♣暗号入門3日の3日目♣

「代数曲線と暗号プリミティブ」

松尾和人

2007年8月10日13:00-16:10

♣ Diffie-Hellman鍵共有アルゴリズム (1976) ♣

システム設定	
p: **素数**, $b \in [1, p-1]$ s.t. $\{b^i \mid i \in [0, p-2]\} = \{1, \ldots, p-1\}$	
鍵ペア生成	
アントニオ	
秘密鍵設定	$K_a \in [1, p-1]$
公開鍵計算	$K'_a \equiv b^{K_a} \bmod p$
	公開鍵 K'_aを公開
ババ	
秘密鍵設定	$K_b \in [1, p-1]$
公開鍵計算	$K'_b \equiv b^{K_b} \bmod p$
	公開鍵 K'_bを公開
共通鍵計算	
アントニオ	$K \equiv K_b'^{K_a} \bmod p$
バ ババ	$K \equiv K_a'^{K_b} \bmod p$
同一の鍵Kを共有できた	

♣ 離散対数問題 ♣

- $K'_* \mapsto K_*$
- Given: p: prime, $b \in [1, p-1]$, $a \in \{b^i \mid i \in [0, p-2]\}$
  Find: $x \in [0, p-2]$ s.t. $a \equiv b^x \bmod p$
  $\mathrm{Ind}_b a := x$
- **簡単：** $(x, b, p) \mapsto a \equiv b^x \bmod p$
  - $x = (x_n x_{n-1} \ldots x_1 x_0)_2$,
    $a \equiv \prod_{0 \le i \le n} b^{2^{x_i}} \bmod p$,
    $n = O(\log p)$
- **困難：** $(a, b, p) \mapsto x$

♣ 離散対数問題の難しさ ♣

- **全数探索**
  - $O(p)$
- Square-root**法**
  - $O\left(\sqrt{l}\right)$
  - l**：**$p-1$**の最大素因子**
- **指数計算法** (Adleman, 1979)
  - $L_x(\alpha,\beta) :=$
    $\exp\left(\beta(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right)$
  - $O(L_p(1/2, 2+o(1)))$
  - $O(L_p(1/3, 1.903+o(1)))$

♣ Square-root法：PollardのRho法 (1978) ♣

- Monte Carlo (Las Vegas) Algo.
- **空間計算量**: $O(1)$
- **パラレル計算可能**
- **基本アイディア：バースデイパラドックスの利用**

クラスメイトが23**人いれば、**
同じ誕生日のペアが居る確率は$1/2$**以上**

$$1-1\times\frac{364}{365}\times\frac{363}{365}\times\cdots\times\frac{343}{365}=0.507\ldots$$

$$\sqrt{365}=19.104\ldots$$

♣ Birthday Paradox ♣

S : set, $n_0 = \#S$

r 個の中に1組も同じ値のペアがない確率:

$$\begin{aligned}
\prod_{i=1}^{r} \frac{n_0 - i + 1}{n_0} &= \prod_{i=1}^{r} \left(1 - \frac{i-1}{n_0}\right) \\
&< \prod_{i=1}^{r} \exp\left(-\frac{i-1}{n_0}\right) \\
&\quad \because 1 + x \leq e^x \\
&= \exp\left(\sum_{i=1}^{r} -\frac{i-1}{n_0}\right) \\
&= \exp\left(-\frac{r(r-1)}{2n_0}\right) \\
&\approx \exp\left(-\frac{r^2}{2n_0}\right)
\end{aligned}$$

$$r = \sqrt{2(\log 2)n_0} \Rightarrow \exp\left(-\frac{r^2}{2n_0}\right) = 0.5$$

$\Rightarrow$ **$O(\sqrt{n_0})$ 個の中には**
一致するペアがある確率が高い

♣ Pollardのρ法（原型）の実際 ♣

Given: $p = 47, a = 40, b = 11$

Find: $\mathrm{Ind}_b a$ i.e. x s.t. $a \equiv b^x \bmod p$

	1	2	3	4	5
α	35	36	17	9	3
β	3	41	15	0	28
$a^\alpha b^\beta \bmod p$	27	43	24	29	<u>30</u>

	6	7	8	9	10
	17	16	37	38	39
	14	7	17	25	8
	15	40	6	13	<u>30</u>

$a^3 b^{28} \equiv a^{39} b^8 \bmod p$

$\Rightarrow$

$a \equiv b^{(8-28)/(3-39)} \bmod p$

$\Rightarrow$

$x \equiv \frac{8-28}{3-39} \equiv \frac{20}{36} \equiv 21 \bmod p - 1$

♣ Pollardのρ法の原型 ♣

Algorithm 1 Pollard's rho.alpha

Input: p: 素数, $a, b \in [1, p-1]$

Output: $x \in [0, p-2]$ s.t. $a \equiv b^x \bmod p$

1: $i := 0$
2: **repeat**
3: $\quad i := i + 1$
4: $\quad$ Choose $\alpha_i, \beta_i \in [0, p-2]$ randomly
5: $\quad c_i \equiv a^{\alpha_i} b^{\beta_i} \bmod p - 1$
6: **until** $^{\exists} j$ s.t. $1 \leq j < i, c_j = c_i$
7: $x \equiv (\beta_j - \beta_i)(\alpha_i - \alpha_j)^{-1} \bmod p - 1$
$\quad$ /*$\alpha_i x + \beta_i \equiv \alpha_j x + \beta_j \bmod p - 1$*/
8: Output x and terminate

（平均）時間計算量:

$O(\sqrt{p}) \rightarrow O\left(\sqrt{l}\right)$, l：$p-1$**の最大素因子**

（平均）空間計算量:

$O(\sqrt{p}) \rightarrow O(1)$

♣ 指数計算法の実際 ♣

Given: $p = 47, a = 40, b = 11$

Find: $\mathrm{Ind}_b a$ i.e. x s.t. $a \equiv b^x \bmod p$

因子基底：$T = \{2, 3, 5, 7, 11, 13\}$

$\#T$**個の** relation：

$$\begin{pmatrix} 11^{42} \\ 11^{3} \\ 11^{29} \\ 11^{11} \\ 11^{31} \\ 11^{1} \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 15 \\ 10 \\ 39 \\ 35 \\ 11 \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 3 \times 5 \\ 2 \times 5 \\ 3 \times 13 \\ 5 \times 7 \\ 11 \end{pmatrix}$$

$$\equiv \begin{pmatrix} 11^{\mathrm{Ind}_{11} 2} \\ 11^{\mathrm{Ind}_{11} 3} \times 11^{\mathrm{Ind}_{11} 5} \\ 11^{\mathrm{Ind}_{11} 2} \times 11^{\mathrm{Ind}_{11} 5} \\ 11^{\mathrm{Ind}_{11} 3} \times 11^{\mathrm{Ind}_{11} 13} \\ 11^{\mathrm{Ind}_{11} 5} \times 11^{\mathrm{Ind}_{11} 7} \\ 11^{\mathrm{Ind}_{11} 11} \end{pmatrix} \bmod p$$

$$\begin{pmatrix} 42 \\ 3 \\ 29 \\ 11 \\ 31 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Ind}_{11}2 \\ \mathrm{Ind}_{11}3 \\ \mathrm{Ind}_{11}5 \\ \mathrm{Ind}_{11}7 \\ \mathrm{Ind}_{11}11 \\ \mathrm{Ind}_{11}13 \end{pmatrix} \mod p-1$$

