Ristretto: 萃取素数阶点群

longcpp

longcpp9@gmail.com

March 24, 2020

大量的密码协议的构造都依赖**素数阶群** $\mathbb{G}$, 工程实现时又多采用基于椭圆曲线的素数阶点群 $\mathbb{G}$: 例如基于 secp256k1, secp256r1 等曲线的素数阶点群. 密码机制的实现为了防止侧信道攻击等通常要求涉及到敏感信息的运算为常量时间 (Constant-Time), 然而 secp256k1/secp256r1 上的素数阶点群由于加法运算规则的不完备性, 在做群运算 (椭圆曲线点群加法) 时, 需要对输入的点做条件判断并用不同的执行分支处理不同的情形, 因此很难做到常量时间实现, 由此带来了安全隐患. 虽然可以构造具有完备加法运算 (Complete Addition Law) 素数阶点群, 但扭曲爱德华曲线 (Twisted Edwards Curves), 雅各比四次曲线 (Jacobi Quartic Curves) 等新的曲线类型上的点群有完备的加法运算规则并且通常更快更简单, 但代价是这些曲线上的点群的阶不是素数. 用 n 表示这些曲线上点群的阶, 则 n 可以表示为 $n = p \cdot h$, 其中 p 为大素数, h 称为余因子 (Cofactor). secp256k1/secp256r1 曲线上点群的 $h = 1$, 而扭曲爱德华曲线中 h 通常为 4 的倍数.

基于余因子不为 1 的点群构建密码协议通常需要在协议中添加额外检查以保证协议的安全性, 也因此为密码协议的构造带来额外的复杂度. 参见蒙哥马利曲线 Curve25519 上的 Diffie-Hellman 协议 X25519 以及基于扭曲爱德华曲线 Edwards25519(与 Curve25519 双向有理等价) 的 Ed25519 签名机制. (本文后续不再严格区分扭曲爱德华曲线和爱德华曲线, 爱德华曲线可以看成是扭曲爱德华曲线的特殊情况) 然而爱德华曲线等所带来的速度和易于安全实现的优点受到许多密码协议设计者青睐: TLS1.3 中采纳了基于爱德华曲线和蒙哥马利曲线的签名和 Diffie-Hellman 密钥交换协议, 而 CryptoNote 协议 (Monero, ByteCoin 的隐私交易协议) 也基于 Edwards25519 曲线构造.

在上层密码协议设计时考虑底层群结构的特殊性, 增加了协议设计的复杂性, 也增加了设计正确协议的难度, 这一点可参见 CryptoNote 协议中由于余因子不为 1 导致的双花漏洞以及该漏洞在 ByteCoin 中的真实利用[1]. 为了方便叙述, 记曲线点群的阶为 $n = h \cdot \ell$,

[1] longcpp. Edwards25519 余因子与双花交易. 20200212. https://github.com/longcpp/CryptoInAction/blob/master/intro-ed25519/200212-edwards25519-cofactor.pdf

其中 h 为余因子, ℓ 为大素数. 用 G 表示大素数子群的基点: $\mathsf{order}(G) = \ell$, 用 H 表示小子群的基点: $\mathsf{order}(H) = h$, 则椭圆曲线中的任一点 P 可以表示为: $P = x \cdot G + y \cdot H$, $x \in \mathbb{Z}_\ell$, $y \in \mathbb{Z}_h$. 假设 $P_1 = xG + y_1 H$, $P_2 = xG + y_2 H, x \in \mathbb{Z}_\ell$, y_1, $y_2 \in \mathbb{Z}_h$, 则当 $k \in \mathbb{Z}_n$, $k|h$ 时, 有 $kP_1 = kP_2$, 由于 Edwards25519 的点群的余因子为 8, 也就使得 CryptoNote 中的八花成为可能.

