boldsymbol{b}]E_A$.由于类群具有可交换的性质,因此 Alice 和 Bob 均可以计算共享的密钥,即$[\boldsymbol{a}][\boldsymbol{b}]E=[\boldsymbol{a}]E_B=[\boldsymbol{b}]E_A$.

对于 CSIDH 的实现,主要是计算$[\boldsymbol{a}]E$ 的过程,如下面算法 1 所示.

算法 1[22].

输入:超奇异椭圆曲线 E_0 和理想类 $[\boldsymbol{a}]=[\boldsymbol{l}_1^{e_1}\ldots\boldsymbol{l}_n^{e_n}]$,其中,$e_i\in\{-5,\ldots,5\}$;

输出:曲线 E_A,满足 $[\boldsymbol{l}_1^{e_1}\ldots\boldsymbol{l}_n^{e_n}]E=E_A$.

1. 当 $e_i\neq0$:
   - 1.1. 机选择 $x\in\mathbb{F}_p$,令点 P 的横坐标为 x;
   - 1.2. 令 $s\leftarrow+1$,若 x^3+ax^2+x 在$\mathbb{F}_p$为一个平方;否则,$s\leftarrow-1$;
   - 1.3. 令 $S=\{i|e_i\neq0,sign(e_i)=s\}$.若 $s=\varnothing$,重新选择 x;
   - 1.4. 令 $k\leftarrow\prod_{i\in S}\ell_i$, 计算 $Q\leftarrow\left[\frac{p+1}{k}\right]P$;
   - 1.5. For $i\in S$:
     - 1.5.1. 计算 $R\leftarrow\left[\frac{k}{\ell_i}\right]Q$.若 $R=O$,则跳过 i;
     - 1.5.2. 计算在核$\langle R\rangle$下的同源$\varphi:E\rightarrow E_a:y^2=x^3+ax^2+x$;
     - 1.5.3. 令 $a\leftarrow a,Q\leftarrow\varphi(Q),k\leftarrow\frac{k}{\ell_i},e_i\leftarrow e_i-s$.
2. 返回 a

对于 CSIDH 的优化,主要考虑算法 1 的优化,有以下几种可能.

(1) 基点 P 的选取.算法 1 中,点 P 的选取与密钥的正负有直接的联系:当密钥均为正时,随机选取的点均在$\mathbb{F}_p$上;当密钥均为负时,随机选取的点均在$\mathbb{F}_{p^2}$上.随机选取不能保证每次都成功选到合法的点,从而在一定程度上影响到算法的实现效率.因此,设计有效的基点生成算法,将在一定程度上优化算法 1 的实现;

(2) 标量乘的计算.在算法 1 的步骤 1.4 和步骤 1.5.1,需要进行标量乘计算,而这些标量乘的计算都形如$\ell_1\ell_2\ldots\ell_nP$,其中,$\ell_1,\ell_2,\ldots,\ell_n$ 均为不同的素数,对于这种形式的标量乘的优化,也将能够提高算法 1 的实现效率;

(3) 同源的计算和同源曲线的计算.对于算法 1 中计算的同源是一些不同素数次的同源的复合,对于这样的同源是否有类似于 SIDH 的最优策略,也是值得研究的一个方面.另外,考虑到 CSIDH 中需要计算的同源的次数相对于 SIDH 中的次数均要大(针对素数次同源),计算效率也比较低,对这类同源的优化也将在很大程度上促进 CSIDH 的优化.对于同源曲线的计算,注意到算法 1 中并没有类似 SIDH 中需

要计算的同源点来恢复同源曲线,因此,对同源曲线公式的优化也是优化算法 1 的一个方面;

(4) 常数时间的算法.注意到:算法 1 需要计算的同源的个数依赖于密钥,不能抵抗侧信道攻击.因此,如何设计一个有效的常数时间的算法来计算$[\boldsymbol{a}]E$,也是目前研究的一个热点问题.

2 SIDH 实现的改进概述

SIDH 目前的实现主要是在 Montgomery 曲线上坐标$(X_M:Z_M)$来实现的,可参见文献[7,24].下面将从不同理论角度来综述目前已发现的 SIDH 实现的改进技巧.

2.1 加快有限域中的基本运算

对于优化有限域中的基本运算,目前的研究主要包括 3 个方面:优化有限域中的基本代数运算、减少有限域的尺寸、将扩域中的基本代数运算转化到基域中进行计算.

对于第 1 个方面,Koziel 等人[25]借助加法链的方法,优化有限域$\mathbb{F}_{p^2}$中的平方根和求逆运算.Joppe 等人[26]通过探索有限域特征的特殊素数形式$p=2^xf^y-1$(其中,f为素数),利用 Montgomery 归约算法提高有限域中模运算的速度,从而提高基本的模加、模减、模乘和模逆运算.Seo 等人[27]在 64 比特的 ARM 上对有限域中的模加、模减和模乘都进行了优化.Costello 等人[28]考虑在进行密钥交换协议时,若一方的计算速度比较快,则设定有限域的特征为 $p=2^ef-1$,或者 $p=2^n-2^m-1$,或者满足 $p+1$ 和 $p-1$ 含有小素因子的素数 p,且这些小素因子的乘积到达相应的安全级别,进而利用 $p+1$ 阶扭点和 $p-1$ 阶扭点,加快 SIDH 中 Alice 的实现速度;

对于第 2 个方面,Flynn 等人[29]利用在同等安全级别下亏格为 2 的基于椭圆曲线同源的密钥交换协议要求的有限域的特征 p 的尺寸比在亏格为 1 的有限域的特征 p 要小的优势,提出了在亏格为 2 的扩域$\mathbb{F}_{p^2}$上实现超椭圆曲线同源的密钥交换协议;

对于第 3 个方面,Costello 等人[30]借助在同样的特征下,基域$\mathbb{F}_p$中的模代数运算比在扩域$\mathbb{F}_{p^2}$中要快,即:

$$A_{\mathbb{F}_{p^2}}=2A_{\mathbb{F}_p},S_{\mathbb{F}_{p^2}}=2S_{\mathbb{F}_p}+M_{\mathbb{F}_p}+4A_{\mathbb{F}_p},M_{\mathbb{F}_{p^2}}=3M_{\mathbb{F}_p}+5A_{\mathbb{F}_p},I_{\mathbb{F}_{p^2}}=I_{\mathbb{F}_p}+2M_{\mathbb{F}_p}+2S_{\mathbb{F}_p},$$