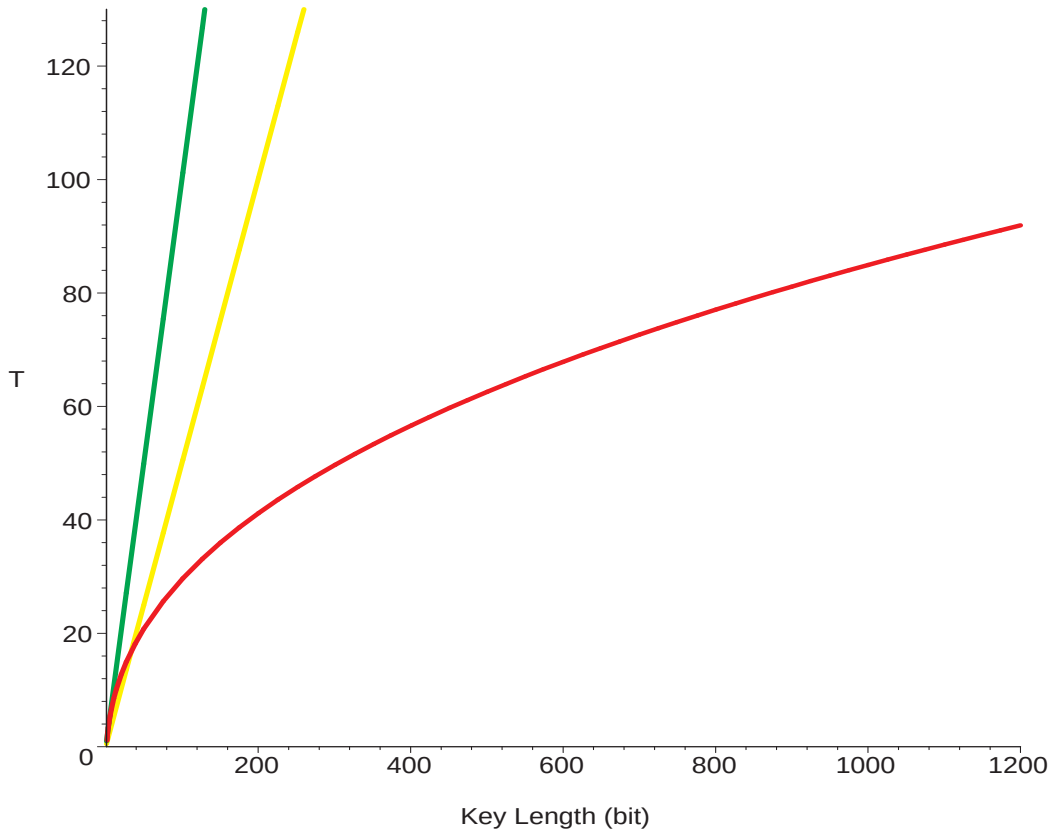
$$\begin{pmatrix} \mathrm{Ind}_{11}2 \\ \mathrm{Ind}_{11}3 \\ \mathrm{Ind}_{11}5 \\ \mathrm{Ind}_{11}7 \\ \mathrm{Ind}_{11}11 \\ \mathrm{Ind}_{11}13 \end{pmatrix} \equiv \begin{pmatrix} 42 \\ 16 \\ 33 \\ 44 \\ 1 \\ 41 \end{pmatrix} \mod p-1$$

$$\begin{aligned} 40 \times 11^{33} &\equiv 12 \\ &\equiv 2^2 \times 3 \bmod p \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \mathrm{Ind}_{11}40 &\equiv 2\mathrm{Ind}_{11}2 + \mathrm{Ind}_{11}3 - 33 \\ &\equiv 2 \times 42 + 16 - 33 \\ &\equiv 21 \bmod p-1 \end{aligned}$$

♣ 離散対数問題に必要な計算量 ♣



緑：全数探索

黄：Square-root**法**

赤：指数計算法的方法

♣ 離散対数問題の解読コスト ♣

- 解読コストはpのサイズに依存
- 2^{80}程度の手間はかけられない
  と考えられている

$\Rightarrow$ 2^{80}程度の手間が必要なpのサイズは？

- Square-root法：$\log_2 p \approx 160$
- 指数計算法　：$\log_2 p \approx 1024$ (?)

- 将来は？(漸近的計算量)：
  - Square-root法:$\log_2 p$の指数関数時間
  - 指数計算法　:$\log_2 p$の準指数関数時間

何とかならないか？ $\Rightarrow$ 離散対数問題の一般化

♣ 有限体 ♣

- 有限集合で四則演算が定義されたもの
  - $\mathbb{F}_p := \{$整数を素数pで割った余り$\}$
  - $\mathbb{F}_{p^d} := \{\mathbb{F}_p$係数の$d$次多項式の根$\}$

$$\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

♣ 有限可換群 ♣

- **有限集合で可換な演算が一つ定義され、単位元、逆元有り**
  - $+ \Rightarrow \mathbb{F}_p, (\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C})$
  - $+ \not\Rightarrow (\mathbb{N})$
  - $\times \Rightarrow \mathbb{F}_p \backslash \{0\}, (\mathbb{Q} \backslash \{0\}, \mathbb{R} \backslash \{0\}, \mathbb{C} \backslash \{0\})$
  - $\times \not\Rightarrow (\mathbb{Z})$
- $\mathbb{F}_p^* := \mathbb{F}_p \backslash \{0\}$
- **可換群の演算には$+$を用いる**

♣ 離散対数問題の一般化 ♣

- **離散対数問題**
  - Given: p: **素数**, $b \in [1, p-1]$, $a \in \{b^i | i \in [0, p-2]\}$
  - Find: $x \in [0, p-2]$ s.t. $a \equiv b^x \bmod p$

$$\Downarrow$$

- **（有限体の乗法群上の）離散対数問題**
  - Given: $\mathbb{F}_p$: **位数pの有限体**, $b \in \mathbb{F}_p^*$, $a \in \langle b \rangle$
  - Find: $x \in [0, p-2]$ s.t. $a = b^x$

$$\Downarrow$$

- **離散対数問題**
  - Given: G: **有限可換群**, $b \in G$, $a \in \langle b \rangle$
  - Find: $x \in [0, \#G - 1]$ s.t. $a = [x]b$
  - $a = [x]b = \underbrace{b + b + \cdots + b}_{x\text{個}}$

♣ $G = \mathbb{F}_p$ ♣

Given: $b \in \mathbb{F}_p,\ a \in \langle b \rangle$

Find: $x \in [0, p-1]$ s.t. $a = [x]b$

$\#G$が素数なので、

pを160 bit**程度にとれば**

square-root**法に対し安全**

ところが、

$x \in \mathbb{Z}/(p-1)\mathbb{Z}$**と考えることができるので、**

$x = a/b \in \mathbb{Z}/(p-1)\mathbb{Z}$

$\Rightarrow$

$T(p) = O((\log p)^2)$ bit-operations

♣ Pollardのρ法の一般化 ♣

Algorithm 2 Pollard's rho.alpha

Input: G: 素数, 有限可換群, $a, b \in G$

Output: $x \in [0, \#G-1]$ s.t. $a \equiv [x]b$

1: $i := 0$
2: **repeat**
3: $\quad i := i + 1$
4: $\quad$ Choose $\alpha_i, \beta_i \in [0, \#G-1]$ randomly
5: $\quad c_i = [\alpha_i]a + [\beta_i]b$
6: **until** $^{\exists}j$ s.t. $1 \le j < i, c_j = c_i$
7: $x \equiv (\beta_j - \beta_i)(\alpha_i - \alpha_j)^{-1} \bmod \#G$
$\quad$ /*$\alpha_i x + \beta_i \equiv \alpha_j x + \beta_j \bmod \#G$*/
8: Output x and terminate