由此可见, 这种通过在上层协议设计中不断为底层点群数学结构的添加防范措施的方式, 很难构建正确的密码协议, 而且也会使得密码协议的安全论证更加复杂. Mike Hamberg 提出的 Decaf 方法[2] 能够在特定条件下从余因子为 4 的非素数阶的点群上" 萃取" 出素数阶的点群. 基于新的素数阶的点群, 余因子不为 1 相关的安全隐患与技术障碍都得以规避. 通过在爱德华曲线的点群上采用 Decaf 技术, 既可以利用爱德华曲线的速度等优势, 又能够避开点群余因子不为 1 的问题. Decaf 技术无需引入新的安全假设, 对既有的椭圆曲线点运算的实现的改动也很少. 然而 Decaf 技术无法直接应用于余因子为 8 的 Edwards25519 点群. Isis Agora Lovecruft 和 Henry de Valence 提出 Ristretto 技术通过扩展 Decaf 技术可以从余因子为 8 的非素数阶点群萃取出素数阶点群, 得到的素数阶点群记为 ristretto255. 由此上层协议可以安心利用曲线的各项优势而不再被余因子不为 1 的事实所阻碍.

余因子不为 1 所带来的主要问题是, 不同的点有可能在计算中表现出相同的行为, e.g. $P = x \cdot G + y \cdot H$ 中的 H 部分可能被消除. 消除问题的一种思路就是检查所用的点均属于大素数子群. 然而这种检查比较耗时, 需要一个点的标量乘法运算: 检查 $k \cdot P$ 是否为无穷远点 $\mathcal{O}$. 另一种思路调整点相等的判断逻辑, 对于点 P_1, P_2, 如果 $P_1 - P_2 = yH, y \in \mathbb{Z}_h$, 则认为 P_1, P_2 相等. 也即对于固定的 $x \in \mathbb{Z}_\ell$, 将集合 $\{P : P = xG + yH, y = 0, 1, \ldots, (h-1)\}$ 整个视为一个元素. 在这种视角下, 差值为低阶点的两个点变成了同一个点, 则阶为 $n = h \cdot \ell$ 的点群也就变成了素数 ℓ 阶的点群. 这就要求在椭圆曲线点群的实现中, 对应的调整 3 个计算过程: 点的编解码 $\mathsf{Encode}(\cdot)$ 和 $\mathsf{Decode}(\cdot)$ 以及点的相等判断 $\mathsf{Equal}(\cdot, \cdot)$. 其中 $\mathsf{Equal}(P_1, P_2) = \mathsf{True} \iff \exists y \in \mathbb{Z}_h$, s.t. $P_1 - P_2 = yH$. 当 $\mathsf{Equal}(P_1, P_2) = \mathsf{True}$ 时, 有 $\mathsf{Encode}(P_1) = \mathsf{Encode}(P_2)$, 并且 $\mathsf{Decode}(\cdot)$ 只接受 Encode 的输出为合法的输入参数.

基于这种思路, 一个自然的解决方案是在商群的视角下实现 $\mathsf{Equal}(P_1, P_2)$ 的逻辑. 并通过对陪集 (Coset) 添加逻辑约束的方式选定其中一个元素并进行编码来代表整个的陪集. 这也是 Hamberg 在 Decaf 的论文中提出的第一种方法, 这种方式的在于有些情况下不容易构造 $\mathsf{Decode}(\cdot)$ 函数, 并且无法适配蒙哥马利阶梯算法. Mike Hamberg 进一步提出了基于雅各比四次曲线 (Jacobi Quartic Curve) 的解决方案, 可以避开前述解决方案的

[2] Mike Hamburg. Decaf: Eliminating cofactors through point compression
https://www.shiftleft.org/papers/decaf/decaf.pdf

问题, 也即有清晰的编解码计算过程, 并且快速转换可参与爱德华曲线点群和蒙哥马利曲线点群运算. Decaf 可以处理余因子为 4 的点群, 而 Ristretto 技术则可以处理余因子为 8 的点群 (例如 Edwards25519 曲线上的点群). 两种技术都涉及到了 3 中椭圆曲线类型.