其中,$A_{\mathbb{F}_{p^2}}$、$S_{\mathbb{F}_{p^2}}$、$M_{\mathbb{F}_{p^2}}$和$I_{\mathbb{F}_{p^2}}$分别为扩域中的加法、平方、乘和求逆运算,$A_{\mathbb{F}_p}$、$S_{\mathbb{F}_p}$、$M_{\mathbb{F}_p}$和$I_{\mathbb{F}_p}$分别表示基域中的加法、平方、乘和求逆运算,将扩域$\mathbb{F}_{p^2}$中亏格为 1 的椭圆曲线的 Kummer 线(将椭圆曲线 E 的商 $E/\langle\pm1\rangle$称为 Kummer 线)上 2-同源的计算通过 Weil 限制的方法转化到基域$\mathbb{F}_p$上亏格为 2 的超椭圆曲线的 Kummer 面(对于亏格为 2 的超椭圆曲线 C,J_C 为其雅克比,其商 $J_C/\langle\pm1\rangle$为 Kummer 面)上的(2,2)-同源的计算.

2.2 优化椭圆曲线中的标量乘计算

SIDH 中涉及到的标量乘主要的形式包括 $P+kQ$ 和ℓR,其中,P、Q、R 均为椭圆曲线上的点,k、ℓ均为正整数.对于 $P+kQ$ 的计算,主要是利用 Ladder 算法[9]进行优化计算.Faz-Hernendez 等人[31]给出了一种左右 Ladder 算法,并借助预计算的技巧进行了更进一步的优化.对于ℓR 的计算,目前的方法主要是利用加法链进行优化计算[29].

2.3 探索适合的曲线形式、坐标形式

在不同的曲线模型上,利用不同坐标形式计算点加、倍点、同源以及同源曲线的计算量是不同的.探索一个最适合的曲线模型以及相应的坐标形式,使其在上面这些计算的耗费量最小,也是一种非常重要的优化 SIDH 实现的方式.Montgomery 等人[32]最早提出了在 Montgomery 曲线上仅利用坐标$(X_M:Z_M)$就可以进行倍点和标量乘的计算.De Feo 等人[33]给出了在 Montgomery 曲线上坐标$(X_M:Z_M)$的 3-同源和 4-同源公式.利用这些基本运算,Costello 等人[6]在 Montgomery 曲线上优化了 SIDH 的实现.另外,Costello 等人[24]又利用计算同源和同源曲线之间共用的 3 阶点、4 阶点继续对 3 倍点、3-同源和 4-同源以及相应的同源曲线进行优化.Renes 等人[34]给出了核生成点不在(0,0)的 2-同源公式,通过 1 次复合该 2-同源公式,则很容易得到核生成点不在$\left(1,\mp\sqrt{\frac{a+2}{b}}\right)$的 4-同

源公式,其中,a,b 为 Montgomery 曲线的系数.与 DeFeo 等人[33]的方法相比较,该方法避开了求根号以及额外 8 阶扭点的计算.

注意到,Edwards 曲线与 Montgomery 曲线之间存在着双有理关系[35]:

$$(x_E, y_E) = \left(\frac{x_M}{y_M}, \frac{x_M - 1}{x_M + 1}\right).$$

即,Edwards 曲线上的坐标 y_E 完全可以利用 Montgomery 曲线上的坐标 x_M 表示.Kim 等人[36]利用该双有理关系推导出在 Edwards 曲线坐标$(Y_E:Z_E)$下的 4-同源以及 4-同源曲线公式,并发现:在该坐标下实现 SIDH 的效率比在 Montgomery 曲线坐标$(X_M:Z_M)$下实现 SIDH 的效率要稍微高一点.除了在坐标$(Y_E:Z_E)$下可以优化计算外,Farashahi 等人[37]也探索了新的 Edwards 曲线上坐标$(W_E:Z_E)$对应的倍点和点加公式,发现其计算量与在坐标$(Y_E:Z_E)$下相同.另外,Kim 等人[38]研究了在坐标$(W_E:Z_E)$下的奇数次同源公式和相应的同源曲线公式,其同源公式的计算量与在 Montgomery 曲线上的计算量也是相同的,同源曲线的计算量相比于在 Montgomery 曲线上要有优势.然而,对于偶数次同源在 Edwards 曲线坐标$(W_E:Z_E)$上的公式,目前还没有被提出.

除了以上对于 Montgomery 曲线和 Edwards 曲线的不同坐标形式的同源和同源曲线公式的研究,Moody 等人[39]还给出了 Huff 曲线下的同源和同源曲线公式,Dang 等人[40]给出了 Hessian 曲线下的奇数次同源和同源曲线公式,Xu 等人[41]给出了 Jacobi quartic 曲线下的同源和同源曲线公式.然而,对于在这几种曲线上的点加、倍点以及同源的计算与在 Montgomery 和 Edwards 曲线的相应的计算相比,计算的耗费量差异较大,相应的优化还没有给出.此外,对于其他曲线形式,如对 Jacobi intersections[42]的同源的研究也没有给出.

2.4 优化同源曲线的计算公式

注意到,同源曲线的优化计算也可以加快 SIDH 的实现,因此,Costello 等人[24]提出了 3 种不同的方法来计算奇数次同源曲线,分别是:

(1) 利用奇数次同源曲线公式来恢复曲线系数.

在 SIDH 中,Alice 和 Bob 最终共享的密钥是椭圆曲线的 j 不变量,当曲线为 Montgomery 曲线$(by^2=x^3+ax^2+x)$时,其 j 不变量为 $j=\dfrac{256(a^2-3)^3}{a^2-4}$,只与系数 a 有关,系数 b 不参与计算.因此,实现 SIDH 过程中也只需考虑利用表 1 的公式计算同源曲线的系数 a.