（平均）時間計算量:

$O(\sqrt{\#G}) \to O\left(\sqrt{l}\right)$, l**：**$\#G$**の最大素因子**

（平均）空間計算量:

$O(\sqrt{\#G}) \to O(1)$

♣ 楕円・超楕円曲線暗号 ♣

- Square-root **法は一般に適用可**: $\sqrt{l}$,
  l **:** $\#G$ **の最大素因子**

- **有限可換群** G **で**
  指数計算法が適用できないものはあるか？

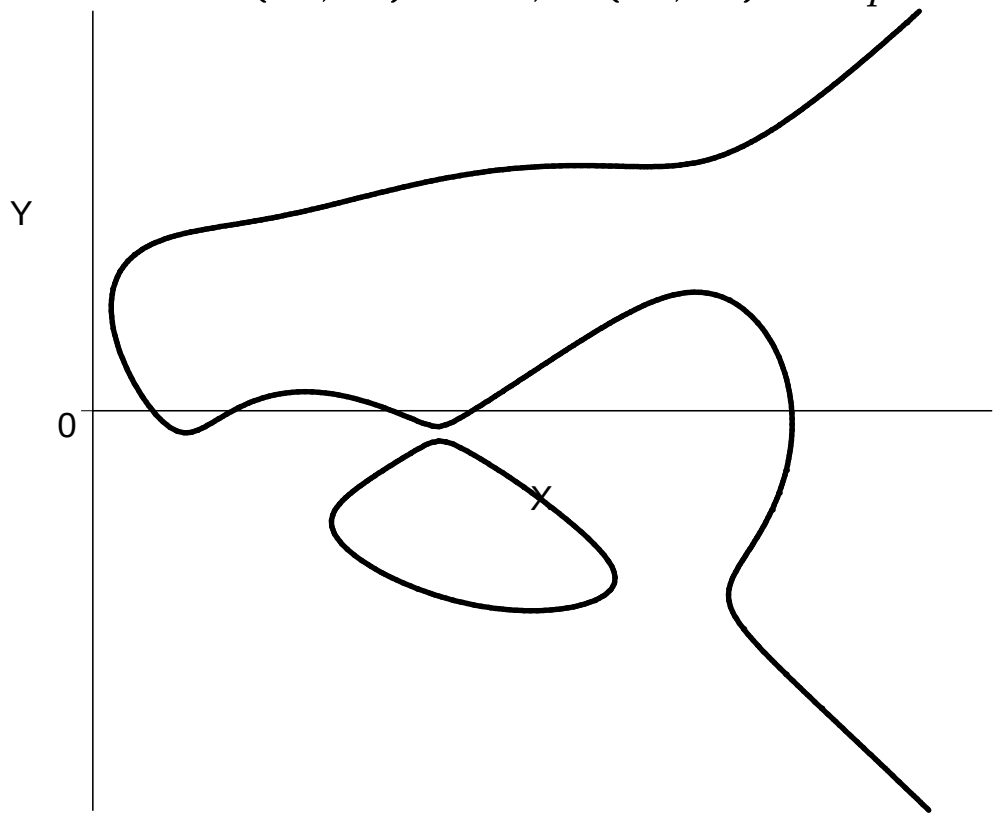
$\Rightarrow$ **代数曲線には可換群の構造を入れられる**

$\Rightarrow$ **楕円・超楕曲線円暗号**
有限体の乗法群上の離散対数問題に基づく
暗号アルゴリズムを
（有限体上の）楕円曲線、超楕円曲線の
群構造を利用して実現したもの

$\therefore$ **暗号アルゴリズム自体の研究は**
（あまり）行なわれない

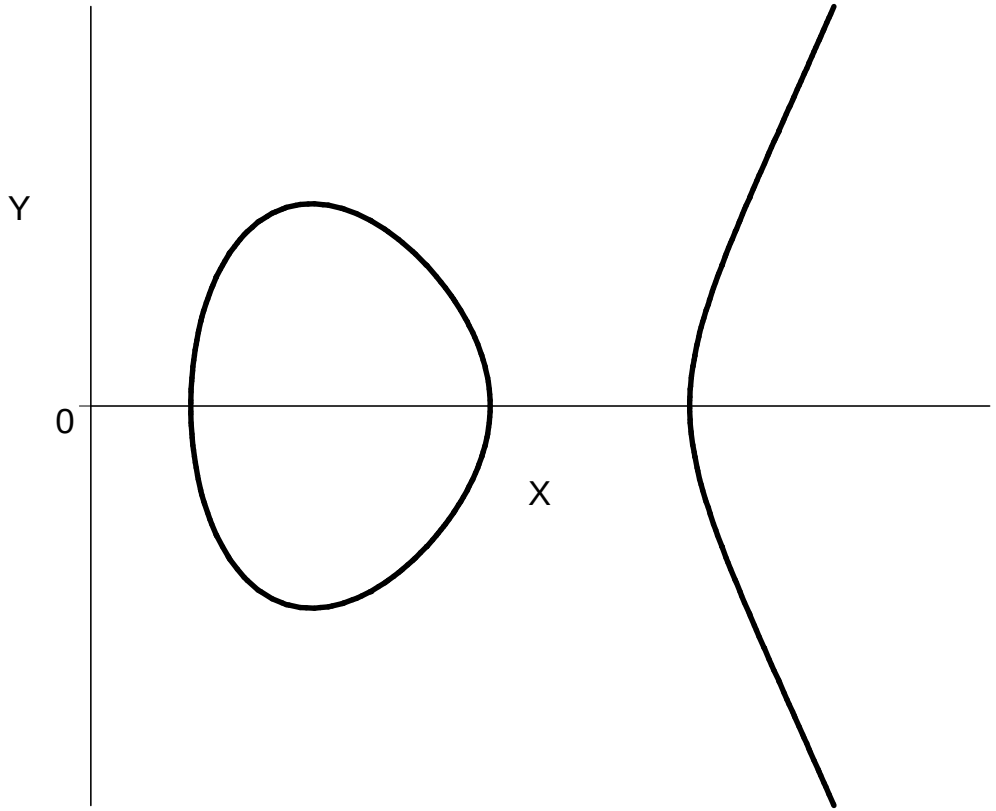
♣ 代数曲線の例 ♣

$C : F(X,Y) = 0, F(X,Y) \in \mathbb{F}_p$



♣ 楕円曲線 ♣

$$E : Y^2 = X^3 + a_4X + a_6,\ a_i \in \mathbb{F}_p$$



♣ 楕円曲線上の群構造 ♣

$$E : Y^2 = X^3 + a_4X + a_6,\ a_i \in \mathbb{F}_p$$

$$\downarrow$$

$$E(\mathbb{F}_p) := \{P = (x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + a_4x + a_6\} \cup \{P_\infty\}$$

