第 15 回 整数論サマースクール 報告集, pp.223-238

代数曲線暗号とその安全性

松尾 和人*

1 はじめに

本稿では「代数曲線暗号」とその安全性に関する議論を紹介する。代数曲線暗号の研究者には整数論出身者が多く、また多くの整数論研究者がその中に問題を見出し研究に取り組んでいる。整数論を学んだ方が新たに代数曲線暗号の研究を始めるときや整数論を専門とする研究者の方々が関連研究を新たに始めるときに、その研究の暗号学的な背景や意義を知る手掛かりとなることを意図して書いた。この主旨の下、研究の流れに沿って一本の筋を通した記述を行った。その結果いくつかの重要な結果について触れることができなかったことに御留意頂きたい。

2 公開鍵暗号と離散対数問題

（代数曲線暗号が含まれる）公開鍵暗号は 1976 年に Diffie と Hellman によって提案された [12]。この Diffie と Hellman のプロトコルは事前に秘密情報のやりとりをせずに共通鍵暗号に利用する共通鍵を二者間で共有しようというものである。表 1 に Diffie-Hellman プロトコルを示す。

	システム設定	
	p: 素数, $b \in \mathbb{F}_p^*$ (s.t. $\langle b \rangle = \mathbb{F}_p^*$)	
	太郎	花子
	鍵ペア生成	
秘密鍵設定	$K_a \in \mathbb{Z}/(p-1)\mathbb{Z}$	$K_b \in \mathbb{Z}/(p-1)\mathbb{Z}$
公開鍵計算	$K_a' = b^{K_a}$	$K_b' = b^{K_b}$
鍵公開	公開鍵 K_a' を公開	K_b' を公開
	共通鍵計算	
	$K = K_b'^{K_a}$	$K = K_a'^{K_b}$
	同一の鍵 K を共有できた	

表 1: Diffie-Hellman 鍵共有プロトコル

*情報セキュリティ大学院大学

Diffie と Hellman の提案の後、Rivest, Shamir, Adleman [35] によって RSA 暗号・署名が ElGamal [13] によって ElGamal 暗号・署名が提案された。この中で RSA 暗号・署名は素因数分解の困難性に基づいた暗号プロトコロルであり、Diffie-Hellman, ElGamal は離散対数問題の困難性に基づいたアルゴリズムである。

「共通鍵計算」の速度が Diffie-Hellman プロトコルの暗号化速度であるが、これは明らかに p に依存する。この計算は $K_a \in \mathbb{Z}/(p-1)\mathbb{Z}$ を整数と看做し $K_a = (x_{k-1}x_{k-2}\dots x_1x_0)_2$ と 2 進展開すれば $K = \prod_{0\le i<k} K_b'^{2^{x_i}}$ と計算され、$k = O(\log p)$ より $O(\log p)$ 回の $\mathbb{F}_p$-乗算によって実現される。また、$\mathbb{F}_p$-乗算は（暗号アルゴリズムに利用される）標準的な方法では $O((\log p)^2)$ のビット演算量を必要とする。従って p が大きくなるに連れて暗号化速度が遅くなりプロトコルの実用性は低くなる。一方、p を小さくとると $K_a' = b^{K_a}$ に対する全数探索により K_a を知ることが可能となり、$K = K_b'^{K_a}$ から誰でも秘密 K を知ることが可能な（「暗号」としての機能を持たない）プロトコルとなる。従って、p は K が求められない程度に大きくとる必要がある。この K_a を求める問題を一般に離散対数問題という。

定義 2.1 (離散対数問題) 与えられた $b \in \mathbb{F}_p^*$, $a \in \langle b\rangle$ に対し $a = b^x$ を満足する $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ を求める問題を $\mathbb{F}_p$ 上の離散対数問題という。また、この x を $\mathrm{Ind}_b a$ と書く。

離散対数問題は全数探索により $O(p)$ の $\mathbb{F}_p$-演算で解くことが可能である。現在のところ 2^{80} 程度の手間の掛かる計算は不可能であると考えられているので、もし離散対数問題の解法として全数探索が最良であるならば、80 ビット程度の p を利用すれば安全な暗号が得られる。しかし、全数探索より効率的な離散対数問題の解法アルゴリズムが存在するならば、安全性を確保するためにはより大きな p を選択する必要があり、それを利用した Diffie-Hellman 等の暗号プロトコルの効率が悪くなる。また、もし $\log p$ の低次多項式時間のアルゴリズムが存在した場合には、もはや Diffie-Hellman プロトコル等を暗号アルゴリズムと呼ぶことは出来ない。