(2) 利用额外的 2 阶扭点的同源值来恢复曲线系数.

在 Montgomery 曲线上,当给定初始曲线时,即 $by^2=x^3+ax^2+x$,其二阶扭点很容易获得.即令 $x^3+ax^2+x=0$,利用韦达定理可得两个根,分别为 x_1 和 x_2,从而有二阶扭点分别为$(0,0)$、$(x_1,0)$和$(x_2,0)$,计算$(x_1,0)$或者$(x_2,0)$在任意奇数次同源ϕ的值依然为 2 阶扭点,即$(\phi(x_1),0)$或者$(\phi(x_2),0)$,通过这两个同源值,反过来由公式:

$$a'=\frac{-(\phi^2(x_1)+1)}{\phi(x_1)}\text{或者}a'=\frac{-(\phi^2(x_2)+1)}{\phi(x_2)}$$

即可恢复同源曲线.

(3) 利用固定的 3 个点的同源值恢复曲线系数.

在 SIDH 中需要计算两个独立点 P 和 Q 以及 $P-Q$ 的同源值.而当知道这 3 个点的横坐标时,利用公式:

$$a=\frac{(1-x_Px_Q-x_Px_{P-Q}-x_Qx_{P-Q})^2}{4x_Px_Qx_{P-Q}}-x_P-x_Q-x_{P-Q}$$

即可恢复同源曲线.

方法(1)、方法(2)的优点是适合计算小次数同源曲线,缺点是会随着同源次数的增加而增加计算量.方法(3)适合计算较大次数的同源曲线,并且计算量是固定的,均为 $8M+5S$,前提是要有额外的同源点.

表 2 给出了利用以上 3 种方法分别计算 3-同源曲线和 19-同源曲线的计算量.计算 3-同源曲线,利用方法(1)的计算量最小;计算 19-同源曲线,利用方法(3)的计算量最小.

Table 2 Compute 3-isogeny curve and 19-isogeny curve

表 2 计算 3-同源曲线和 19-同源曲线

方法	利用奇数次同源公式	利用 2 阶扭点	利用 3 个点的同源值
适用情形	小次数同源	小次数同源	大次数同源
3-同源曲线	2*M*+3*S*	4*M*+2*S*	8*M*+5*S*
19-同源曲线	45*M*+20*S*	36*M*+2*S*	8*M*+5*S*

2.5 减少ℓ^e-同源的循环次数

对于ℓ^e-同源的计算,主要是将其分解为 e 个ℓ-同源的复合.De Feo 等人[33]提出了 3 种方法,分别是基于同源的方法、基于标量乘的方法和最优策略算法.其中,基于同源的方法是在复合过程用计算同源的方式去计算每次同源的核生成点;而基于标量乘的方法是在复合过程中用基于标量乘的方式计算每次同源的核生成点;最优策略算法是结合前两种方法的优势提出的一种新的方法,通过比较每次复合中标量乘计算和同源计算的耗费量,并结合最优策略的性质,即一个策略是最优的当且仅当其分解为两个最优子策略,得到最优路径,利用该路径来计算每次同源的核生点.3 种方法相比较而言,第 3 种方法的计算效率是最优的.

图 2 给出了分别用基于标量乘的方法、基于同源的方法和最优策略计算ℓ^7-的同源.假设计算ℓ-同源的计算量为 q,计算标量乘$[\ell]R$(其中,R 为椭圆曲线上任意一个点)的计算量为 p,那么利用基于标量乘的方法需要的计算量为 21p+6q,利用基于同源的方法的计算量为 21q+6p,利用最优策略的方法的计算量为 11p+9q.显然,最优策略所需要的计算量最小,是最优的.

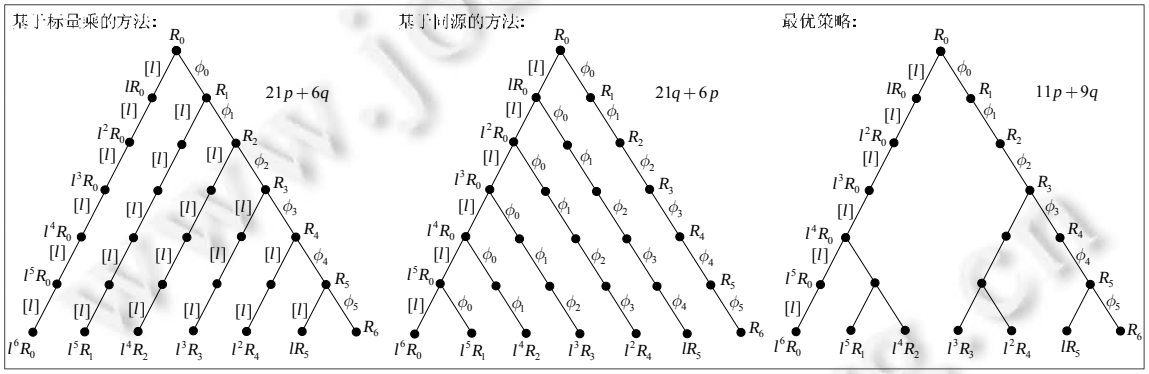

Fig.2 Compute ℓ^7-isogeny

图 2 计算ℓ^7-同源

Hutchinson 等人[43]利用并行处理的方法对基于最优策略算法进行了更进一步的优化.如图 2 所示,在计算同源ϕ_0、ϕ_1、ϕ_3时,可以利用并行处理的方法进行计算.

2.6 减少公钥的尺寸,优化压缩公钥的算法

对于基于超奇异椭圆曲线密钥交换协议的公钥尺寸的压缩,主要有两种方法:(1) 通过减少公钥的参数缩小公钥的规模;(2) 通过将公钥的点表示为基点的线性组合,将相应的坐标代替点达到压缩公钥的目的.此外,对于其压缩算法的效率也是目前研究的一个热点.