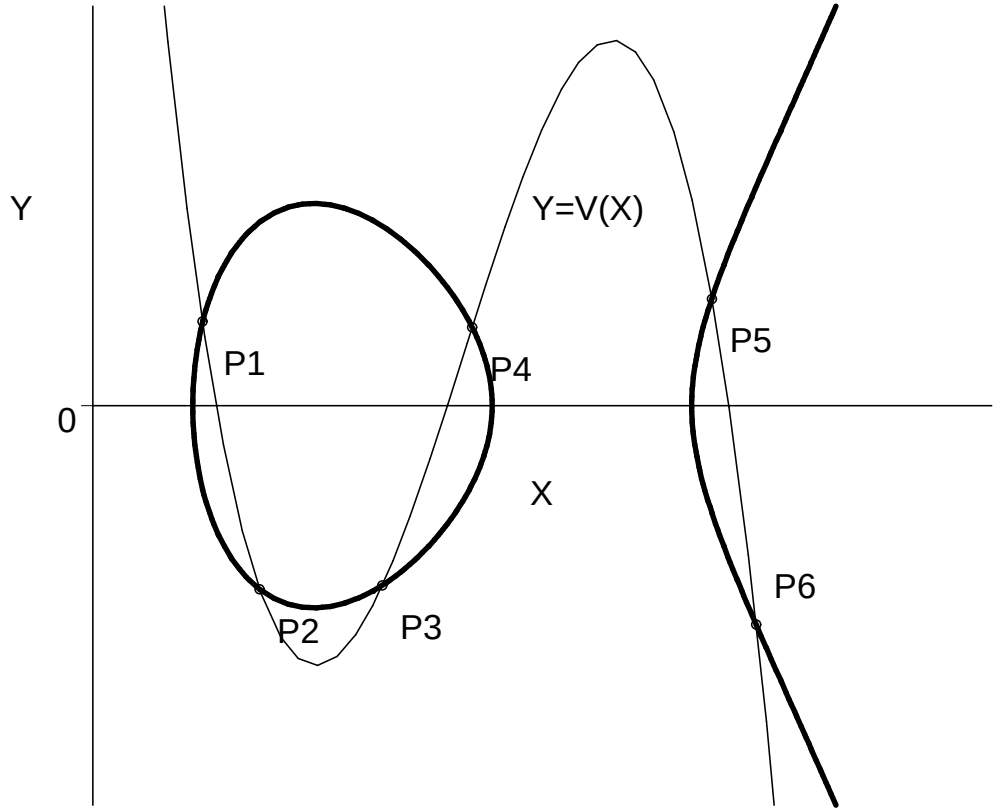
$$\downarrow$$

$E(\mathbb{F}_p)$ **は有限可換群**

$$\#E(\mathbb{F}_p) \approx p$$

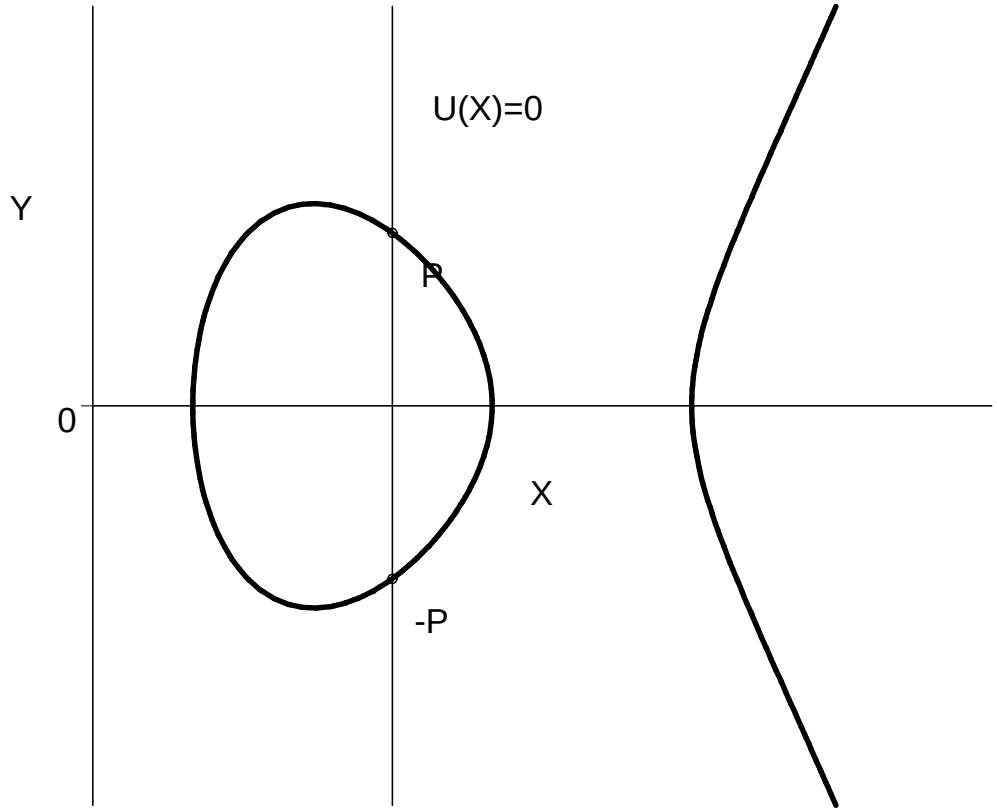
♣ 楕円曲線上の加法１ ♣

$$P_1 + P_2 + P_3 + P_4 + P_5 + P_6 = 0$$



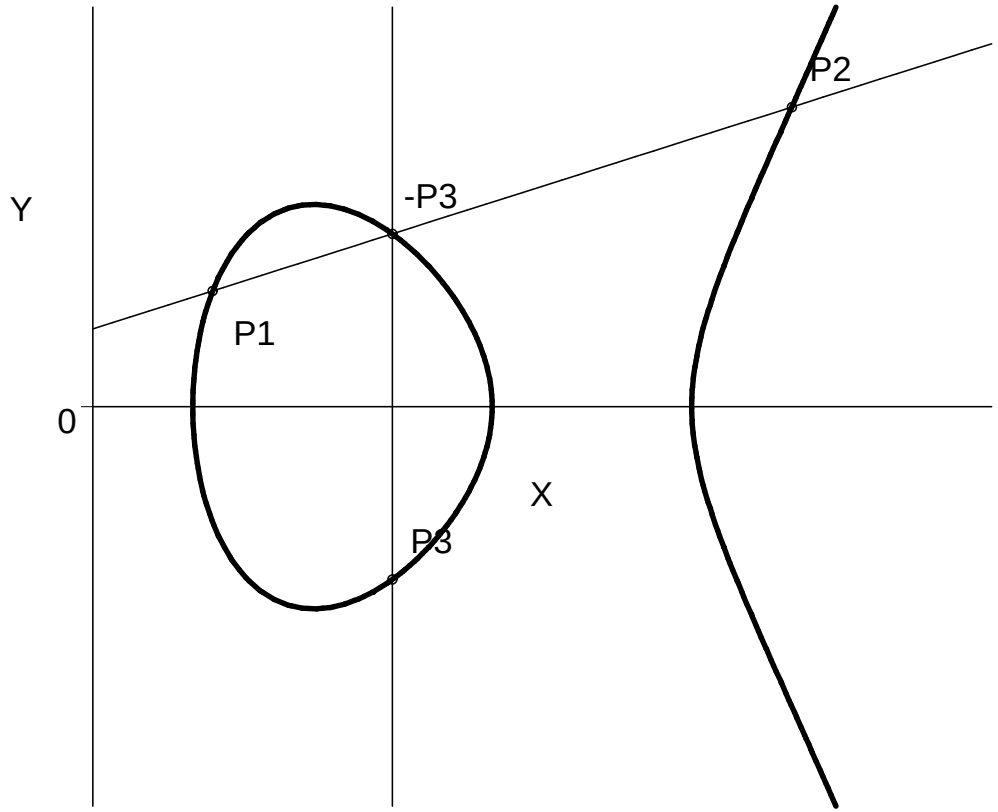
♣ 楕円曲線上の加法２ ♣

$$P = (x, y) \Rightarrow -P = (x, -y)$$



♣ 楕円曲線上の加法公式 ♣

$$P_3 = P_1 + P_2$$



♣ 楕円曲線上の加法公式 ♣

$$E : Y^2 = X^3 + a_4X + a_6$$

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2)$$

$$P_3 = (x_3, y_3) = P_1 + P_2$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \frac{3x_1^2 + a_4}{2x_1} & \text{if } P_1 = P_2 \end{cases}$$

$$x_3 = \lambda^2 - x_1 - x_2,$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

逆元計算	乗算
I	$3M$ or $4M$

♣ 楕円曲線上の加算速度 ♣

$\mathbb{F}_p$**上の演算コスト**:

$ab : M = O((\log p)^2)$

$a + b : O(\log p) \ll M$

$a^{-1} : I \approx 20M$

$-a : O(1)$

加算：$I + 3M \approx 23M$

2**倍算：**$I + 4M \approx 24M$

解読計算量が同じであるならば、
通常の離散対数問題ベースの暗号のほうが
20**倍以上速いであろう。**

逆に、同一の安全性を得るためにp**のサイズを**
$1/5$**以下にできれば、**
楕円曲線暗号のほうが速くなりそうだ。

♣ 楕円暗号の速度 ♣

楕円暗号の安全性

– $\#E(\mathbb{F}_p) = O(p)$

– Square-root**法のみ適用可**
E**の適切な選択の下**:
$O\left(\sqrt{\#E(\mathbb{F}_p)}\right) = O\left(\sqrt{p}\right)$

