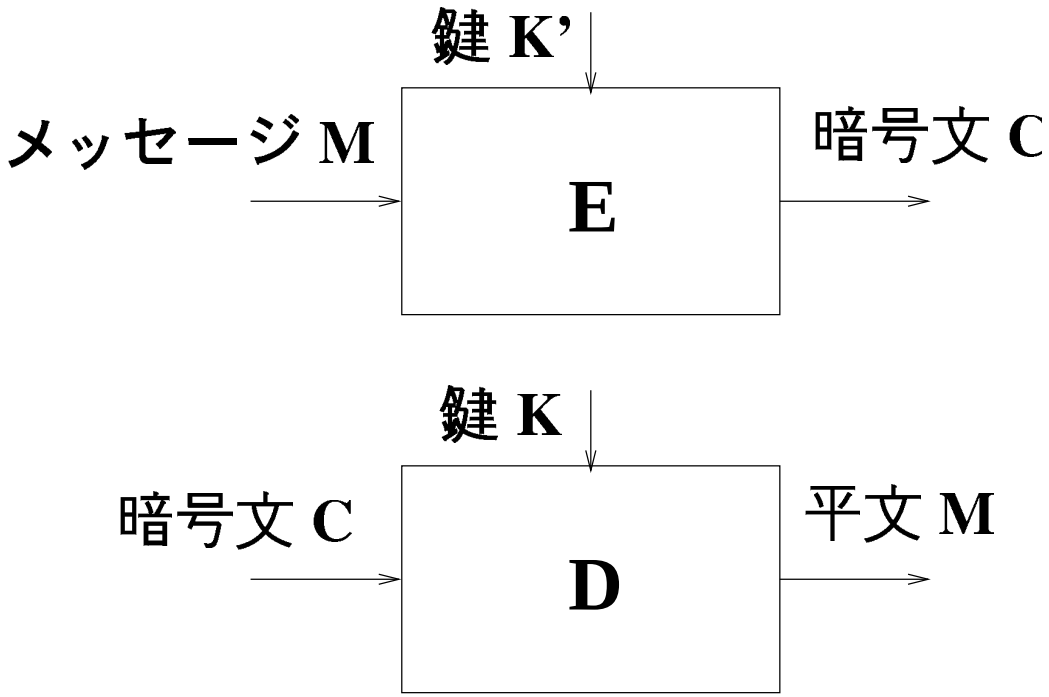
代数曲線と楕円・超楕円暗号

松尾和人

2005年4月24日

現代暗号と公開鍵暗号



- **現代暗号：** E, D **は公開、** K **は受信者の秘密**
- **公開鍵暗号：** $K' \neq K$

公開鍵暗号の例：ElGamal暗号

- **鍵生成**
  1. **素数**p**と**$b \in \{1, \ldots, p-1\}$**を適切に選択**
  2. $x \in \{0, \ldots, p-2\}$**を選択**
  3. $y \equiv b^x \bmod p$**を計算**
  - **秘密鍵：**x
  - **公開鍵：**(p, b, y)
- **暗号化**
  1. $r \in \{0, \ldots, p-2\}$**を選択**
  2. $c_1 \equiv b^r \bmod p$**を計算**
  3. $c_2 \equiv My^r \bmod p$**を計算**
  - **暗号文：**$C = (c_1, c_2)$
- **復号**
  - **平文：**$M \equiv c_2/c_1^x \bmod p$

公開鍵暗号に利用される一方向性関数

- **素因数分解（RSA等）**
  - **簡単：** $p, q \mapsto n = pq$
  - **困難：** $n \mapsto \{p, q\}$
- **離散対数問題（ElGamal等）**
  - **簡単：** $(x, b, p) \mapsto y \equiv b^x \bmod p$ **cf. 爽快！** 2^{100} **三話**
  - **困難：** $(y, b, p) \mapsto x$

離散対数問題の解読コスト

- 離散対数問題の解読コストは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

$\times$	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**
  - p**: 素数,** $b \in \{1, \ldots, p-1\}$**,** $x \in \{0, \ldots, p-2\}$
  - $y \equiv b^x \bmod p$

$$\Downarrow$$

- **（有限体の乗法群上の）離散対数問題**
  - $b \in \mathbb{F}_p^*$**,** $x \in \{0, \ldots, \#\mathbb{F}_p^* - 1\}$
  - $y = b^x$

$$\Downarrow$$

- **離散対数問題**
  - G**: 有限可換群,** $b \in G$**,** $x \in \{0, \ldots, \#G - 1\}$
  - $y = xb = \underbrace{b + b + \cdots + b}_{x \text{個}}$

楕円・超楕円暗号

- Square-root法は一般に適用可: $\sqrt{\#G}$
- 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？