Costello 等人[24]利用 Montgomery 曲线下的坐标$(X_M:Z_M)$实现 SIDH,用 3 个点的横坐标:

$$x(\phi_A(P_B)),x(\phi_A(Q_B)),x(\phi_A(P_B-Q_B))$$

代替原来的公钥:

$$\phi_A(P_B),\phi_A(P_B),E_B,$$

其中,曲线系数的计算可以利用第 2.4 节中的方法(3),从而将公钥尺寸由原来的 $12\log_2 p$ 降低到 $6\log_2 p$.

Azarderakhsh 等人[44]通过将公钥中的点表示为

$$\phi_A(P_B)=a_0U_B+b_0V_B,\phi_A(Q_B)=a_1U_B+b_1V_B,$$

其中,$\left(a_i,b_i\in Z/\ell_B^{e_B}Z,i=0,1\right)$, U_B 和 V_B 为同源曲线 E_B 上阶为 $\ell_B^{e_B}$ 的线性独立点,计算双线性对:

$$g_0=e_{\ell_B^{e_B}}(U_B,V_B),$$

$$g_1=e_{\ell_B^{e_B}}(U_B,\phi_A(P_B))=e_{3^n}(U_B,a_0U_B+b_0Q_B)=g_0^{b_0},$$

$$g_2=e_{\ell_B^{e_B}}(U_B,\phi_A(Q_B))=e_{3^n}(U_B,a_1U_B+b_1Q_B)=g_0^{b_1},$$

$$g_3=e_{\ell_B^{e_B}}(V_B,\phi_A(P_B))=e_{3^n}(V_B,a_0U_B+b_0Q_B)=g_0^{a_0},$$

$$g_4=e_{\ell_B^{e_B}}(V_B,\phi_A(Q_B))=e_{3^n}(V_B,a_1U_B+b_1Q_B)=g_0^{a_1}.$$

求解离散对数得到 a_0、a_1、b_0、b_1,从而将公钥变为(E_B,a_0,a_1,b_0,b_1),将其尺寸减少到 $4\log_2 p$.然而,压缩公钥的算法由于需要双线性对和离散对数的计算,比以前的计算耗费量要大.Costello 等人[45]构造了 n 阶基点生成算法,优化 Tate 对计算以及 Pohlig-Hellman 算法,将公钥中表示点的线性组合的系数由原来的 4 个减少到 3 个,增加 1 额外比特信息,从而将尺寸减少到 $\frac{7}{2}\log_2 p$, 同时,将压缩公钥的计算效率提高了 2.4 倍.具体如下.

由于

$$\langle\phi_A(P_B)+\ell_B m\phi_A(Q_B)\rangle=\begin{cases}\langle a_0^{-1}\phi_A(P_B)+a_0^{-1}\ell_B m\phi_A(Q_B)\rangle, & \text{如果}a_0\in Z_{\ell_B^{e_B}}^*\\ \langle b_0^{-1}\phi_A(P_B)+b_0^{-1}\ell_B m\phi_A(Q_B)\rangle, & \text{如果}b_0\in Z_{\ell_B^{e_B}}^*\end{cases},$$

而

$$a_0^{-1}\phi_A(P_B)=U_B+a_0^{-1}b_0V_B,a_0^{-1}\phi_A(Q_B)=a_0^{-1}a_1U_B+a_0^{-1}b_1V_B,$$
$$b_0^{-1}\phi_A(P_B)=b_0^{-1}a_0U_B+V_B,b_0^{-1}\phi_A(Q_B)=b_0^{-1}a_1U_B+b_0^{-1}b_1V_B.$$

因此,公钥就变为

$$PK=\begin{cases}(E_B,0,a_0^{-1}b_0,a_0^{-1}a_1,a_0^{-1}b_1), & \text{如果}a_0\in Z_n^*\\ (E_B,1,b_0^{-1}a_0,b_0^{-1}a_1,b_0^{-1}b_1), & \text{如果}b_0\in Z_n^*\end{cases}.$$

Zanon 等人[46]提出了更有效的阶为 2^e 的基点生成算法,利用逆的基分解,即:由

$$\begin{bmatrix}\phi_A(P_B)\\ \phi_A(Q_B)\end{bmatrix}=\begin{bmatrix}a_0 & b_0\\ a_1 & b_1\end{bmatrix}\begin{bmatrix}U_B\\ V_B\end{bmatrix}$$

可推导

$$\begin{bmatrix}U_B\\ V_B\end{bmatrix}=\begin{bmatrix}c_0 & d_0\\ c_1 & d_1\end{bmatrix}\begin{bmatrix}\phi_A(P_B)\\ \phi_A(Q_B)\end{bmatrix},$$

其中,$\begin{bmatrix}a_0 & b_0\\ a_1 & b_1\end{bmatrix}=\frac{1}{D}\begin{bmatrix}d_1 & -d_0\\ -c_1 & c_0\end{bmatrix},D=c_0d_1-c_1d_0.$

计算双线性对:

$$h_0=e_{\ell_B^{e_B}}(\phi_A(P_B),\phi_A(Q_B))=e_{3^n}(P_B,Q_B)^{2^n},$$

$$h_1=e_{\ell_B^{e_B}}(\phi_A(P_B),U_B)=e_{3^n}(\phi_A(P_B),c_0\phi_A(P_B)+d_0\phi_A(Q_B))=h_0^{d_0},$$

$$h_2=e_{\ell_B^{e_B}}(\phi_A(P_B),V_B)=e_{3^n}(\phi_A(P_B),c_1\phi_A(P_B)+d_1\phi_A(Q_B))=h_0^{d_1},$$

$$h_3=e_{\ell_B^{e_B}}(\phi_A(Q_B),U_B)=e_{3^n}(\phi_A(Q_B),c_0\phi_A(P_B)+d_0\phi_A(Q_B))=h_0^{-c_0},$$

$$h_4=e_{\ell_B^{e_B}}(\phi_A(Q_B),V_B)=e_{3^n}(\phi_A(Q_B),c_1\phi_A(P_B)+d_1\phi_A(Q_B))=h_0^{-c_1}.$$

由于 P_B 和 Q_B 是公开参数,h_0 可以预计算,相比于 Costello 等人的算法减少了一个双线性对的计算.Naehrig

等人[47]利用 2-同源的对偶同源比 2-同源计算(核生成点非(0,0))要快的性质,将 SIKE 中密钥生成阶段的开销由原来的 140%~153%减少到 61%~74%,将密钥封装由原来的 67%~90%减少到 38%~57%,将解封装由原来的 59%~65%减少到 34%~38%.