$\mathbb{F}_p^*$**に対する指数計算法的方法と**$E(\mathbb{F}_p)$**に対する**square-root**法の計算量を合わせると：**

$\mathbb{F}_p^*$	$E(\mathbb{F}_p)$	
512	120?	4.3
1024	160?	6.4
2048	220?	9.3

♣ 参考：安全な楕円曲線の構成 ♣

Algorithm 3 安全な楕円曲線の構成

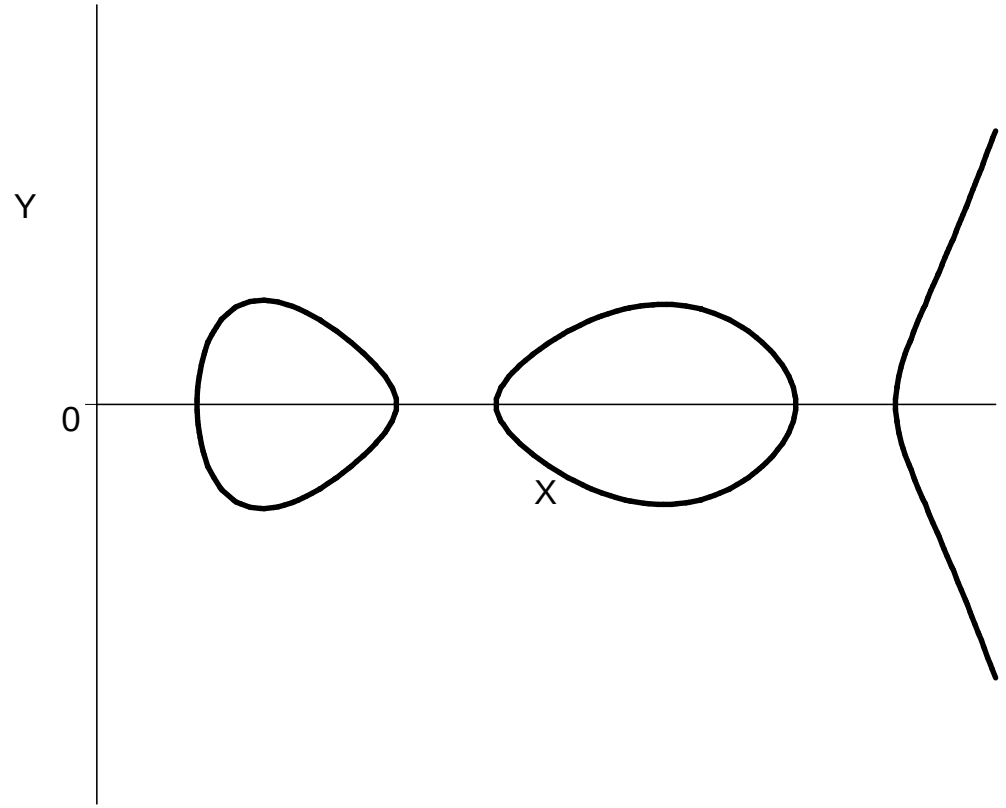
Input: p: 素数

Output: A secure elliptic curve E and $\#E(\mathbb{F}_p)$

1: **repeat**
2: **repeat**
3: Choose an elliptic curve E randmly
4: Compute $N = \#E(\mathbb{F}_p)$ /*ここが楽しい*/
5: **until** N : prime $\neq p$
6: **until** E satisfies MOV condition
7: Output $E, \#E(\mathbb{F}_p)$ and terminate

♣ 種数gの超楕円曲線 ♣

$$C : Y^2 = X^{2g+1} + f_{2g}X^{2g} + \cdots + f_1X + f_0,$$
$$f_i \in \mathbb{F}_p$$



♣ 超楕円曲線上の群構造 ♣

$$C : Y^2 = X^{2g+1} + f_{2g}X^{2g} + \cdots + f_1 X + f_0,$$
$$f_i \in \mathbb{F}_p$$

$$\downarrow$$

$$C(\mathbb{F}_p) := \{P = (x, y) \in \mathbb{F}_p^2 \mid y^2 = x^{2g+1} + \cdots + f_0\} \cup \{P_\infty\}$$

$$\downarrow$$

$C(\mathbb{F}_p)$ **は群構造を持たない**

♣ 超楕円曲線上の群構造 ♣

$$C : Y^2 = X^{2g+1} + f_{2g}X^{2g} + \cdots + f_1 X + f_0,$$
$$f_i \in \mathbb{F}_p$$

$$\downarrow$$

$$\mathcal{J}_C(\mathbb{F}_p) :=$$
$$\{D = \{P_1, \ldots, P_n \in C(\mathbb{F}_{p^g}) \setminus \{P_\infty\}\} \mid n \leq g, D^p = D\}$$