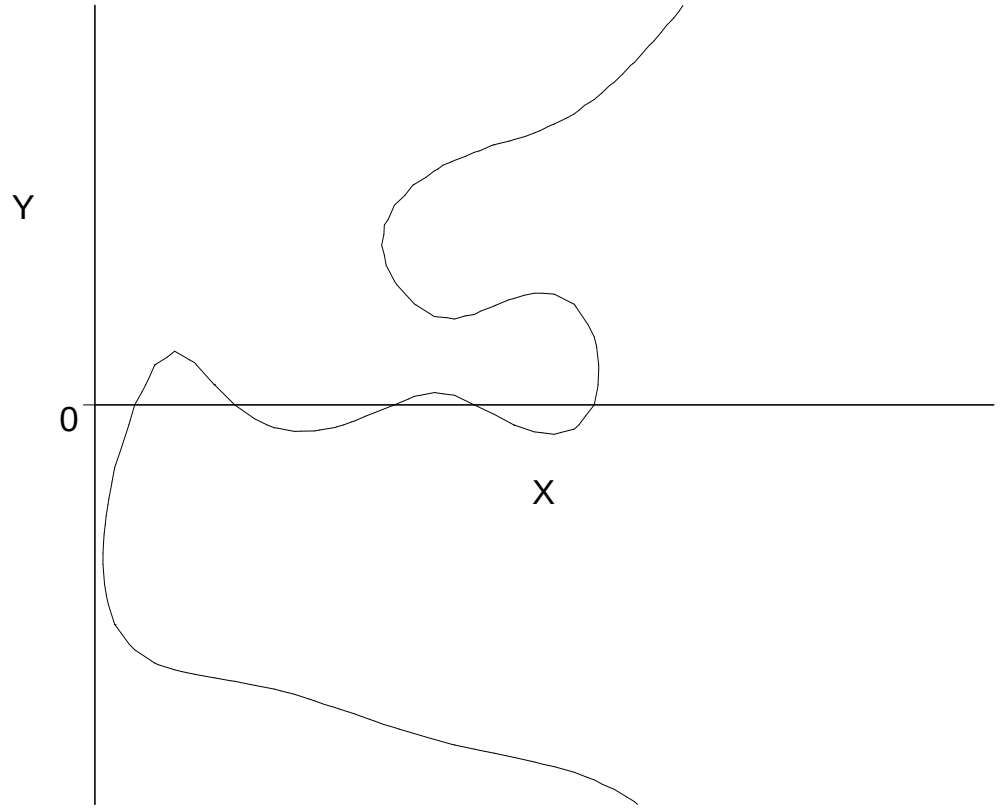
$\Rightarrow$ 代数曲線には可換群の構造を入れられる

$\Rightarrow$ 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行なわれない

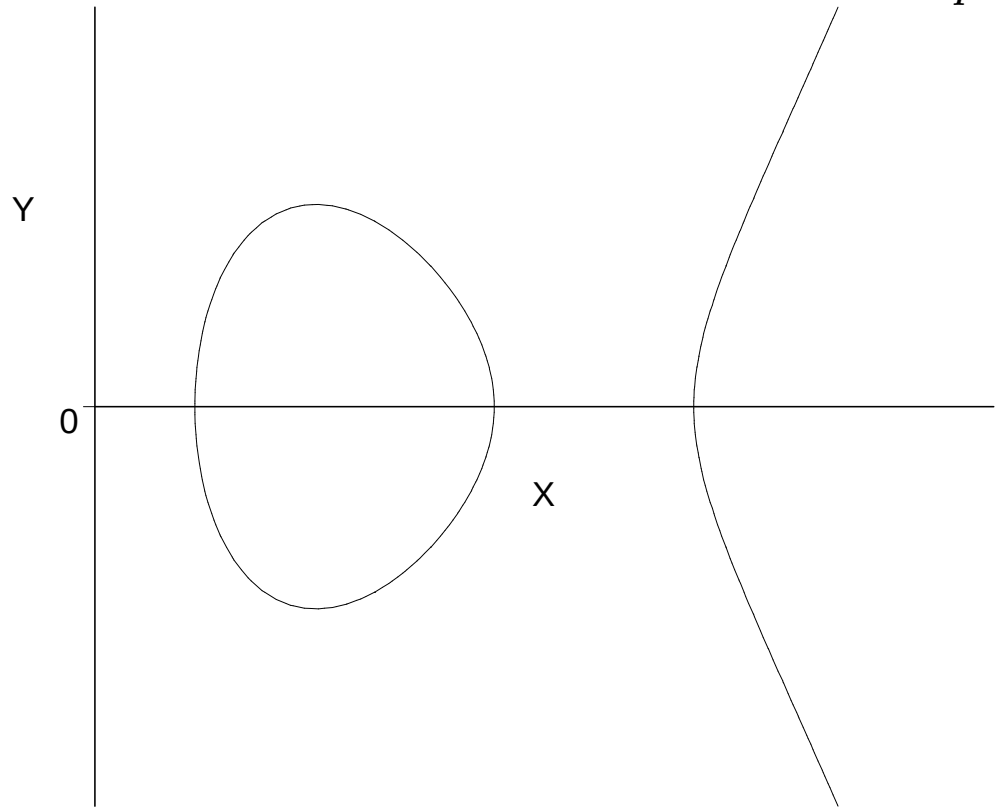
代数曲線の例

$$C : Y^4 + Y - XY^2 - X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X^2 + f_0 = 0,\ f_i \in \mathbb{F}_p$$