3 CSIDH 实现的改进概述

根据前面对于 CSIDH 协议以及算法 1 的描述,优化 CSIDH 的实现关键在于提高算法 1 的效率,这一节主要总结对于算法 1 的各种优化方法.

3.1 设计常数时间的算法

对于算法 1 中的密钥 $[\boldsymbol{a}]=[\boldsymbol{l}_1^{e_1}\boldsymbol{l}_2^{e_2}...\boldsymbol{l}_n^{e_n}]$,指标 $e_1,\ldots,e_n$ 包含了需要计算的奇数次同源的个数.密钥不同,相应的指标不同,所需要计算的同源个数也会不同.另外,这些指标均从集合{−5,…,5}中选取,其正负情况决定了所选的基点在 $\mathbb{F}_p$ 中还是在 $\mathbb{F}_{p^2}$ 中,相应的同源的计算量也会有较大区别.因此,算法 1 的计算耗费量不是常数时间的,会随着密钥的变化而变化,不能抵抗侧信道攻击.针对上述存在的问题,Meyer 等人[48]利用差分加法的计算与同源的计算耗费量相同这一点,通过增加冗余的标量乘计算来固定同源的计算个数,并将相应的指标区间从集合{−5,…,5}变为集合{0,…,10},改进了算法 1,其计算的耗费量也是常数时间的,能够抵抗测信道的攻击.然而,其实现的效率却是算法 1 的 2 倍.Onuki 等人[49]设定密钥的空间为集合{−5,…,5},结合上述增加冗余固定同源个数的方式,提出了更快的实现 CSIDH 的算法,其运行时间依然为常数时间.然而,注意到增加冗余并不能抵抗错误注入攻击,Cervantes-Vázquez 等人[50]移除冗余的同源计算,利用 Edwards 曲线上的代数操作和加法链,提出更加有效的且能抵抗侧信道攻击的算法.

3.2 优化椭圆曲线中的标量乘计算

Meyer 等人[51]通过调整算法 1 中初始点的计算,计算 $P=4P_0,\alpha=\ell_1\ell_2\ldots\ell_n$,其中,$\ell_1>\ell_2>\ldots>\ell_n$.将 $K_0=\left[\frac{\alpha}{\ell_1}\right]P_0$ 作为第 1 个次数为 ℓ_1 同源的核生成点,并在具体的迭代计算中优先计算次数较大的同源,再计算次数较小的同源,减少标量乘的计算,从而使得算法 1 的实现效率比之前提高了 1.096 倍.之后,Meyer 等人将密钥空间的指标集合化分为多个集合,从而将同源的计算分解为多个分支,在很大程度上减少了每次循环需要的标量乘计算.

3.3 优化基点*P*的生成算法

当密钥的每个分量均变为正时,随机点的选取只需考虑定义在 $\mathbb{F}_p$ 上的点.Meyer 等人[51]首次利用 Eligator 算法快速生成定义在 $\mathbb{F}_p$ 上的点 P 的横坐标 x.给定曲线 $y^2=x^3+ax^2+x$,其中,$a\in\mathbb{F}_p$,该算法对于步骤(2)中的 $\frac{1}{u^2-1}$ 可以采取预先计算的方式,避免了求逆,比随机选取点的算法效率要高.

Eligator 算法[51].

输入:$u\in\mathbb{F}_p$;

输出:$x\in\mathbb{F}_p$.

1. 在集合 $\left\{2,3,...,\frac{p-1}{2}\right\}$ 中随机选一个 u,
2. 计算 $v=\frac{a}{u^2-1}$.
3. 计算 $e=\left(\frac{v^3+av^2+v}{p}\right)$.
4. 如果 e=1,输出 $x=v$;否则,输出 $x=-v-a$.

当密钥的每个分量有正有负时,Onuki 等人[49]利用 Eligator 算法快速生成两个合法点,分别是定义在$\mathbb{F}_p$上的点和定义在 $\mathbb{F}_{p^2}$ 上的点.

3.4 优化同源的计算

考虑到 CSIDH 中同源的次数相对于 SIDH 中同源的次数较大,Bernstein 等人[52]利用 Strassen 算法进行优化,将ℓ-同源的计算由原来的 O(ℓ)降为 O($\sqrt{\ell}$).

3.5 优化同源曲线的计算

注意到,CSIDH 中同源曲线的计算与在 SIDH 中的计算方式是不同的:在 SIDH 中,需要计算额外点的同源值,同源曲线的计算刚好可以利用这些点的同源值,避免了因同源次数增加而相应的同源曲线的计算量增加所带来的不便;在 CSIDH 中,不需要额外点的计算,同源曲线的计算只能利用推导的公式本身去优化计算.由于利用 Montgomery 曲线形式计算同源曲线的效率不是很高[24],Meyer 等人[48]借助 Montgomery 曲线与扭 Edwards 曲线之间双有理等价关系,利用扭 Edwards 曲线上的同源曲线公式来优化计算.即:在 Montgomery 曲线上,坐标$(X_M:Z_M)$可以通过变换:

$$(X_M:Z_M)\rightarrow(Y_E:Z_E)=(X_M+Z_M:X_M-Z_M)$$

转化到扭 Edwards 曲线射影坐标$(Y_E:Z_E)$.同时,Montgomery 曲线系数$(A:C)$可以通过变换 $a=A+2C$ 和 $d=A-2C$ 转化到扭 Edwards 曲线参数(a,d).在扭 Edwards 曲线上,利用公式:

$$a'=a^{\ell}(\prod_{i=1}^{S}Y_{[iP]})^8, d'=d^{\ell}(\prod_{i=1}^{S}Z_{[iP]})^8,$$

可以计算扭 Edwards 曲线上的ℓ-同源曲线参数(a',d'),其中,ℓ-同源的核为$\langle P\rangle=\{[iP]:i=0,\ldots,\ell=2s+1\}$.通过变换$(A':C')=(2(a'+d'):a'-d')$,可以得到 Montgomery 曲线上的ℓ-同源曲线.Kim 等人[38]借助奇数阶点在 2 倍标量乘的作用下不改变其阶这一性质,优化 Ewards 曲线上在新坐标$(W_E:Z_E)$下的同源曲线公式,如表 1 所示.