注意 2.1 離散対数問題が解ければ *Diffie-Hellman* プロトコルを破れるが、離散対数問題を解かずに *Diffie-Hellman* プロトコルを破る方法がないことは示されていない。

3 離散対数問題の解法

前節を受け本節では定義 2.1 で与えられた離散対数問題の解法について紹介する。離散対数問題の解法として全数探索より効率的な方法が 2 種類知られている。その一つは square-root 法と呼ばれる方法であり、もう一つは（一般に）より効率的な指数計算法である。ここでは、まず square-root 法を紹介し、次に指数計算法を紹介する。

3.1 Square-root 法

良く知られた square-root 法に、Shanks [36] の baby-step giant-step アルゴリズムと Pollard [34] の rho 法がある。これらのアルゴリズムは同一の漸近計算量を持つがその性質は大きく異なる。現実的な離散対数問題に対してはメモリー効率の優位性から rho 法を用い

ることが通常である。そこでここではrho法を紹介する。また、中国の剰余定理を利用したこれらのアルゴリズムの効率向上策[33]が知られているので、これについても紹介する。

3.1.1 Pollardのrho法

Rho法は「誕生日のパラドクス」を利用したアルゴリズムである。誕生日のパラドクスについては例えば[9, Section 5.4.1]を参照されたい。Algorithm 1にrho法の原型を示す。

Algorithm 1 Rho法の原型

Input: p: prime, $b \in \mathbb{F}_p^*$, $a \in \langle b \rangle$
Output: $x \in [0, p-2]$ s.t. $a = b^x$

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

Algorithm 1のループ回数の期待値は誕生日のパラドクスより$O(\sqrt{p})$となる。従ってこれの計算量は$O(\sqrt{p})$ $\mathbb{F}_p$-演算であり全数探索と比較し大幅に効率的である。実際、これにより例えばpが160ビット程度のとき離散対数問題の解読は2^{80}倍程度高速化することが見込まれる。

例 3.1 *Rho*法の原型による離散対数計算の具体例として与えられた$p = 47$, $a = 40$, $b = 11$に対し$a \equiv b^x \bmod p$を満足するxを求める。

下表のように$i = 1, 2, \ldots$に対しα_i $\beta_i \in [0, 45]$をランダムに選択し$c_i \equiv a^{\alpha_i} b^{\beta_i} \bmod p$を計算していく。

i	1	2	3	4	5	6	7	8	9	10
α_i	35	36	17	9	3	17	16	37	38	39
β_i	3	41	15	0	28	14	7	17	25	8
c_i	27	43	24	29	<u>30</u>	15	40	6	13	<u>30</u>

表から*10*ステップの計算の後$i = 10$において計算した結果が*5*ステップ目の計算結果と一致することが判る。従って

$$a^{\alpha_5} b^{\beta_5} \equiv a^{\alpha_{10}} b^{\beta_{10}} \bmod p$$

であり、

$$x \equiv \frac{\beta_{10} - \beta_5}{\alpha_5 - \alpha_{10}} \equiv 21 \bmod p - 1$$

を得る。

注意 3.1 ここで示したアルゴリズムは、テーブルサイズの多項式時間で探索可能なデータベースに全ての (c_i, α_i, β_i) を記録する必要があり、空間計算量 $O(\sqrt{p})$ を必要とする。そこで通常は「ランダムウォーク関数」を利用して空間計算量を $O(1)$ とした変形が利用される。このランダムウォーク関数の選択などについても多くの研究がなされている。これらについては *[39]* 等を参照されたい。

3.1.2 中国の剰余定理の利用

Square-root 法は中国の剰余定理によって効率化されることが知られている[1][33]。

いま $d \mid p-1$ に対し $a_d = a^{(p-1)/d}$, $b_d = a^{(p-1)/d}$ とすると、

$$a_d \equiv b_d^{x_d} \bmod p$$

を満足する x_d に対し

$$x \equiv x_d \bmod \frac{p-1}{d}$$

が成立する。適切に選択した十分な数の d に対し square-root 法によって x_d を求めれば、中国人の剰余定理（と、場合によっては Newton 反復）によって x を求めることが可能である。この方法の詳細については、例えば [37, Section 11.2], [25, Section 3.6.4] を参照されたい。

この方法により離散対数問題に対する square-root 法の計算量は $O(\sqrt{l})$ となる。ここで l は $p-1$ を割る最大素因数を表す。

3.2 指数計算法

離散対数問題に対して一般に square-root 法より効率的な「指数計算法」と呼ばれるアルゴリズムが知られている [1]。Algorithm 2 にこの指数計算法を示す。