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

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

$$\downarrow$$

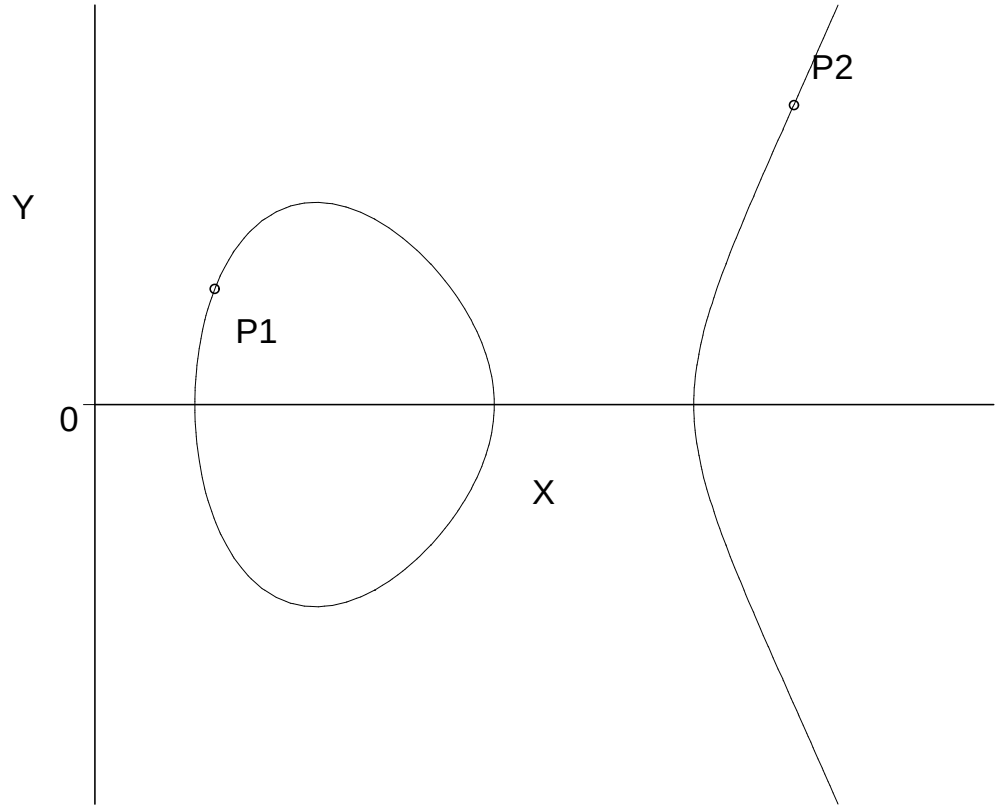
$$E(\mathbb{F}_p) := \{P = (x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + a_4x + a_6\} \cup \{P_\infty\}$$