4 总结和展望

自椭圆曲线同源在公钥密码学中得到广泛应用,其对应的公钥加密和密钥封装进入美国 NIST 标准第 2 轮,对于 SIDH 和 CSIDH 的有效计算引起了学者们的重视,正如上面所分析的,取得了很多突破性的进展.但是,这一领域还有许多问题亟待解决.

(1) 借助不同曲线模型,探索不同坐标形式以及在该坐标下的倍点、点加、同源和同源曲线的计算公式,利用这些优化的公式来优化 SIDH 和 CSIDH 的实现.目前比较主流的实现 SIDH 和 CISDH 均在 Montgomery 和 Edwards 曲线上,对于其他曲线,如 Huff 曲线、Hessian 曲线、Legendre 曲线和 Jacobian Intersections 等的研究还比较少.是否最优的实现就是在 Montgomery 曲线或者 Edwards 曲线,亦或者有更好的曲线代替这两种曲线去实现 SIDH 和 CISDH,是值得关注的问题;

(2) 对于优化 SIDH,除了考虑以上的方法外,还可以利用超椭圆曲线的优势去优化计算.注意到利用亏格为 2 的 Kummer 面实现 SIDH,其域的尺寸在同等安全级别下比亏格为 1 的 Kummer 线上实现 SIDH 要小,其上的(2,2)-同源也有快速计算公式,然而,其(3,3)-同源的计算比较复杂,需要我们进一步加以深入研究;

(3) 对于同源次数形如 $\ell_1^{e_1}\ell_2^{e_2}\ldots\ell_k^{e_k}$ 的同源的有效计算,也是一个值得研究的问题.除了基于同源和基于标量乘的算法,设计最优策略算法或者借助并行处理的技巧优化计算,这也是我们迫切需要研究的;

(4) 对于 SIDH 中有限域 $\mathbb{F}_{p^2}$ 的特征 p 的选取,也是一个研究的热点.之前的 SIDH 中参数 $p=f\cdot\ell_A^{e_A}\cdot\ell_B^{e_B}-1$,要求 $\ell_A^{e_A}\approx\ell_B^{e_B}$. 当对于 SIDH 中一方要求计算比较快时,参数 p 的选取也可以有其他的方法,如 $p=2^nf-1$ 形式或者 $p=2^n-2^m-1$ 或者满足 $p+1$ 和 $p-1$ 均含有多个小素因子的乘积,且能够达到后量子相应的安全级别的素数 p;

(5) 对于公钥尺寸的有效压缩,也是目前研究的一个热点.基于 Kummer 线上的 SIDH 的加密和密钥封装的公钥压缩,已有很多工作.然而,对于阶为 3^e 点的有效压缩算法,依然是一个亟待解决的问题.此外,利用 Kummer 面上 SIDH 设计的加密方案,其上的公钥尺寸还比较大.能否缩短公钥的尺寸,如将公钥中的点表示为固定基点的线性组合?这些问题均是值得我们后续研究的;

(6) 对于 CSIDH 的优化,设计一个常数时间且有效的算法实现$[\boldsymbol{a}]\boldsymbol{E}$ 的计算,依然是目前研究的一个热点.目前设计的算法,实现的效率还比较低.能否借助一些特性,如基域上的ℓ-同源只有两种情况,减少同源的计算,或者构造更加有效的基点生成算法等,对其进行更进一步的优化?都是值得研究的;

(7) 借助其他优化标量乘或者双线性对[53]的技术,如借助多维标量乘[54]、椭圆网[55]等技术优化 SIDH 或者 CSIDH 的方式,也值得更进一步地加以研究.

References:

[1] Tate J. Endomorphisms of abelian varieties over finite fields. Inventiones Mathematicae, 1966,2:134−144. [doi: 10.1007/BF01404549]

[2] Rostovtsev A, Stolbunov A. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive: Report, 2006/145.

[3] Hu J. Research on cryptosystem based on elliptic curve isogeny [Ph.D. Thesis]. Wuhan: Wuhan University, 2010 (in Chinese with English abstract).

[4] Han WW, He DB. Provably secure key agreement protocol one elliptic curve isogenies. Computer Engineering, 2011,37(1): 128−130 (in Chinese with English abstract). [doi: 10.3724/SP.J.1146.2010.00230]

[5] Childs A, Jao D, Soukharev V. Constructing elliptic curve isogenies in quantum subexponential time. Journal of Mathematical Cryptology, 2014,8:1−29. [doi: 10.1515/jmc-2012-0016]

[6] Biasse JF, Jao D, Sankar A. A quantum algorithm for computing isogenies between supersingular elliptic curves. In: Meier W, Mukhopadhyay D, eds. Proc. of the INDOCRYPT 2014. LNCS 8885, Springer-Verlag, 2014. 428−442. [doi: 10.1007/978-3-319-13039-2_25]

[7] Costello C, Longa P, Naehrig M. Efficient algorithms for supersingular isogeny Diffie-Hellman. In: Robshaw M, Katz J, eds. Proc. of the ASIACRYPT 2016. LNCS 9814, Springer-Verlag, 2016. 572−601. [doi: 10.1007/978-3-662-53018-4_21]

[8] Jao D, De Feo L. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang BY, ed. Proc. of the PQCrypto 2011. LNCS 7071, Springer-Verlag, 2011. 19−34. [doi: 10.1007/978-3-642-25405-5_2]

[9] Jao D, Azarderakhsh R, Campagna M, Costello C, *et al*. SIKE. In: NIST Post-quantum Cryptography. 2018. https:/ /sike.org/

[10] Melchor CA, Aragon N, Bettaieb S, *et al*. HQC. In: NIST Post-quantum Cryptography. 2018. http://pqc-hqc.org/