[1] この手法を全数探索とともに用いることも可能であるが、通常は square-root 法とともに用いる。

Algorithm 2 指数計算法

Input: p: 素数, $a, b \in \mathbb{F}_p^*$ s.t. $\langle b \rangle = \mathbb{F}_p^*$, $s \in \mathbb{N}$ s.t. $s < p$
Output: $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ s.t. $a = b^x$

1: $B := \{l_j \in \mathbb{Z} \mid l_j : \text{prime number} \leq s\}, n := \#B$
`/*`因子基底 `(factor base)*/`
2: $i := 1$
3: **repeat** `/*STAGE 1:` 対数表の作成`*/`
4: Choose $r_i \in \mathbb{Z}/(p-1)\mathbb{Z}$ randomly
5: **if** $(b^{r_i} \bmod p) = \prod_{j=1}^{n} l_j^{e_j} \in \mathbb{Z}$ **then** `/*via trial division*/`
6: $e_{ij} := e_j$ for $j = 1 \ldots n$
7: $i = i + 1$
8: **until** $\mathrm{rank}(e_{ij}) = n$ (over $\mathbb{Z}/(p-1)\mathbb{Z}$)
9: Compute $\mathrm{Ind}_b l_i$ for $i = 1 \ldots n$
10: **repeat** `/*STAGE 2:` $\mathrm{Ind}_b a$ の求解`*/`
11: Choose $r \in \mathbb{Z}/(p-1)\mathbb{Z}$ randomly
12: **until** $(ab^r \bmod p) = \prod_{j=1}^{n} l_j^{f_j} \in \mathbb{Z}$ `/*via trial division*/`
13: Output $\sum_{j=1}^{n} f_j \mathrm{Ind}_b l_j - r \bmod p - 1$ as x and terminate

指数計算法は、まず小さな素数からなる因子基底と呼ばれる集合を決め、次にこの因子基底の要素に対する対数表を作成する。そして、この対数表を用いて与えられた離散対数問題を解くものである。以下にこのアルゴリズムの実例を示す。

例 3.2 例 *3.1* と同様に、与えられた $p = 47$, $a = 40$, $b = 11$ に対し $a \equiv b^x \bmod p$ を満足する x を求める。まず、因子基底として $B = \{2, 3, 5, 7, 11, 13\}$ を選ぶ。そして、*Algorithm 2* のステップ *4* に従い $r_1 \in \mathbb{Z}/(p-1)\mathbb{Z}$ をランダムに選択する。例えば $r_1 = 9$ を選択すると $b^{r_1} \equiv 38 = 2 \cdot 19 \bmod p$ が得られる。しかし、この場合はステップ *5* に示された因子基底の要素による関係が（簡単には）得られないのでこれを破棄し、新たに r_1 を選択し直す。幾度かの試行の後に $r_1 = 42$ を選択すると $b^{r_1} \equiv 2 = \bmod p$ が得られ、因子基底の要素による関係式が得られたこととなる。同様の試行を n 個の関係式が得られる間で繰り返すと、例えば

$$\begin{pmatrix} 11^{42} \\ 11^{3} \\ 11^{29} \\ 11^{11} \\ 11^{31} \\ 11^{1} \end{pmatrix} = \begin{pmatrix} 2 \\ 15 \\ 10 \\ 39 \\ 35 \\ 11 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \cdot 5 \\ 2 \cdot 5 \\ 3 \cdot 13 \\ 5 \cdot 7 \\ 11 \end{pmatrix} = \begin{pmatrix} 11^{\mathrm{Ind}_{11} 2} \\ 11^{\mathrm{Ind}_{11} 3} \cdot 11^{\mathrm{Ind}_{11} 5} \\ 11^{\mathrm{Ind}_{11} 2} \cdot 11^{\mathrm{Ind}_{11} 5} \\ 11^{\mathrm{Ind}_{11} 3} \cdot 11^{\mathrm{Ind}_{11} 13} \\ 11^{\mathrm{Ind}_{11} 5} \cdot 11^{\mathrm{Ind}_{11} 7} \\ 11^{\mathrm{Ind}_{11} 11} \end{pmatrix}$$

が得られる。この式は指数に関し線形方程式系

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

を満足するので、これを解き小さな素数に対する「対数表」

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

が得られる。次に *Algorithm 2* のステップ *10* 以降を実行し、対数表を用いて与えられた問題を解く。まず、*Algorithm 2* のステップ *4* に従い $r \in \mathbb{Z}/(p-1)\mathbb{Z}$ をランダムに選択する。そして、選択した r に対し ab^r を計算し、対数表の作成と同様にこれが因子基底の要素に分解されるまで繰り返す。例えば、$r = 33$ を選択すると $ab^5 \equiv 40 \cdot 11^{33} \equiv 12 \equiv 2^2 \cdot 3 \bmod p$ が得られ、これと対数表から

$$\mathrm{Ind}_{11}40 \equiv 2\mathrm{Ind}_{11}2 + \mathrm{Ind}_{11}3 - 33 \equiv 21 \bmod p-1$$