$$\downarrow$$

$E(\mathbb{F}_p)$ **は有限可換群**

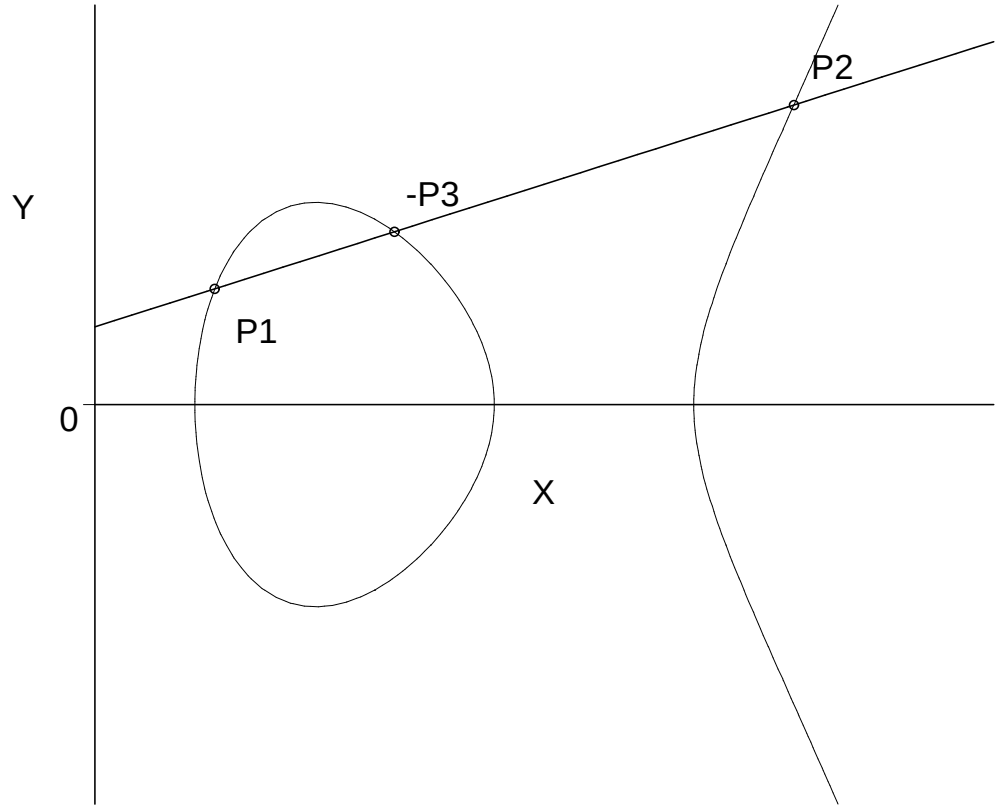
楕円曲線上の加法公式

$$P_3 = P_1 + P_2$$



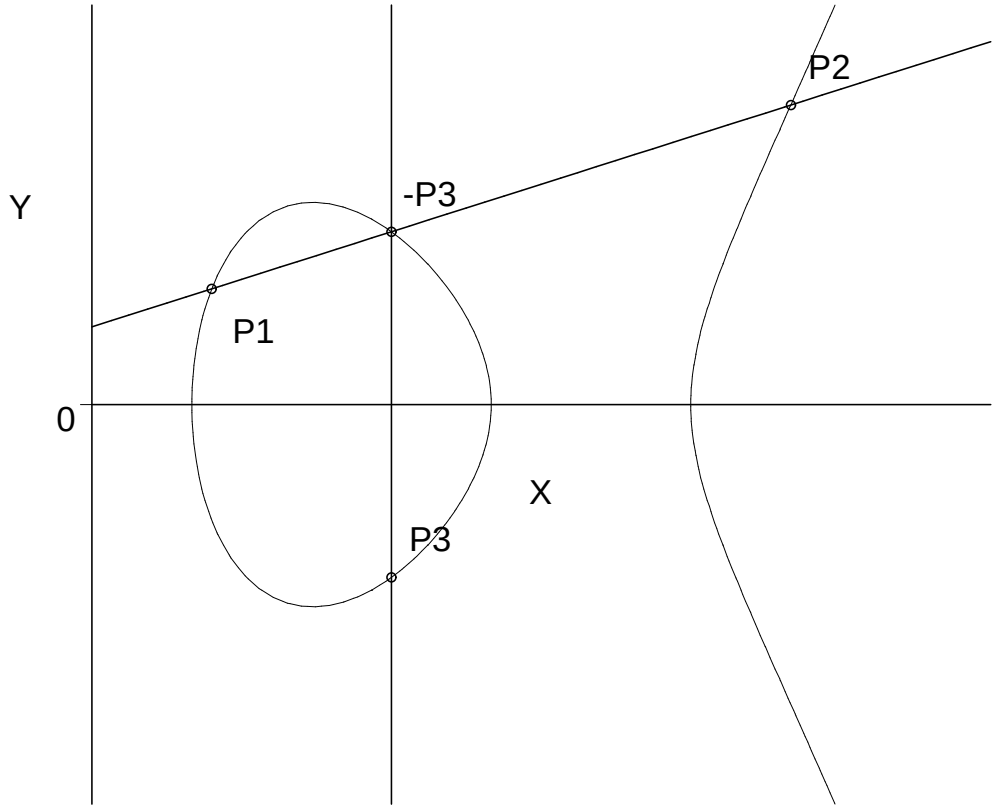
楕円曲線上の加法公式

$$P_3 = P_1 + P_2$$



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

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

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

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

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

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned}$$

逆元計算	乗算
I	$3M$

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:** $O\left(\sqrt{\#E(\mathbb{F}_p)}\right)$
    $\Rightarrow\ p = O\left(2^{160}\right)$

- **群演算一回あたりのコスト**
  - **有限体の乗法群：** M_{1024}
  - **楕円曲線：** $I_{160} + 3M_{160} = 23M_{160}$ **if** $I_{160} = 20M_{160}$