1 曲线类型与同源关系

每一个有 2 阶点的 Weierstrass 曲线都双向有理等价于雅各比四次曲线 (Jacobi Quartic Curve), 该曲线是由两个参数 e, A 决定的如下形式的曲线:

$$\mathcal{J}_{e,A} : t^2 = es^4 + 2As^2 + 1,$$

其中单位元为 $(0, 1)$, 当 $e(A^2 - e) \neq 0$ 时是非奇异 (Non-Singular) 的. 当 e 为有限域上的二次剩余,(仅考虑 e 为有限域上的二次剩余的情形) 也即存在 δ 满足 $e = \delta^2$ 时, $\mathcal{J}_{e,A}$ 有完整的 2-torsion: $\mathcal{J}[2] \cong \mathbb{Z}_2 \times \mathbb{Z}_2$.[3] 点 $P = (s, t)$ 的 $\mathcal{J}[2]$ 陪集为:

$$P + \mathcal{J}[2] = \{(s, t), (-s, -t), (1/\delta s, -t/\delta s^2), (-1/\delta s, t/\delta s^2)\}.$$

注意到由于 $(-\delta)^2 = e$ 并且当选取 $-\delta$ 时, 得到的仍然是同样的陪集.

扭曲爱德华曲线 (Twisted Edwards Curve) 是由参数 a, d 定义的如下形式的曲线:

$$\mathcal{E}_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

当 $a = 1$ 即为爱德华曲线, 而当 $a = -1$ 时曲线上的点群运算最快. 当 a 为二次剩余而 d 为非二次剩余时, 图线上的点群具有完备的加法规则并且具有循环 4-torsion 子群:

$$\mathcal{E}_{a,d}[4] = \{(0, 1), (0, -1), (1/\sqrt{a}, 0), (-1/\sqrt{a}, 0)\}$$

[3] $\mathcal{J}[2] \cong \mathbb{Z}_2 \times \mathbb{Z}_2$ 意味着存在 3 个 2 阶点. $\mathcal{J}_{e,A}$ 的特殊之处在于, 仿射坐标系下 $\mathcal{J}_{e,A}$ 仅有一个 2 阶点 $(0, -1)$, 另外 2 个 2 阶点存在于坐标系下. $\mathcal{J}_{e,A}$ 的投影闭包的方程为:

$$\overline{\mathcal{J}}_{e,A} : Y^2Z^2 = eX^4 + 2AX^2Z^2 + Z^4,$$

曲线上的点 $(X : Y : Z), Z \neq 0$ 对应 $\mathcal{J}_{e,A}$ 上仿射坐标 $(X/Z, Y/Z)$. 然而不幸运的是, $\overline{\mathcal{J}}_{e,A}$ 上的点 $(0 : 1 : 0)$ 为奇点. 为了消除该奇点会额外引入两个点记为 $\mathcal{O}_1, \mathcal{O}_2$. 消除该奇点的一种方式是考虑带权重的投影方程:

$$\hat{\mathcal{J}}_{e,A} : Y^2 = eX^4 + 2AX^2Z^2 + Z^4,$$

曲线上的点 $(X : Y : Z), Z \neq 0$ 对应 $\mathcal{J}_{e,A}$ 上仿射坐标 $(X/Z, Y/Z^2)$. 而点 $(1 : \delta : 0), (1 : -\delta : 0) \in \hat{\mathcal{J}}_{e,A}$ 就对应 $\mathcal{O}_1, \mathcal{O}_2$, 并且 $\hat{\mathcal{J}}_{e,A}$ 上不存在 $Z = 0$ 的其它点. 可以将 $\mathcal{O}_1 = (1 : \delta : 0), \mathcal{O} = (1 : -\delta : 0)$ 看做是两个无穷远点. 比较特殊的是, 这两个无穷远点都是 2 阶点. 点 $P = (s, t)$ 的 $\mathcal{J}[2]$ 陪集的表示形式, 没有能够完整的推导推出来. 推测一个可行的路径是, 在带权重的投影方程下根据点加公式计算完成点 P 与 $(0, 1), (0, -1), \mathcal{O}_1, \mathcal{O}_2$ 的和, 然后将结果映射到仿射坐标系下. 关于雅各比四次曲线的更多细节可以参考 Hüseyin Hişil 2010 年的博士论文 "Elliptic curves, group law, and efficient computation." https://eprints.qut.edu.au/33233