[11] Bernstein DJ, Chou T, Lange T, *et al*. Classic McEliee. In: NIST Post-quantum Cryptography. 2018. https://classic.mceliece.org/

[12] Chen C, Danba O, Hoffstein O, *et al*. NTRU. In: NIST Post-Quantum Cryptography. 2018. https://ntru.org/

[13] Bernstein DJ, Chuengsatiansoup C, Lange T, *et al*. NTRU Prime. In: NIST Post-quantum Cryptography. 2018. https://ntruprime.cr.yp.to/

[14] Yoo Y, AzarderakhshR, Jalali A, Jao D, Soukharev V. A post-quantum digital signature scheme based on supersingular isogenies. In: Kiayias A, ed. Proc. of the FC 2017. LNCS 10322, Springer-Verlag, 2017. 163−181. [doi: 10.1007/978-3-319-70972-7_9]

[15] Galbraith SD, Petit C, Silva J. Identification protocols and signature schemes based on supersingular isogeny problems. In: Takagi T, Peyrin T, eds. Proc. of the ASIACRYPT 2017. LNCS 10624, Springer-Verlag, 2017. 3−33. [doi: 10.1007/978-3-319-70694-8_1]

[16] Unruh D. Post-quantum security of Fiat-Shamir. In: Takagi T, Peyrin T, eds. Proc. of the ASIACRYPT 2017. LNCS 10624, Springer-Verlag, 2017. 65−95. [doi: 10.1007/978-3-319-70694-8_3]

[17] Unruh D. Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald E, Fischlin M, eds. Proc. of the EUROCRYPT 2015. LNCS 9057, Springer-Verlag, 2015. 755−784. [doi: 10.1007/978-3-662-46803-6_25]

[18] Prest T, Fouque PA, Hoffstein J, Kirchner P, *et al*. FALCON. In: NIST Post-quantum Cryptography. 2018. https://falcon-sign.info/

[19] Bindel N, Akleylek S, Alkim E, *et al*. qTESLA. In: NIST Post-quantum Cryptography. 2018. https://qtesla.org/

[20] Aumasson JP, Endignoux G. Gravity-SPHINCS. In: NIST Post-quantum Cryptography. 2017. https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-Submissions

[21] Bernstein JD, Dobraunig C, Eichlseder M, Fluhrer S, *et al*. SPHINCS+. In: NIST Post-Quantum Cryptography. 2018. https://sphincs.org/

[22] Castryck W, Langes T, Martindale C, Panny L, Renes J. CSIDH: An effficient post-quantum commutative group action. In: Peyrin T, Galbraith S, eds. Proc. of the ASIACRYPT 2018. LNCS 11274, Springer-Verlag, 2018. 395−427. [doi: 10.1007/978-3-030-03332-3_15]

[23] Vélu J. Isogénies entre courbeselliptiques. Academie des Sciences de Paris, 1971,273:238−241.

[24] Costello C, Hisil H. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: Takagi T, Peyrin T, eds. Proc. of the ASIACRYPT 2017. LNCS 10625, Springer-Verlag, 2017. 303−329. [doi: 10.1007/978-3-319-70697-9_11]

[25] Koziel B, Azarderakhsh R, Jao D, Mozaffari-Kermani M. On fast calculation of addition chains for isogeny-based cryptography. In: Chen K, Lin D, Yung M, eds. Proc. of the Inscrypt 2016. LNCS 10143, Springer-Verlag, 2016. 323−342. [doi: 10.1007/978-3-319-54705-3_20]

[26] Joppe B, Simon F. Arithmetic considerations for isogeny based cryptography. IEEE Trans. on Computers, 2018, 1. [doi: 10.1109/TC.2018.2851238]

[27] Seo H, Jalali A, Azarderakhsh R. Optimized SIKE round 2 on 64-bit ARM. In: You I, ed. Proc. of the WISA 2019. LNCS 11897, Springer-Verlag, 2019. 341−353. [doi: 10.1007/978-3-030-39303-8_26]

[28] Costello C. B-SIDH: Supersingular isogeny Diffie-Hellman using torsion. In: Moriai S, Wang H, eds. Proc. of the ASIACRYPT 2020. LNCS 12492, Springer-Verlag, 2020. 440−463. [doi: 10.1007/978-3-030-64834-3_15]

[29] Flynn E, Ti YB. Genus two isogeny cryptography. In: Ding J, Steinwandt R, eds. Proc. of the PQCrypto 2019. LNCS 11505, Springer-Verlag, 2019. 286−306. [doi: 10.1007/978-3-030-25510-7_16]

[30] Costello C. Computing supersingular isogenies on kummer surfaces. In: Peyrin T, Galbraith S, eds. Proc. of the ASIACRYPT 2018. LNCS 11274, Springer-Verlag, 2018. 428−456. [doi: 10.1007/978-3-030-03332-3_16]

[31] Faz-Hernández A, López J, Ochoa-Jiménez E, Rodríguez-Henríquez F. A faster software implementation of the supersingular isogeny Diffie-Hellman key exchange protocol. IEEE Trans. on Computers, 2017,1. [doi: 10.1109/TC.2017.2771535]

[32] Montgomery PL. Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation, 1987,48(177): 243−264. [doi: 10.2307/2007888]

[33] De Feo L, Jao D, Plût J. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology, 2014,8:209−247. [doi: 10.1515/jmc-2012-0015]

[34] Renes J. Computing isogenies between Montgomery curves using the action of (0,0). In: Lange T, Steinwandt R, eds. Proc. of the PQCrypto 2018. LNCS 10786, Springer-Verlag, 2018. 229−247. [doi: 10.1007/978-3-319-79063-3_11]

[35] Berstein DJ, Birkner P, Joye M, Lange T, Peters C. Twisted Edwards curves. In: Vaudenay S, ed. Proc. of the AFRICACRYPT 2008. LNCS 5023, Springer-Verlag, 2008. 389−405. [doi: 10.1007/978-3-540-68164-9_26]

[36] Kim S, Yoon K, Kwon J, Hong S, Park YH. Efficient isogeny computations on twisted Edwards curves. Security and Communication Networks, 2018. [doi: 10.1155/2018/5747642]

[37] Farashahi RR, Hosseini SG. Differential addition on twisted Edwards curves. In: Pieprzyk J, Suriadi S, eds. Proc. of the ACISP 2017. LNCS 10343, Springer-Verlag, 2017. 366−378. [doi: 10.1007/978-3-319-59870-3_21]

[38] Kim S, Yoon K, Park YH, Hong S. Optimized method for computing odd-degree isogenies on Edwards curves. In: Galbraith S, Moriai S, eds. Proc. of the ASIACRYPT 2019. LNCS 11922, Springer-Verlag, 2019. 273−292. [doi: 10.1007/978-3-030-34621-8_10]

[39] Moody D, Shumow D. Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves. Mathematics of Computation, 2015,85:1929−1951. [doi: 10.1090/mcom/3036]

[40] Dang T, Moody D. Twisted hessian isogenies. Cryptology ePrint Archive: Report, 2019/1003.