を得る。

注意 3.2 *Algorithm 2* は標準的なものだが、実際には対数表を作成しない変形を用いることも多い。この方法ついては例えば *[37, Chapter 16]* を参照されたい。

3.3 指数計算法の計算量

Algorithm 2 の計算量は明らかに s の選択に依存する。すなわち、s を小さくとると「関係式」を得られる確率が低くなり、逆に s を大きくとると、より多くの「関係式」集める必要が生じ、さらに行列の次数が高くなるので線形代数計算のコストもより大きくなる。s の最良の設定は素数分布等の知見から得られ、Algorithm 2 の漸近計算量は $O(L_p(1/2, 2+o(1)))$ となる。ここで

$$L_n(\alpha, \beta) := \exp\left(\beta(\log n)^{\alpha}(\log\log n)^{1-\alpha}\right)$$

である。この記法を用いると rho 法の計算量は $O(L_p(1, 1/2))$ となり、指数計算法は rho 法と比較し著しく高速であることが判る。さらに指数計算法に対し多くの改良が行われており、上式において $\alpha = 1/3$ のアルゴリズムが知られている。このような $0 < \alpha < 1$ のアルゴリズムは準指数時間アルゴリズムと呼ばれる[2]。Algorithm 2 の計算量評価については [37, Chapter 16] を、指数計算法とその改良についてはこれの他に [10, Section 6.4], [25, Section 3.6.5] とこれらに挙げられている文献を参照されたい。

以上のように離散対数問題に対しては全数探索や square-root 法と比較し著しく効率的なアルゴリズムが知られているため、解読に 2^{80} 程度の手間を必要とする暗号を離散対数を利用して構成する場合には 1024 ビット程度の p を選択する必要があると考えられている。（より大きな p が必要であるとの推測もある。）

[2] $\alpha = 1$ が指数時間アルゴリズム、$\alpha = 0$ が多項式時間アルゴリズムである。

4 離散対数問題の一般化と楕円曲線暗号

以上で見たように、定義 2.1 で与えた離散対数問題には準指数時間計算量アルゴリズムが知られており、近年のコンピュータ能力の指数関数的な進歩が、これを利用した暗号にとって両刃の剣となった。すなわち、計算速度の急激な向上によって暗号解読時間もまた急激に短くなり、安全性確保のために問題のサイズを準指数関数的に増加させる必要が生じ、結果として暗号化速度の面でコンピュータ性能の進歩を完全には享受できないこととなった。そこで、このようなコンピュータ性能の進歩による性能の劣化が生じない暗号が要求されることとなった。

定義 2.1 で与えた離散対数問題は以下に示すように一般の有限可換群 G 上の問題として一般化可能である。

定義 4.1 (離散対数問題) 与えられた有限可換群 G, $b \in G$, $a \in \langle b \rangle$ に対し $a = [x]b$ を満足する $x \in \mathbb{Z}/\#G\mathbb{Z}$ を求める問題を G 上の離散対数問題という。また、この x を $\mathrm{Ind}_b a$ と書く。

3.1 節で示した square-root 法は一般の G 上の離散対数問題に適用可能であるが、一方 Algorithm 2 に示した指数計算法は一般の G 上の離散対数問題に適用可能なアルゴリズムではない。従って、square-root 法以上に効率的なアルゴリズムが存在しない問題を設定できれば、それを用いた暗号は上述の課題を解決できることとなる。このような G として有限体上の楕円曲線の有理点群が知られている。これを用いた楕円曲線暗号は、Miller [27] と Koblitz [20] によって独立に提案された。特に [27] は指数計算法が楕円曲線上の離散対数問題 (ECDLP) に対して有効に働かないことを主張している。実際にその後も ECDLP に対する指数計算法的なアルゴリズムの研究が盛んに行われているが、現在まで一般的且つ効率的なアルゴリズムは得られていない。一方 [20] は中国の剰余定理を利用した square-root 法に対する耐性が高い曲線の構成法を主旨とした論文となっている。即ち、楕円曲線暗号では固定された定義体の上においても曲線を選択により $\#G$ が変化するので、適切な選択により（中国の剰余定理が無意味となる）素数若しくは素数に近い $\#G$ に設定可能である。与えられた曲線の群位数が計算が可能であればそのような位数を持つ曲線を選択可能であり、長い間、曲線の位数計算アルゴリズムは楕円曲線暗号研究の主要課題の一つであった。多くの研究の結果として現状では暗号に利用するサイズの位数を実用的な時間で計算することが可能となっている。