注意到 Edwards25519 中 $a=-1$ 为二次剩余并且, d 为非二次剩余时, 则 Edwards25519 $(0,1)$ 为单位元, $(0,-1)$ 为 2 阶点, 而 $(1/\sqrt{-1},0),(-1/\sqrt{-1},0)$ 为 4 阶点. 爱德华曲线上的加法公式, 两个点 $(x_1,y_1),(x_2,y_2)$ 的和记为 (x_3,y_3), 则有

$$(x_3,y_3)=(x_1,y_1)+(x_2,y_2)=\left(\frac{x_1y_2+y_1x_2}{1+dx_1x_2y_1y_2},\frac{y_1y_2-ax_1x_2}{1-dx_1x_2y_1y_2}\right)$$

根据该计算公式有, 点 $P=(x,y)\in\mathcal{E}_{a,d}$ 关于 $\mathcal{E}_{a,d}[4]$ 的陪集为:

$$P+\mathcal{E}_{a,d}[4]=\{(x,y),(-x,-y),(y/\sqrt{a},-\sqrt{a}x),(-y/\sqrt{a},\sqrt{a}x)\}.$$

根据 Decaf 的论文有, $\mathcal{J}_{a_1^2,a_1-2d_1}$ 与 $\mathcal{E}_1=\mathcal{E}_{a_1,d_1}$ 是 2-同源的 (2-Isogenous):

$$(x,y)\in\mathcal{E}_1=\mathcal{E}_{a_1,d_1}=\phi(s,t)=\left(\frac{2s}{1+a_1s^2},\frac{1-a_1s^2}{t}\right),$$

并且有

$$(s,t)\in\mathcal{J}_{a_1^2,a_1-2d_1}=\hat{\phi}(x,y)=\left(\frac{x}{y},\frac{2-y^2-a_1x^2}{y^2}\right)$$

意味对于点 $(s,t)\in\mathcal{J}_{a_1^2,a_1-2d_1}$, 有 $\hat{\phi}\circ\phi(s,t)=2(s,t)$. 论文 "Jacobi Quartic Curves Revisited"[4] 中指出, 雅各比四次曲线 $\mathcal{J}_{e,A}$ 的点加公式 $(s_3,t_3)=(s_1,t_1)+(s_2+t_2)$ 为

$$(s_3,t_3)=\left(\frac{s_1t_2+t_1s_2}{1-es_1^2s_2^2},\frac{(t_1t_2+2As_1s_2)(1+es_1^2s_2^2)+2es_1s_2(s_1^2+s_2^2)}{(1-es_1^2s_2^2)^2}\right)$$

由此得到可以得到点的倍乘公式:

$$2(s,t)=\left(\frac{2st}{1-es^4},\frac{(t^2+2As^2)(1+es^4)+4es^4}{(1-es^4)^2}\right)\tag{1}$$

根据上述公式以及 ϕ 和 $\hat{\phi}$, 我们接下来验证 $\mathcal{J}_{a_1^2,a_1-2d_1}$ 与 $\mathcal{E}_1=\mathcal{E}_{a_1,d_1}$ 之间的 2 同源. 取点 $(s,t)\in\mathcal{J}_{a_1^2,a_1-2d_1}$, 则

$$(x,y)=\phi(s,t)=\left(\frac{2s}{1+a_1s^2},\frac{1-a_1s^2}{t}\right)$$

假设点 $(s',t')\in\mathcal{J}_{a_1^2,a_1-2d_1}=\hat{\phi}(x,y)$, 则有

$$(s',t')=\left(\frac{x}{y},\frac{2-y^2-a_1x^2}{y^2}\right)=\left(\frac{\frac{2s}{1+a_1s^2}}{\frac{1-a_1s^2}{t}},\frac{2-\left(\frac{1-a_1s^2}{t}\right)^2-a_1\left(\frac{2s}{1+a_1s^2}\right)^2}{\left(\frac{1-a_1s^2}{t}\right)^2}\right)$$

则有

$$s'=\frac{2st}{(1+a_1s^2)(1-a_1s^2)}=\frac{2st}{1-a_1^2s^4}=\frac{2st}{1-es^4}.\tag{2}$$

[4]Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, Ed Dawson. Jacobi Quartic Curves Revisited https://eprint.iacr.org/2009/312.pdf