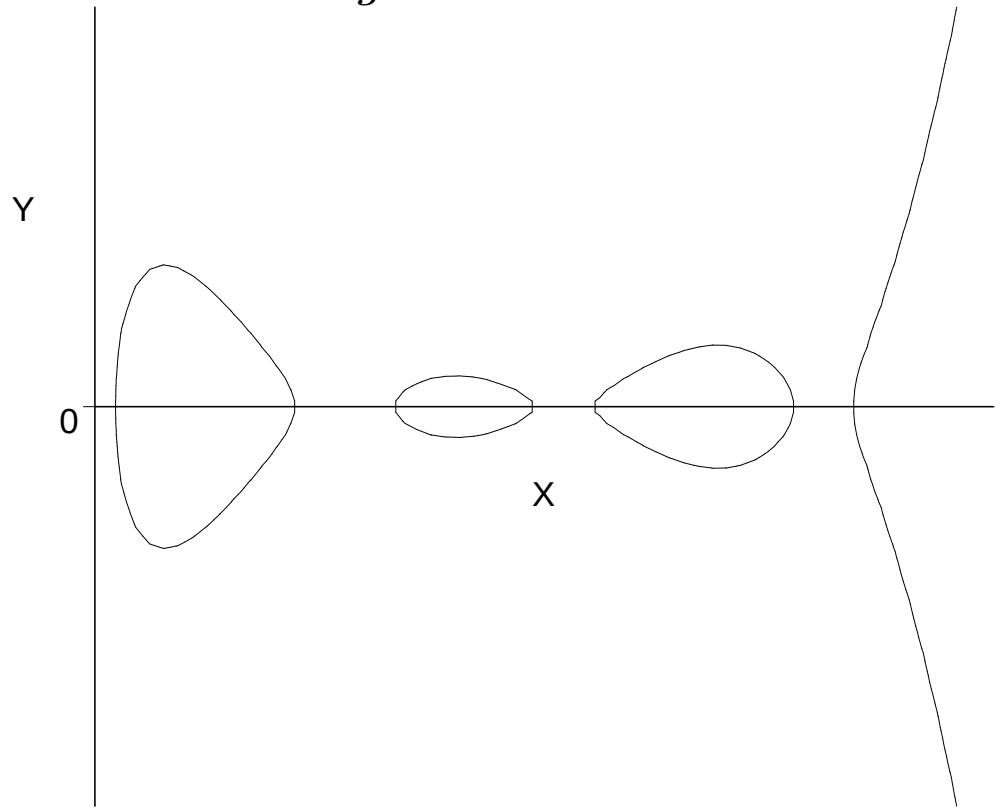
$\Rightarrow M_{1024} > 23M_{160}$

楕円暗号の速度

	加算(c.)	2倍算(c.)	整数倍算	
Aoki et al.	2692	2528	573μs	Pentium II 450MHz
(ICICS02)	1524	1164	254μs	Alpha EV6 500MHz

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

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

$$\downarrow$$

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

$$\downarrow$$

$C(\mathbb{F}_p)$ **は群構造を持たない**

超楕円曲線上の群構造

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

$$\downarrow$$

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

$$C(\mathbb{F}_p) \subseteq J_C(\mathbb{F}_p)$$

$$\downarrow$$

$J_C(\mathbb{F}_p)$ **は有限可換群**

Mumford表現

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

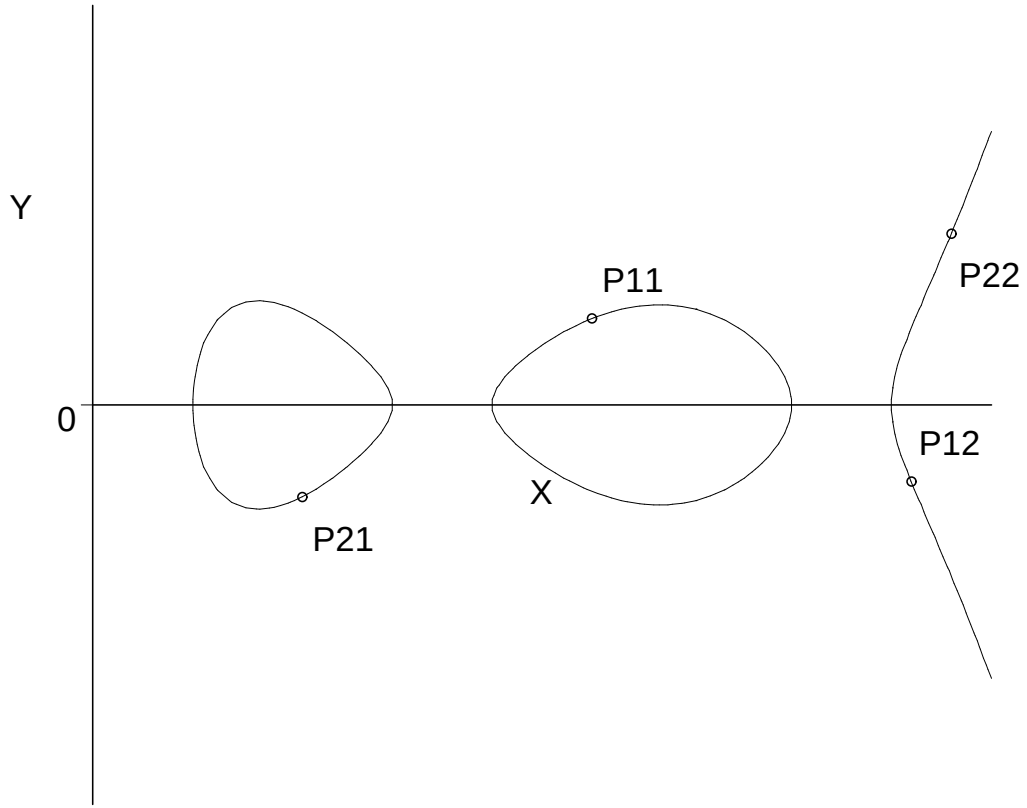
$$D = \{P_1, \ldots, P_n \in C(\mathbb{F}_{p^g}) \setminus \{P_\infty\}\} \mid n \le g, D^p = D,\ P_i = (x_i, y_i)$$