$$C(\mathbb{F}_p) \subseteq \mathcal{J}_C(\mathbb{F}_p)$$

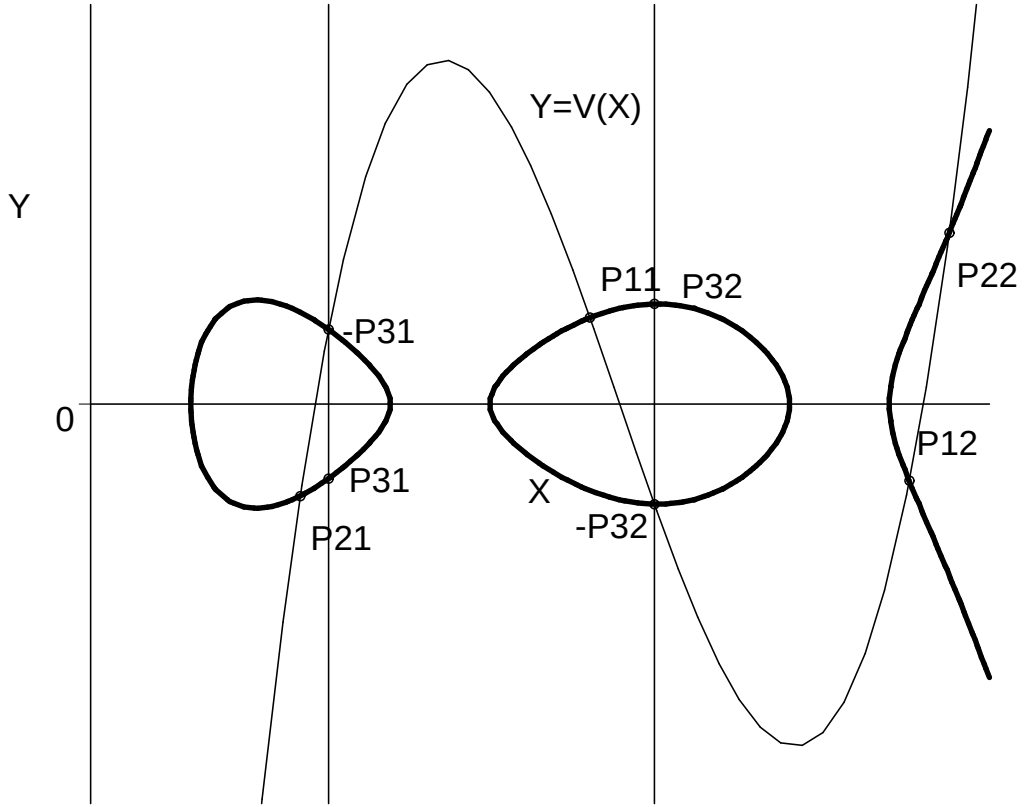
$$\downarrow$$

$\mathcal{J}_C(\mathbb{F}_p)$ **は有限可換群**

$$\#\mathcal{J}_C(\mathbb{F}_p) \approx p^g$$

♣ 超楕円曲線上の加法公式 ($g = 2$) ♣

$$D_3 = D_1 + D_2,\ D_i = \{P_{i1}, P_{i2}\}$$



♣ Mumford表現 ♣

$$C : Y^2 = F(X),\ F \in \mathbb{F}_p[X],$$
$$\deg F = 2g + 1$$

$$D = \{P_1, \ldots, P_n \in C(\mathbb{F}_{p^g}) \setminus \{P_\infty\}\} \mid n \leq g, D^p = D,$$
$$P_i = (x_i, y_i)$$

$$\Downarrow$$

$$\begin{aligned} {}^{\exists 1}(U, V) \ \in\ & (\mathbb{F}_p[X])^2 \text{ s.t.} \\ & U = \prod_{1 \leq i \leq n} (X - x_i), \\ & \deg U > \deg V, \\ & U \mid F - V^2, \\ & y_i = V(x_i). \end{aligned}$$

$$\begin{aligned} \mathcal{J}_C(\mathbb{F}_p) \ =\ & \{(U, V) \in (\mathbb{F}_p[X])^2 \mid \\ & \mathrm{lc}(U) = 1, \\ & \deg V < \deg U \leq g, \\ & U \mid F - V^2\} \end{aligned}$$

♣ 超楕円曲線上の加法公式 ♣

Input	Weight two coprime reduced divisors $D_1 = (U_1, V_1), D_2 = (U_2, V_2)$	
Output	A weight two reduced divisor $D_3 = (U_3, V_3) = D_1 + D_2$	

Step	Procedure	Cost
1	Compute the resultant r of U_1 and U_2. $z_1 \leftarrow u_{21} - u_{11}$; $z_2 \leftarrow u_{21} z_1$; $z_3 \leftarrow z_2 + u_{10} - u_{20}$; $r \leftarrow u_{10}(z_3 - u_{20}) + u_{20}(u_{20} - u_{11} z_1)$;	$4M$
2	If $r = 0$ then call the sub procedure.	—
3	Compute $I_1 \equiv 1/U_1 \bmod U_2$. $w_0 \leftarrow r^{-1}$; $i_{11} \leftarrow w_1 z_1$; $i_{10} \leftarrow w_1 z_3$;	$I + 2M$
4	Compute $S \equiv (V_2 - V_1) I_1 \bmod U_2$. (Karatsuba) $w_1 \leftarrow v_{20} - v_{10}$; $w_2 \leftarrow v_{21} - v_{11}$; $w_3 \leftarrow i_{10} w_1$; $w_4 \leftarrow i_{11} w_2$; $s_1 \leftarrow (i_{10} + i_{11})(w_1 + w_2) - w_3 - w_4(1 + u_{21})$; $s_0 \leftarrow w_3 - u_{20} w_4$;	$5M$
5	If $s_1 = 0$ then call the sub procedure.	—
6	Compute $U_3 = s_1^{-2}((S^2 U_1 + 2SV_1)/U_2 - (F - V_1^2)/(U_1 U_2))$. $w_1 \leftarrow s_1^{-1}$; $u_{30} \leftarrow w_1(w_1(s_0^2 + u_{11} + u_{21} - f_4) + 2(v_{11} - s_0 w_2)) + z_2 + u_{10} - u_{20}$; $u_{31} \leftarrow w_1(2 s_0 - w_1) - w_2$; $u_{32} \leftarrow 1$;	$I + 6M$
7	Compute $V_3 \equiv -(SU_1 + V_1) \bmod U_3$.(Karatsuba) $w_1 \leftarrow u_{30} - u_{10}$; $w_2 \leftarrow u_{31} - u_{11}$; $w_3 \leftarrow s_1 w_2$; $w_4 \leftarrow s_0 w_1$; $w_5 \leftarrow (s_1 + s_0)(w_1 + w_2) - w_3 - w_4$ $v_{30} \leftarrow w_4 - w_3 u_{30} - v_{10}$; $v_{31} \leftarrow w_5 - w_3 u_{31} - v_{11}$;	$5M$
Total		$2I + 21M$

In.	Genus 3 HEC $C : Y^2 = F(X)$, $F = X^7 + f_5 X^5 + f_4 X^4 + f_3 X^3 + f_2 X^2 + f_1 X + f_0$; Reduced divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12} X^2 + u_{11} X + u_{10}$, $V_1 = v_{12} X^2 + v_{11} X + v_{10}$, $U_2 = X^3 + u_{22} X^2 + u_{21} X + u_{20}$, $V_2 = v_{22} X^2 + v_{21} X + v_{20}$;	
Out.	Reduced divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32} X^2 + u_{31} X + u_{30}$, $V_3 = v_{32} X^2 + v_{31} X + v_{30}$;	