接下来考虑 t':

$$t' = \frac{2t^2(1+a_1s^2)^2 - (1-a_1s^2)^2(1+a_1s^2)^2 - a_1(2s)^2t^2}{(1-a_1s^2)^2(1+a_1s^2)^2},$$

$$t' = \frac{2t^2(1+a_1s^2)^2 - 2t^2\cdot 2a_1s^2 - (1-a_1s^2)^2(1+a_1s^2)^2}{(1-a_1^2s^4)^2},$$

$$t' = \frac{2t^2(1+a_1^2s^4) - (1-a_1^2s^4)^2}{(1-a_1^2s^4)^2} = \frac{2t^2(1+a_1^2s^4) - [(1+a_1^2s^4)^2 - 4a_1^2s^4]}{(1-a_1^2s^4)^2}$$

$$t' = \frac{(1+a_1^2s^4)\cdot\left[2t^2 - (1+a_1^2s^4)\right] + 4a_1^2s^4}{(1-a_1^2s^4)^2} \tag{3}$$

注意到 $\mathcal{J}_{a_1^2,a_1-2d_1}$ 中有 $e = a_1^2$ 以及 $t^2 = es^4 + 2As^2 + 1 \Rightarrow t^2 - (es^4+1) = 2As^2$, 代入上式就有

$$t' = \frac{(1+es^4)\cdot(t^2+2As^2) + 4es^4}{(1-es^4)^2}$$

对比公式 (2), 公式 (3) 以及公式 (1), 可以验证:

$$\hat{\phi}\circ\phi(s,t) = 2(s,t)$$

同理也可验证

$$\phi\circ\hat{\phi}(x,y) = 2(x,y)$$

蒙哥马利曲线 (Montgomery Curves) 是由参数 $A, B, B(A^2-4)\neq 0$ 定义的如下形式的曲线:

$$\mathcal{M}_{B,A} : Bv^2 = u(u^2+Au+1),$$

其中单位元为无穷远点. 值得注意的是, $\mathcal{J}_{a_1^2,a_1-2d_1}$ 与 $\mathcal{M}_{B,A}, B = a_1, A = (2-4d_1)/a_1$ 也是 2 同源的:

$$\psi(s,t) = \left(\frac{1}{a_1s^2}, \frac{-t}{a_1s^3}\right),\ \hat{\psi}(u,v) = \left(\frac{1-u^2}{2a_1v}, \frac{a_1(u+1)^4 + 8d_1u(u^2+1)}{4a_1^2v^2}\right)$$

而当 $(A+2)/a_2B$ 为二次剩余时, $\mathcal{M}_{B,A}$ 与爱德华曲线 $\mathcal{E}_2 = \mathcal{E}_{a_2,d_2}$ 同构, 其中

$$a_2 = \pm 1, d_2 = a_2\frac{A-2}{A+2}.$$

同构映射以及逆映射分别用 $\eta(u,v)$ 和 $\hat{\eta}(x,y)$ 表示, 具体定义参见 https://ristretto.group/details/isogeny_encoding.html. 其中 a_1, a_2, d_1, d_2 之间的关系为 $a_1 = -a_2, d_1 = (a_2d_2)/(a_2-d_2)$, 也即

$$\mathcal{J}_{a_1^2,a_1-2d_1} = \mathcal{J} = \mathcal{J}_{a_2^2,-a_2(a_2+d_2)/(a_2-d_2)}.$$

根据 $\psi, \hat{\psi}, \eta, \hat{\eta}$ 就有 $\mathcal{J}$ 与 $\mathcal{E}_2$ 为 2 同源的:

$$\theta_{a_2,d_2}(s,t) = \eta \circ \psi(s,t) = \left(\frac{1}{\sqrt{a_2 d_2 - 1}} \cdot \frac{2s}{t}, \frac{1 + a_2 s^2}{1 - a_2 s^2} \right),$$

其对偶映射为

$$\hat{\theta}_{a_2,d_2}(x,y) = \hat{\psi} \circ \hat{\eta}(x,y) = \left(\sqrt{a_2 d_2 - 1} \cdot \frac{xy}{1 - a_2 x^2}, \frac{y^2 + a_2 x^2}{1 - a_2 x^2} \right)$$

Decaf 使用了曲线 $\mathcal{E}_1 = \mathcal{E}_{a_1,d_1}$, 而 Ristretto 使用了曲线 $\mathcal{E}_2 = \mathcal{E}_{a_2,d_2}$. Ristretto 之所以选用与 Decaf 中不同的爱德华曲线表示, 是因为在 Edwards25519 曲线上实现 Ristretto255 点群时, 这种方式可以有更好的实现效率.