$$\Downarrow$$

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

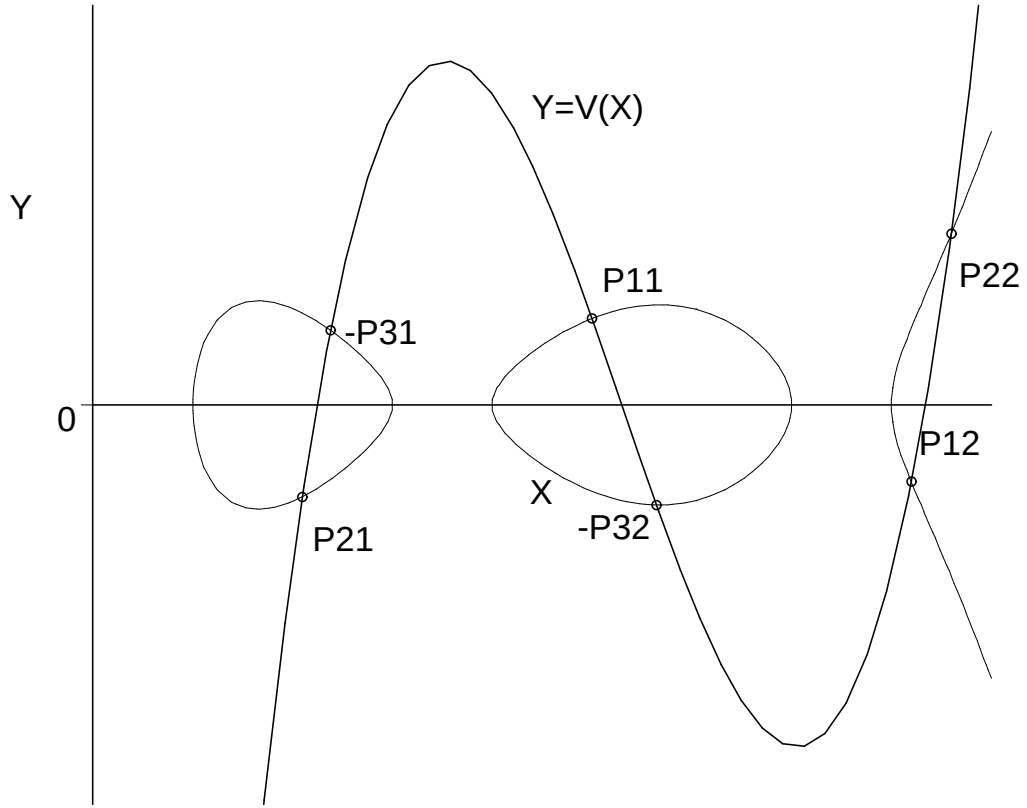
超楕円曲線上の加法公式 ($g = 2$)

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



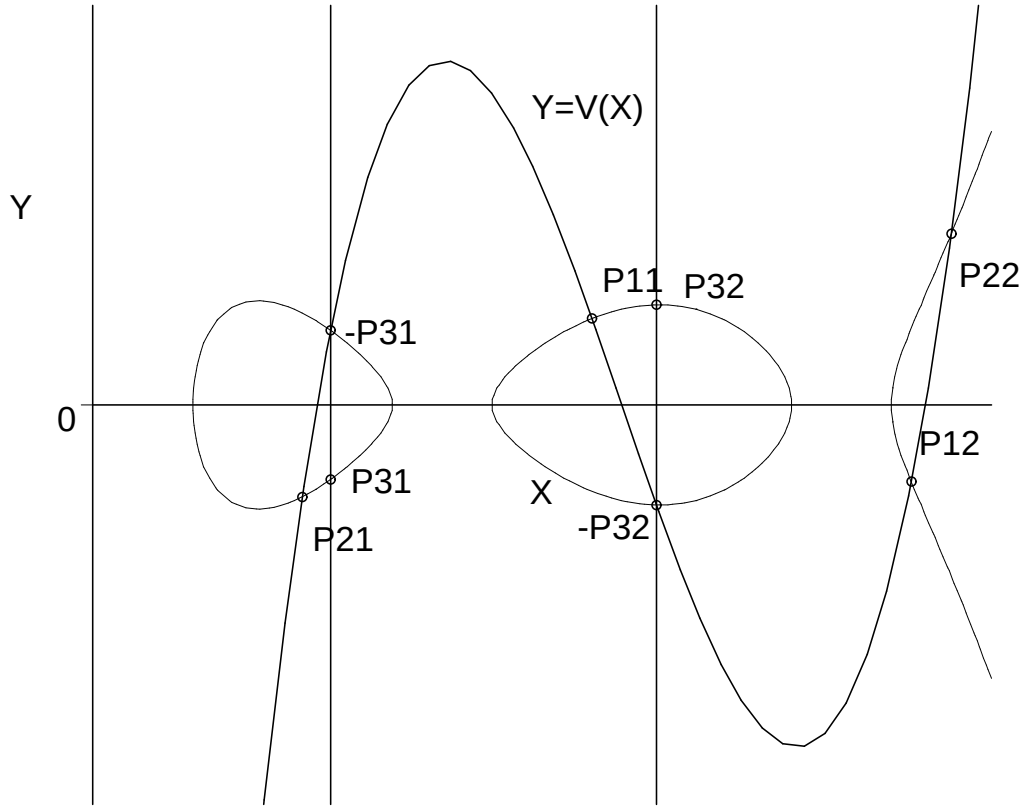
超楕円曲線上の加法公式 ($g = 2$)

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



超楕円曲線上の加法公式 ($g = 2$)

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



超楕円曲線上の加法公式 ($g = 2$)

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 $U_3 = s_1^{-2}((S^2 U_1 + 2SV_1)/U_2 - (F - V_1^2)/(U_1 U_2))$.** $w_1 \leftarrow s_1^{-1}$; $u_{30} \leftarrow w_1(w_1(s_0^2 + u_{11} + u_{21} - f_4) + 2(v_{11} - s_0 w_2)) + z_2 + u_{10} - u_{20}$; $u_{31} \leftarrow w_1(2s_0 - w_1) - w_2$; $u_{32} \leftarrow 1$;	$I + 6M$
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		$2I + 21M$