Step	Procedure	Cost
1	**Compute the resultant r of U_1 and U_2** $t_1 = u_{11} u_{20} - u_{10} u_{21}$; $t_2 = u_{12} u_{20} - u_{10} u_{22}$; $t_3 = u_{20} - u_{10}$; $t_4 = u_{21} - u_{11}$; $t_5 = u_{22} - u_{12}$; $t_6 = t_4^2$; $t_7 = t_3 t_4$; $t_8 = u_{12} u_{21} - u_{11} u_{22} + t_3$; $t_9 = t_3^2 - t_1 t_5$; $t_{10} = t_2 t_5 - t_7$; $r = t_8 t_9 + t_2(t_{10} - t_7) + t_1 t_6$;	$14M + 12A$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2 X^2 + i_1 X + i_0 \equiv r/U_1 \bmod U_2$** $i_2 = t_5 t_8 - t_6$; $i_1 = u_{22} i_2 - t_{10}$; $i_0 = u_{21} i_2 - (u_{22} t_{10} + t_9)$;	$4M + 4A$
4	**Compute $S' = s_2' X^2 + s_1' X + s_0' = rS \equiv (V_2 - V_1) I \bmod U_2$ (Karatsuba, Toom)** $t_1 = v_{10} - v_{20}$; $t_2 = v_{11} - v_{21}$; $t_3 = v_{12} - v_{22}$; $t_4 = t_2 i_1$; $t_5 = t_1 i_0$; $t_6 = t_3 i_2$; $t_7 = u_{22} t_6$; $t_8 = t_4 + t_6 + t_7 - (t_2 + t_3)(i_1 + i_2)$; $t_9 = u_{20} + u_{22}$; $t_{10} = (t_9 + u_{21})(t_8 - t_6)$; $t_9 = (t_9 - u_{21})(t_8 + t_6)$; $s_0' = -(u_{20} t_8 + t_5)$; $s_2' = t_6 - (s_0' + t_4 + (t_1 + t_3)(i_0 + i_2) + (t_{10} + t_9)/2)$; $s_1' = t_4 + t_5 + (t_9 - t_{10})/2 - (t_7 + (t_1 + t_2)(i_0 + i_1))$;	$10M + 31A$
5	**If $s_2' = 0$ then call the Cantor algorithm**	–
6	**Compute S, w and $w_i = 1/w$ s.t. $wS = S'/r$ and S is monic** $t_1 = (r s_2')^{-1}$; $t_2 = r t_1$; $w = t_1 s_2'^2$; $w_i = r t_2$; $s_0 = t_2 s_0'$; $s_1 = t_2 s_1'$;	$I + 7M$
7	**Compute $Z = X^5 + z_4 X^4 + z_3 X^3 + z_2 X^2 + z_1 X + z_0 = SU_1$ (Toom)** $t_6 = s_0 + s_1$; $t_1 = u_{10} + u_{12}$; $t_2 = t_6(t_1 + u_{11})$; $t_3 = (t_1 - u_{11})(s_0 - s_1)$; $t_4 = u_{12} s_1$; $z_0 = u_{10} s_0$; $z_1 = (t_2 - t_3)/2 - t_4$; $z_2 = (t_2 + t_3)/2 - z_0 + u_{10}$; $z_3 = u_{11} + s_0 + t_4$; $z_4 = u_{12} + s_1$;	$4M + 15A$
8	**Compute $U_t = X^4 + u_{t3} X^3 + u_{t2} X^2 + u_{t1} X + u_{t0} =$** **$(S(Z + 2w_i V_1) - w_i^2((F - V_1^2)/U_1))/U_2$ (Karatsuba)** $t_1 = s_0 z_3$; $t_2 = (u_{22} + u_{21})(u_{t3} + u_{t2})$; $t_3 = u_{21} u_{t2}$; $t_4 = t_1 - t_3$; $u_{t3} = z_4 + s_1 - u_{22}$; $t_5 = s_1 z_4 - u_{22} u_{t3}$; $u_{t2} = z_3 + s_0 + t_5 - u_{21}$; $u_{t1} = z_2 + t_6(z_4 + z_3) + w_i(2v_{12} - w_i) - (t_5 + t_2 + t_4 + u_{20})$; $u_{t0} = z_1 + t_4 + s_1 z_2 + w_i(2(v_{11} + s_1 v_{12}) + w_i u_{12}) - (u_{22} u_{t1} + u_{20} u_{t3})$;	$13M + 26A$
9	**Compute $V_t = v_{t2} X^2 + v_{t1} X + v_{t0} \equiv wZ + V_1 \bmod U_t$** $t_1 = u_{t3} - z_4$; $v_{t0} = w(t_1 u_{t0} + z_0) + v_{10}$; $v_{t1} = w(t_1 u_{t1} + z_1 - u_{t0}) + v_{11}$; $v_{t2} = w(t_1 u_{t2} + z_2 - u_{t1}) + v_{12}$; $v_{t3} = w(t_1 u_{t3} + z_3 - u_{t2})$;	$8M + 11A$
10	**Compute $U_3 = X^3 + u_{32} X^2 + u_{31} X + u_{30} = (F - V_t^2)/U_t$** $t_1 = 2v_{t3}$; $u_{32} = -(u_{t3} + v_{t3}^2)$; $u_{31} = f_5 - (u_{t2} + u_{32} u_{t3} + t_1 v_{t2})$; $u_{30} = f_4 - (u_{t1} + v_{t2}^2 + u_{32} u_{t2} + u_{31} u_{t3} + t_1 v_{t1})$;	$7M + 11A$
11	**Compute $V_3 = v_{32} X^2 + v_{31} X + v_{30} \equiv V_t \bmod U_3$** $v_{32} = v_{t2} - u_{32} v_{t3}$; $v_{31} = v_{t1} - u_{31} v_{t3}$; $v_{30} = v_{t0} - u_{30} v_{t3}$;	$3M + 3A$
Total		$I + 70M + 113A$

♣ 超楕円暗号の速度 ♣

- **群演算一回あたりのコスト**
  - $g = 1$**:** $I + 3M = 23M$ if $I = 20M$
  - $g = 2$**:** $I + 25M = 45M$ if $I = 20M$
  - $g = 3$**:** $I + 70M = 90M$ if $I = 20M$
- **超楕円暗号の安全性**
  - $\#E(\mathbb{F}_p) = O(p)$

    $\rightarrow$

    $\#\mathcal{J}_C(\mathbb{F}_p) = O(p^g)$
  - Square-root **法のみ適用可** (?)

    C **の適切な選択の下**:

    $O\left(\sqrt{\#\mathcal{J}_C(\mathbb{F}_p)}\right)$

♣ 超楕円暗号の速度 ♣

- **解読に2^{80}程度の手間がかかる**$p = 2^{160/g}$
  - $g = 1$: $p \approx 2^{160}$
  - $g = 2$: $p \approx 2^{80}$
  - $g = 3$: $p \approx 2^{54}$
- **群演算一回あたりのコスト**
  - $g = 1$: $I_{160} + 3M_{160} = 23M_{160}$
  - $g = 2$: $I_{80} + 25M_{80} = 45M_{80}$
  - $g = 3$: $I_{54} + 70M_{54} = 90M_{54}$

$\Rightarrow 23M_{160} > 45M_{80} > 90M_{54}$???