[41] Xu X, Yu W, Wang K, He X. Constructing isogenies on extended Jacobi quartic curves. In: Chen K, Lin D, Yung M, eds. Proc. of the Inscrypt 2016. LNCS 10143, Springer-Verlag, 2016. 416−427. [doi: 10.1007/978-3-319-54705-3_26]

[42] Washington LC. Elliptic Curves: Number Theory and Cryptography. Boca Raton: CRC Press, 2008.

[43] Hutchinson A, Karabina K. Constructing canonical strategies for parallel implementation of isogeny based cryptography. In: Chakraborty D, Iwata T, eds. Proc. of the INDOCRYPT 2018. LNCS 11356, Springer-Verlag, 2018. 169−189. [doi: 10.1007/978-3-030-05378-9_10]

[44] Azarderakhsh R, Jao D, Kalach K, Koziel B, Leonardi C. Key compression for isogeny-based cryptosystems. In: Proc. of the AsiaPKC 2016. ACM, 2016. 1−10. [doi: 10.1145/2898420.2898421]

[45] Costello C, Jao D, Longa P, Naehrig M, Renes J, Urbanik D. Efficient compression of SIDH public keys. In: Coron JS, Nielsen J, eds. Proc. of the EUROCRYPT 2017. LNCS 10210, Springer-Verlag, 2017. 679−706. [doi: 10.1007/978-3-319-56620-7_24]

[46] Zanon G, Simplicio Marcos A, Pereira GCCF, Doliskani J, Barreto PSLM. Faster key compression for isogeny-based cryptosystems. IEEE Trans. Computers, 2019,68(5):688−701.

[47] Naehrig M, Renes J. Dual isogenies and their application to public-key compression for isogeny-based cryptography. In: Galbraith S, Moriai S, eds. Proc. of the ASIACRYPT 2019. LNCS 11922, Springer-Verlag, 2019. 243−272. [doi: 10.1007/978-3-030-34621-8_9]

[48] Meyer M, Reith S. A faster way to the CSIDH. In: Chakraborty D, Iwata T, eds. Proc. of the INDOCRYPT 2018. LNCS 11356, Springer-Verlag, 2018. 137−152. [doi: 10.1007/978-3-030-05378-9_8]

[49] Onuki H, Aikawa Y, Yamazaki T, Takagi T. A faster constant-time algorithm of CSIDH keeping two points. In: Attrapadung N, Yagi T, eds. Proc. of the IWSEC 2019. LNCS 11689, Springer-Verlag, 2019. 23−33. [doi: 10.1007/978-3-030-26834-3_2]

[50] Cervantes-Vázquez D, Chenu M, Chi-Domínguez JJ, De Feo L, Rodríguez-Henríquez F, Smith B. Stronger and faster side-channel protections for CSIDH. In: Schwabe P, Thériault N, eds. Proc. of the LATINCRYPT 2019. LNCS 11774, Springer-Verlag, 2019. 173−193. [doi: 10.1007/978-3-030-30530-7_9]

[51] Meyer M, Campos F, Reith S. On Lions and Elligators: An efficient constant-time implementation of CSIDH. In: Ding J, Steinwandt R, eds. Proc. of the PQCrypto 2019. LNCS 11505, Springer-Verlag, 2019. 307−325. [doi: 10.1007/978-3-030-25510-7_17]

[52] Bernstein DJ, De Feo L, Leroux A, Smith B. Faster computation of isogenies of large prime degree. Cryptology ePrint Archive: Report, 2020/341.

[53] Zhao CA, Zhang FG. Research and development on efficient pairing computations. Ruan Jian Xue Bao/Journal of Software, 2009, 20(11):3001−3009 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3651.htm [doi: 10.3724/SP.J.1001.2009.03651]

[54] Hisil H, Hutchinson A, Karabina K. d-MUL: Optimizing and implementing a multidimensional scalar multiplication algorithm over elliptic curves. In: Chattopadhyay A, Rebeiro C, eds. Proc. of the SPACE 2018. LNCS 11348, Springer-Verlag, 2018. 198−217. [doi: 10.1007/978-3-030-05072-6_12]

[55] Stange KE. The tate pairing via elliptic nets. In: Takagi T, Okamoto T, eds. Proc. of the Pairing 2007. LNCS 4575, Springer-Verlag, 2007. 329−348. [doi: 10.1007/978-3-540-73489-5_19]

附中文参考文献:

[3] 胡静.基于椭圆曲线同源的密码系统的研究[博士学位论文].武汉:武汉大学,2010.

[4] 韩维维,何德彪.可证安全的椭圆曲线同源密钥交换协商协议.计算机工程,2011,37(1):128−130. [doi: 10.3724/SP.J.1146.2010.00230]

[53] 赵昌安,张方国.双线性对有效计算研究进展.软件学报,2009,20(11):3001−3009. http://www.jos.org.cn/1000-9825/3651.htm [doi: 10.3724/SP.J.1001.2009.03651]

黄艳(1988－),女,博士,主要研究领域为基于椭圆曲线同源的密码体制的研究.

张方国(1972－),男,博士,教授,博士生导师,主要研究领域为椭圆曲线和超椭圆曲线密码体制,安全多方计算,隐私性,匿名性.