2 基于曲线同源的编解码

上一小节中, 引入了三种椭圆曲线, 并且不厌其烦的考察了三种曲线之间的同源关系. 这也是 Decaf 和 Ristretto 技术的核心. Decaf 的方法是利用商群 $\mathcal{J}/\mathcal{J}[2]$ 进行编码, 然后可以将编码方式快速映射成爱德华曲线或者蒙哥马利曲线以参与点群运算. Distretto 技术基于同样的技术路线, 但是在编码 $\mathcal{J}/\mathcal{J}[2]$ 时不同并且支持余因子为 8 的情况.

如前所述, 点 $P = (s,t) \in \mathcal{J}_{e,A}$ 的 $\mathcal{J}[2]$ 陪集为:

$$P + \mathcal{J}[2] = \{(s,t), (-s,-t), (1/\delta s, -t/\delta s^2), (-1/\delta s, t/\delta s^2)\}.$$

接下来只需要定义关于该陪集的一个规范表示 (Canonical Representative), 也即选择陪集中的一个点代表整个陪集, 然后编码该点即可. 考虑进点的压缩表示, 则可以用选中的点的 s 坐标表示该点, 并规定 t 坐标的正负即可. Decaf 技术中要求 s 坐标非负并且为有限值, 而 t/s 为非负值或者无限. Ristretto 则选用了不同的符号规定.

接下来考察上一节引入的各种曲线之间的同源关系在 Decaf 和 Ristretto 技术中的应用. 首先回顾一个代数结论: 对于如果 $\mathbb{G}'$ 是交换群 $\mathbb{G}$ 的子群, 并且存在群同态 $\rho : \mathbb{G} \to \mathbb{H}$, 则 ρ 也是从 $\mathbb{G}/\mathbb{G}'$ 到 $\rho(\mathbb{G})/\rho(\mathbb{G}')$ 的群同态, 并且有 $\rho(\mathbb{G})/\rho(\mathbb{G}') \subseteq \mathbb{H}/\rho(\mathbb{G}')$. 更进一步, 如果 $\mathsf{Kernel}(\rho) \subseteq \mathbb{G}'$, 则 ρ 为同构映射. 对于

$$\phi : \mathcal{J} \to \mathcal{E}_1, \theta : \mathcal{J} \to \mathcal{E}_2,$$

都有

$$\mathsf{Kernel}(\phi) \subseteq \mathcal{J}[2], \mathsf{Kernel}(\theta) \subseteq \mathcal{J}[2],$$

则有

$$\frac{\mathcal{J}}{\mathcal{J}[2]} \cong \frac{\phi(\mathcal{J})}{\phi(\mathcal{J}[2])} \cong \frac{[2]\mathcal{E}_1}{\mathcal{E}_1[2]}, \quad \frac{\mathcal{J}}{\mathcal{J}[2]} \cong \frac{\theta(\mathcal{J})}{\theta(\mathcal{J}[2])} \cong \frac{[2]\mathcal{E}_2}{\mathcal{E}_2[2]},$$