超楕円曲線上の加法公式 ($g = 3$)

In.	**Genus 3 HEC** $C : Y^2 = F(X)$, $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; **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 $t_1 = u_{11}u_{20} - u_{10}u_{21}$; $t_2 = u_{12}u_{20} - u_{10}u_{22}$; $t_3 = u_{20} - u_{10}$; $t_4 = u_{21} - u_{11}$; $t_5 = u_{22} - u_{12}$; $t_6 = t_4^2$; $t_7 = t_3t_4$; $t_8 = u_{12}u_{21} - u_{11}u_{22} + t_3$; $t_9 = t_3^2 - t_1t_5$; $t_{10} = t_2t_5 - t_7$; $r = t_8t_9 + t_2(t_{10} - t_7) + t_1t_6$;	$14M + 12A$
2	**If** $r = 0$ **then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse** $I = i_2X^2 + i_1X + i_0 \equiv r/U_1 \bmod U_2$ $i_2 = t_5t_8 - t_6$; $i_1 = u_{22}i_2 - t_{10}$; $i_0 = u_{21}i_2 - (u_{22}t_{10} + t_9)$;	$4M + 4A$
4	**Compute** $S' = s_2'X^2 + s_1'X + s_0' = rS \equiv (V_2 - V_1)I \bmod U_2$ **(Karatsuba, Toom)** $t_1 = v_{10} - v_{20}$; $t_2 = v_{11} - v_{21}$; $t_3 = v_{12} - v_{22}$; $t_4 = t_2i_1$; $t_5 = t_1i_0$; $t_6 = t_3i_2$; $t_7 = u_{22}t_6$; $t_8 = t_4 + t_6 + t_7 - (t_2 + t_3)(i_1 + i_2)$; $t_9 = u_{20} + u_{22}$; $t_{10} = (t_9 + u_{21})(t_8 - t_6)$; $t_9 = (t_9 - u_{21})(t_8 + t_6)$; $s_0' = -(u_{20}t_8 + t_5)$; $s_2' = t_6 - (s_0' + t_4 + (t_1 + t_3)(i_0 + i_2) + (t_{10} + t_9)/2)$; $s_1' = t_4 + t_5 + (t_9 - t_{10})/2 - (t_7 + (t_1 + t_2)(i_0 + i_1))$;	$10M + 31A$
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** $t_1 = (rs_2')^{-1}$; $t_2 = rt_1$; $w = t_1s_2'^2$; $w_i = rt_2$; $s_0 = t_2s_0'$; $s_1 = t_2s_1'$;	$I + 7M$
7	**Compute** $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$ **(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** $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + 2w_iV_1) - w_i^2((F - V_1^2)/U_1))/U_2$ **(Karatsuba)** $t_1 = s_0z_3$; $t_2 = (u_{22} + u_{21})(u_{t3} + u_{t2})$; $t_3 = u_{21}u_{t2}$; $t_4 = t_1 - t_3$; $u_{t3} = z_4 + s_1 - u_{22}$; $t_5 = s_1z_4 - u_{22}u_{t3}$; $u_{t2} = z_3 + s_0 + t_5 - u_{21}$; $u_{t1} = z_2 + t_6(z_4 + z_3) + w_i(2v_{12} - w_i) - (t_5 + t_2 + t_4 + u_{20})$; $u_{t0} = z_1 + t_4 + s_1z_2 + w_i(2(v_{11} + s_1v_{12}) + w_iu_{12}) - (u_{22}u_{t1} + u_{20}u_{t3})$;	$13M + 26A$
9	**Compute** $V_t = v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1 \bmod U_t$ $t_1 = u_{t3} - z_4$; $v_{t0} = w(t_1u_{t0} + z_0) + v_{10}$; $v_{t1} = w(t_1u_{t1} + z_1 - u_{t0}) + v_{11}$; $v_{t2} = w(t_1u_{t2} + z_2 - u_{t1}) + v_{12}$; $v_{t3} = w(t_1u_{t3} + z_3 - u_{t2})$;	$8M + 11A$
10	**Compute** $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F - V_t^2)/U_t$ $t_1 = 2v_{t3}$; $u_{32} = -(u_{t3} + v_{t3}^2)$; $u_{31} = f_5 - (u_{t2} + u_{32}u_{t3} + t_1v_{t2})$; $u_{30} = f_4 - (u_{t1} + v_{t2}^2 + u_{32}u_{t2} + u_{31}u_{t3} + t_1v_{t1})$;	$7M + 11A$
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**
  - $\#E(\mathbb{F}_p) = O(p) \rightarrow \#J_C(\mathbb{F}_p) = O(p^g)$
  - **Square-root 法のみ適用可 (?)**
    C **の適切な選択の下:** $O\left(\sqrt{\#J_C(\mathbb{F}_p)}\right)$