♣ 超楕円曲線上の離散対数問題に対する指数計算法 ♣

- Adleman-DeMarrais-Huang (1991)
  - **因子基底：素数** $< s \rightarrow$ U **の既約因子の** deg $< s$
  - **計算量**: $O(L_{p^{2g+1}}(1/2, c < 2.181))$, $\log p < (2g+1)^{0.98}$, $g \rightarrow \infty$
  - **改良の計算量**: $O(L_{p^g}(1/2, *))$, $p^g \rightarrow \infty$ Enge, Gaudry-Enge

  $\Rightarrow$ **種数の大きな曲線は暗号利用不可**
- Gaudry (1997)
  - **因子基底：**U **の既約因子の** deg $= 1$
  - **計算量**: $O(p^2)$
  - **改良の計算量**: $O(p^{2-2/g})$ Gaudry-Harley, Thériault, Nagao, Gaudry-Thomé-Thériault-Diem

♣ Gaudryの指数計算法（簡易版） ♣

$p = 7$

$$C : Y^2 = X^{13} + 5X^{12} + 4X^{11} + 6X^9 + 2X^8 + 6X^7 + 5X^4 + 5X^3 + X^2 + 2X + 6$$

$\#\mathcal{J}_C(\mathbb{F}_p) = 208697$: 18 bit **素数**

$D_a = ($
$X^6 + 2X^5 + 4X^4 + X^3 + 5X^2 + 3,$
$4X^5 + 5X^3 + 2X^2 + 5X + 4)$
$D_b = ($
$X^5 + 6X^3 + 3X^2 + 1,$
$3X^4 + X^3 + 4X^2 + X + 3)$

Find $\mathrm{Ind}_{D_b} D_a$ s.t. $D_a = [\mathrm{Ind}_{D_b} D_a] D_b$.

$C(\mathbb{F}_p) = \{P_\infty, (1,1), (1,6), (2,1), (2,6), (4,1), (4,6)(5,3), (5,4), (6,3), (6,4)\}$

$\#C(\mathbb{F}_p) = 11$

因子基底：

$T = \{(1,1), (2,1), (4,1), (5,3), (6,3)\}$

$[9343] D_b = ($
$X^5 + 6X^4 + 6X^3 + 5X^2 + 6X + 4,$
$X^4 + X^3 + X^2 + 4X + 6)$

$X^5 + 6X^4 + 6X^3 + 5X^2 + 6X + 4 = (X-1)^2(X-4)^2(X-5)$

$X^4 + X^3 + X^2 + 4X + 6 \,|_{X=1} = 6$

$X^4 + X^3 + X^2 + 4X + 6 \,|_{X=4} = 1$

$X^4 + X^3 + X^2 + 4X + 6 \,|_{X=5} = 3$

$\Rightarrow$

$[9343] D_b = -[2](1,1) + [2](4,1) + (5,3)$

$$\begin{pmatrix} [9343]D_b \\ [120243]D_b \\ [121571]D_b \\ [120688]D_b \\ [151649]D_b \end{pmatrix} =$$

$$\begin{pmatrix} -2 & 0 & 2 & 1 & 0 \\ 0 & -2 & 1 & 1 & -2 \\ -1 & 0 & 2 & -1 & -1 \\ 2 & 1 & 0 & 2 & 0 \\ 1 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} (1,1) \\ (2,1) \\ (4,1) \\ (5,3) \\ (6,3) \end{pmatrix}$$

$$\begin{pmatrix} \mathsf{Ind}_{D_b}(1,1) \\ \mathsf{Ind}_{D_b}(2,1) \\ \mathsf{Ind}_{D_b}(4,1) \\ \mathsf{Ind}_{D_b}(5,3) \\ \mathsf{Ind}_{D_b}(6,3) \end{pmatrix} \equiv \begin{pmatrix} 85159 \\ 114347 \\ 182999 \\ 22360 \\ 136908 \end{pmatrix} \mod \#\mathcal{J}_C(\mathbb{F}_p)$$

$D_a + [105454]D_b =$
$(1,1) + [2](2,1) + (4,1) - (6,3)$

$$\begin{aligned} \mathsf{Ind}_{D_b} D_a &\equiv \mathsf{Ind}_{D_b}(1,1) + 2\mathsf{Ind}_{D_b}(2,1) \\ &\quad + \mathsf{Ind}_{D_b}(4,1) - \mathsf{Ind}_{D_b}(6,3) \\ &\quad - 105454 \\ &\equiv 85159 + 2 \times 114347 \\ &\quad + 182999 - 136908 \\ &\quad - 105454 \\ &\equiv 45793 \mod \#\mathcal{J}_C(\mathbb{F}_p) \end{aligned}$$

♣ 計算量評価 ♣

$$\begin{pmatrix} [9343]D_b \\ [120243]D_b \\ [121571]D_b \\ [120688]D_b \\ [151649]D_b \end{pmatrix} = \begin{pmatrix} \cdots \\ \cdots \\ \vdots \\ \cdots \end{pmatrix} \begin{pmatrix} (1,1) \\ (2,1) \\ (4,1) \\ (5,3) \\ (6,3) \end{pmatrix}$$

- $\#T = O(p)$
- **一行を得るために必要な試行回数**
  - g **次モニック多項式の数**: $O(p^g)$
  - 1 **次式の積に分解する**
    g **次モニック多項式の数**: $O(p^g/g!)$

  $\Rightarrow O(g!)$
- Jacobian **上の加算**: $O(g^2(\log p)^2)$
- **多項式の因数分解**: $O(g^3(\log p)^3)$

$\Rightarrow O(g!g^3p(\log p)^3)$

♣ Gaudry の指数計算の計算量 ♣

疎行列の線形代数:

$O(gp^2(\log \#G)^2) = O(g^3p^2(\log p)^2)$

トータル：

$O(g!g^3p(\log p)^3) + O(g^3p^2(\log p)^2)$

小種数曲線に対しては $\tilde{O}(p^2)$ **と考えられる**

一方、種数 g **の曲線に対する** rho **法の計算量：**

$\tilde{O}(\sqrt{\#G}) = O(p^{g/2})$

$\therefore$ **種数が** 4 **を越える曲線に対して、**
rho **より速くなる可能性有**

♣ アルゴリズムの最適化 ♣

発想 (Gaudry.Harley)：

行列作成と線形代数の計算量のバランスをとる

$\Rightarrow$

因子基底をより小さく取る

$\#T = O(p^r)$, $0 < r < 1$ **とする**

$\tilde{O}(p) + \tilde{O}(p^2) \rightarrow$

$\tilde{O}\left(\frac{p^g}{p^{rg}} p^r\right) + \tilde{O}\left(p^{2r}\right) = \tilde{O}\left(p^{g+(1-g)r} + p^{2r}\right)$

$r = \frac{g}{g+1} \Rightarrow$

$\tilde{O}\left(p^{g+(1-g)r} + p^{2r}\right) = \tilde{O}\left(p^{2g/(g+1)}\right)$

種数が3**を越える曲線に対して、**
rho**より速くなる可能性有**

♣ 超楕円暗号の安全性 ♣

- **準指数時間計算量ではなく指数時間計算量**
- g **により効果が異なる**