根据上述同构映射, $\mathcal{J}/\mathcal{J}[2]$ 的编码可以被转换到 $[2]\mathcal{E}_1/\mathcal{E}_1[2]$ 和 $[2]\mathcal{E}_2/\mathcal{E}_2[2]$. 当余因子为 4, 也即点群的阶为 4ℓ 是, $|[2]\mathcal{E}_1/\mathcal{E}_1[2]| = \ell$, 余因子被消除. 而余因子为 8 时并且 8-torison 为循环子群时, 无法通过相同方式直接消除掉余因子. 然而注意到此时有 $[2](\mathcal{E}[8]) = \mathcal{E}[4]$, 也即 $\mathcal{E}[4] \subseteq [2]\mathcal{E}$. 同时注意到 $|[2]\mathcal{E}/\mathcal{E}[4]| = \ell$ 并且根据前述同构映射可以将 $\mathcal{J}/\mathcal{J}[2]$ 的编码映射到 $[2]\mathcal{E}_2/\mathcal{E}_2[2]$ 上, 则对于余因子为 8 时的编码, 唯一剩下的需要处理的问题就是 $\mathcal{E}/\mathcal{E}[4]$ 与 $\mathcal{E}/\mathcal{E}[2]$ 之间的不同, 也即将点从 $\mathcal{E}/\mathcal{E}[4]$ 扭转到 $\mathcal{E}/\mathcal{E}[2]$.

根据前面对爱德华曲线的描述, $\mathcal{E}_{a,d}[4]$ 的 4-torsion 为:

$$\mathcal{E}_{a,d}[4] = \{(0,1), (0,-1), (1/\sqrt{a}, 0), (-1/\sqrt{a}, 0)\},$$

并且 $(0,1)$ 为单位元, $(0,-1)$ 为 2 阶点, $(1/\sqrt{a},0), (-1/\sqrt{a},0)$ 为 4 阶点. 点 $P = (x,y) \in \mathcal{E}_{a,d}$ 关于 4-torsion 的陪集为:

$$P + \mathcal{E}_{a,d}[4] = \{(x,y), (-x,-y), (y/\sqrt{a}, -\sqrt{a}x), (-y/\sqrt{a}, \sqrt{a}x)\},$$

其中

$$\begin{aligned}
(x,y) + (0,1) &= (x,y) \\
(x,y) + (0,-1) &= (-x,-y) \\
(x,y) + (1/\sqrt{a},0) &= (y/\sqrt{a}, -\sqrt{a}x) \\
(x,y) + (-1/\sqrt{a},0) &= (-y/\sqrt{a}, \sqrt{a}x)
\end{aligned}$$

当要求上述运算的结果的两个坐标 $xy > 0$ 时, 也即为点属于 $\{(x,y), (-x,-y)\}$ 时, 参与加法运算的点 $(0,1), (0,-1) \in \mathcal{E}_{a,d}[2]$. 当 $xy < 0$ 或者 $y = 0$ 时, 对应的点属于 $\{(y/\sqrt{a}, -\sqrt{a}x), (-y/\sqrt{a}, \sqrt{a}x)\}$ 时, 给对应的点再加上一个 4 阶点, $a = 1$ 时选用点 $(1,0)$, $a = -1$ 时选用点 $(i,0)$, 则 $\{(y/\sqrt{a}, -\sqrt{a}x), (-y/\sqrt{a}, \sqrt{a}x)\}$ 被转换成 $\{(x,y), (-x,-y)\}$:

$$(y/\sqrt{a}, -\sqrt{a}x) + (1,0) = (-x,-y), (-y/\sqrt{a}, \sqrt{a}x) + (1,0) = (x,y),$$

或者

$$(y/\sqrt{a}, -\sqrt{a}x) + (i,0) = (x,y), (-y/\sqrt{a}, \sqrt{a}x) + (i,0) = (-x,-y)$$

也即给定点 $P + \mathcal{E}_{a,d}[4]$, 当 $xy > 0$, 点已经在 $\mathcal{E}/\mathcal{E}[2]$ 中, 当 $xy < 0$ 或者 $y = 0$ 时, 给点加上一个 4 阶点就将点扭转 (Torquing) 到了 $\mathcal{E}/\mathcal{E}[2]$. 由此解决了应用 $\mathcal{J}/\mathcal{J}[2]$ 进行编码的最后一个障碍, 也因此从余因子为 8 的点群出发最终萃取出了素数阶点群 $\mathcal{J}/\mathcal{J}[2]$.