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
  - $g = 1$：$I_{160} + \ 3M_{160} = 23M_{160}$
  - $g = 2$：$I_{80} + 25M_{80} = 45M_{80}$
  - $g = 3$：$I_{54} + 70M_{54} = 90M_{54}$

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

超楕円暗号の速度

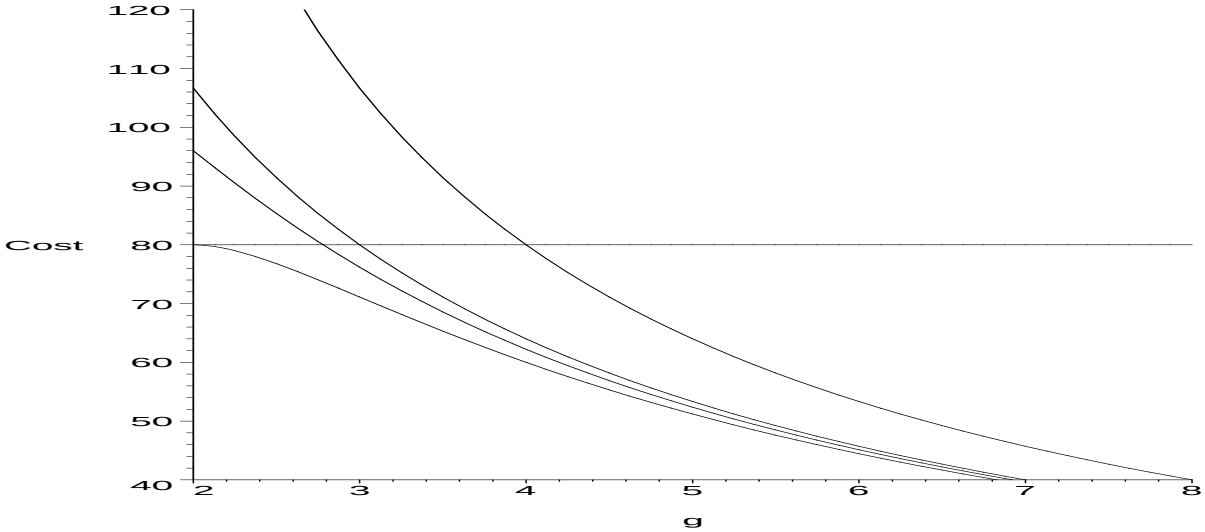
	加算(c.)	2倍算(c.)	整数倍算	
$g = 1$	2692	2528	573μs	Pentium II 450MHz
Aoki et al.	1524	1164	254μs	Alpha EV6 500MHz
(ICICS02)				
$g = 3$	1110	1094	172μs	Alpha EV68 1.25GHz
(SCIS04)				

超楕円暗号の安全性

指数計算法が効果を持ってしまう？

($C(\mathbb{F}_p) \subset J_C(\mathbb{F}_p)$をFBとして用いる)

- 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



安全な楕円・超楕円曲線の構成
数学的に最も興味深い?

- $\#G$ $(\#E(\mathbb{F}_p), \#J_C(\mathbb{F}_p))$ はCやEの選び方により変動する
- Square-root 法の計算量は$\#G$の最大素因子に依存

$\Rightarrow$

- $\#G$ の計算アルゴリズムが必要
- $\#G$を知れば、特殊な曲線に対する攻撃も（大体）回避可能

楕円・超楕円曲線の位数計算

Genus	Field	Size of $\#G$	Running Time
1	$\mathbb{F}_{2^d}$	50021	36h / EV6 750MHz
	$\mathbb{F}_p$	1660	10h / Opteron 2.4GHz
		3322	8d / ↑
		4983	267d / ↑
2	$\mathbb{F}_{2^d}$	65540	8d / EV6 750MHz
3	$\mathbb{F}_{2^d}$	168	69s / Athlon 1.47GHz
		480	67m / ↑

超楕円曲線の位数計算

Genus	Field	Size of $\#G$	Running Time
2	$\mathbb{F}_p$	127	50d / EV6 500MHz
	$\mathbb{F}_{p^d}$	160	21d / Athlon 1.67GHz
	$\mathbb{F}_p$	160	7d / EV67 667MHz
3	$\mathbb{F}_p$	?	

まとめ

研究課題	楕円暗号	超楕円暗号
高速化 安全性 位数計算		

まとめ

研究課題	楕円暗号	超楕円暗号
高速化	○	
安全性	△	
位数計算	○	

まとめ

研究課題	楕円暗号	超楕円暗号
高速化	○	△
安全性	△	△
位数計算	○	△

超楕円暗号の研究は楽しい

参考文献

[1] L. C. Washington. Elliptic Curves: Number Theory and Cryptography. CRC Pr., 2003.

[2] H. Cohen, G. Frey, R. Avanzi, C. Doche, K. Nguyen, T. Lange. Elliptic and Hyperelliptic Curve Crytography. Chapman & Hall, June 2005.

[3] 松尾 和人, 有田 正剛, 趙 晋輝. 代数曲線上の公開鍵暗号. 情報処理, 45(11):1114–1116, November 2004.

[4] 松尾 和人, 有田 正剛, 趙 晋輝. 代数曲線暗号. 日本応用数理学会論文誌, 13(2):231–243, June 2003.

[5] 本２冊