输入 $(x,y) \in [2]\mathcal{E}$, 具体的编码步骤为:

1. 如果 $xy < 0$ 或者 $y = 0$, 进行点扭转 $(x,y) \leftarrow (x,y) + Q_4$, Q_4 为 4 阶点.
2. 如果 $x < 0$ 或者 $y = -1$, $(x,y) \leftarrow (x,y) + (0,1) = (-x,-y)$

3. 计算 $s = \pm\sqrt{(-a)(1-y)/(1+y)}$, 选择正值

则最终有限域上的值 s 的编码记为点 $(x,y) \in [2]\mathcal{E}$ 的 Ristretto 编码. 值得注意的是, Ristretto 编码只处理 $[2]\mathcal{E}$ 中的点, 以 Edwards25519 曲线为例, 由于其 8-torsion 为循环群, 因此 $[2]\mathcal{E}[8] = \mathcal{E}[4]$, 因此任意的 $(x,y) \in [2]\mathcal{E}$, 都隶属于某个点的 $\mathcal{E}[4]$ 陪集. 编码中的第 1 步将点从 $\mathcal{E}/\mathcal{E}[4]$ 扭转到了 $\mathcal{E}/\mathcal{E}[2]$ 中. 对于第 2 步和第 3 步, 根据 $\theta(s,t)$ 有 $y = (1+as^2)/(1-as^2)$, 则 $\theta^{-1}(x,y)$ 的 s 坐标为: $s^2 = (-a)(1-y)/(1+y)$, 则 $x = 2s/(t\cdot\sqrt{ad-1}) \implies s/t = x\sqrt{ad-1}/2$, 也即 s/t 的正负号已经由 x 的符号决定, 也即第 2 步决定了 s/t 的符号. $(s,t) + \mathcal{J}[2]$ 进行规范编码时要求做 2 个符号选择: s 的符号以及 s/t 的符号. s/t 的符号已经在第 2 步中选定, 而第 3 步中则计算 s 并选定其符号. 而第 2 步中, 对 $y = -1$ 情形的处理是是为了避免在第 3 步中发生除以 0 的情况,

输入字节串, 进行 Ristretto 解码的过程如下:

1. 解码得到 s, 如果字节串不是 s 的规范编码, 解码失败
2. 如果 s 为负数, 解码失败
3. 计算 $y = (1+as^2)/(1-as^2)$
4. 计算 $x = +\sqrt{(4s^2)/\left(ad(1+as^2)^2 - (1-as^2)^2\right)}$, 根不存在则解码失败
5. 如果 $xy < 0$ 或者 $y = 0$ 解码失败

有了编解码之后, 还需要考虑在这种编码下如何判断点的相等, 此时需要在商群的视角下进行判断. 如果有两个点位于同样的陪集, 则两个点相等. 通常点群的实现依赖投影坐标系, 而为了判断点是否相等, 往往需要将投影坐标转到换射影坐标, 其中会涉及到计算量很大的求逆运算. 然而判断两个点是否位于同一个陪集, 可以至二级在投影坐标系下完成, 使得此时的点相等判断反而具有更高的效率. 具体实现时, 点 (x,y) 通常表示为扩展的扭爱德华坐标系 $(X:Y:Z:T)$, 其中 $x = X/Z, y = Y/Z, xy = T/Z$. 采用 Ristretto 编码时,$P_1$ 和 P_2 相等, 如果满足

$$P_1 \equiv P_2 \mod \mathcal{E}[2], \text{ or }, P_1 + Q_4 \equiv P_2 \mod \mathcal{E}[2].$$

Decaf 论文中已经证明, 判断 $P_1 \equiv P_2 \mod \mathcal{E}[2]$ 等价于判断 $X_1Y_2 = Y_1X_2$. 记 $P_1 + Q_4 = (X_1' : Y_1' : Z_1' : T_1')$, 在射影坐标系下 $P + Q_4 = (y/\sqrt{a}, -x\sqrt{a})$, 也即 $X_1' = Y_1/\sqrt{a}$, 而 $Y_1' = -X_1\sqrt{a}$, 则有

$$X_1'Y_2 = Y_1'X_2 \iff Y_1Y_2\sqrt{a} = -\sqrt{a}X_1X_2 \iff Y_1Y_2 = -aX_1X_2.$$

也即判断两个点在 $\mod \mathcal{E}[4]$ 的视角下是否相等, 等价于判断

$$X_1Y_2 = Y_1X_2 \text{ or } Y_1Y_2 = -aX_1X_2.$$