位数計算を含め楕円曲線暗号全般については近年多数の良書が出版されているのでそれらを参照されたい [24, 22, 23, 5, 41, 6, 8]。また、特殊な楕円曲線上の離散対数問題に対する効率的なアルゴリズムがいくつか知られているので、これらについてもここに挙げた文献を参照されたい。

現在では多くの製品に楕円曲線暗号が利用されている。これは同一の安全性を仮定した場合、有限体上の離散対数問題に基づく暗号や事実上の標準暗号である RSA 暗号と比較してより高速な暗号を実現可能となったためである[3]。また、これには楕円曲線の持つ豊富な性質を利用した高速化手法に関する研究の寄与も大きい。現在に至るまで高速化に関する研究は盛んに行われ続けている。高速化の実際に関しては例えば [42] 等を参照されたい。

[3]講演ではこれについても触れたが、本稿ではこれ以上触れない。講演資料 [43] を参照されたい。

5 超楕円曲線暗号

楕円曲線暗号は強力だが暗号アルゴリズムは突然その価値を失うことが少なくない。そこで、新たな暗号アルゴリズムの探求が常に行われている。この観点から Koblitz [21] によって楕円曲線暗号の自然な一般化として超楕円曲線暗号が提案された。

以下では種数 g の超楕円曲線 C が

$$\begin{aligned} C: \quad & Y^2 = F(X), \\ & F(X) = X^{2g+1} + f_{2g}X^{2g} + \cdots + f_0 \in \mathbb{F}_p[X] \end{aligned} \tag{1}$$

と定義されているとする。また、C の Jacobian を $\mathcal{J}_C$ と書く。超楕円曲線暗号は定義 4.1 において $G = \mathcal{J}_C(\mathbb{F}_p)$ としたものである。効率的な暗号を構成するためには G 上の効率的な加算アルゴリズムが必要であるが、Koblitz はこれに Cantor [7] が陽に示したアルゴリズムを用いた。この加算アルゴリズムを暗号界では「Cantor アルゴリズム」と呼ぶ。Cantor アルゴリズムは $\mathcal{J}_C(\mathbb{F}_p)$ の要素表現に [29] に記述のある多項式の組による表現を用いている。この表現を最近の暗号界では「Mumford 表現」と呼ぶ。

定義 5.1 (Mumford 表現) 式 (1) で与えられた C に対し、*1.* $\mathrm{lc}(U) = 1$, *2.* $\deg V < \deg U$, *3.* $U \mid F - V^2$ を満足する多項式の組 $(U, V) \in (\overline{\mathbb{F}}_p[X])^2$ を $\mathcal{J}_C(\overline{\mathbb{F}}_p)$ の元の *Mumford* 表現と呼ぶ。

与えられた $\mathcal{D} = \sum_{1 \le i \le n} P_i - nP_\infty \in \mathcal{J}_C(\overline{\mathbb{F}}_p)$ $(P_i \in C(\overline{\mathbb{F}}_p))$ の Mumford 表現 (U, V) は

$$U = \prod_{1 \le i \le n} (X - X(P_i)),\ Y(P_i) = V(X(P_i))$$

と上記 3 条件から得られる。$\mathcal{J}_C(\overline{\mathbb{F}}_p)$ の任意の元は $\deg U \le g$ を満足する Mumford 表現で一意表現されることが知られている。さらに、

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

と看做すことが可能である。この性質によって（手間のかかる）拡大体上の演算が不要となるため Mumford 表現は暗号実装に向いた表現であるといえる。Cantor アルゴリズムと Mumford 表現の詳細については本報告集の志村氏の報告や [26] を参照されたい。また、偶数次の超楕円曲線に対する Mumford 表現と Cantor アルゴリズムが [32] に示されていることを付記する[4]。

$\#\mathcal{J}_C(\mathbb{F}_p) \approx p^g$ より、(square-root 法より効率的な解読アルゴリズムが存在しないとの仮定の下で) 同一の安全性を持つ楕円曲線暗号と比較し、より小さい定義体上で超楕円曲線暗号を構成可能であり、プラットフォームによってはより効率的な実装が可能となる。このような利点を有するので、例えば [44] 等 $\mathcal{J}_C$ 上の加算アルゴリズムの研究が現在に至るまで盛んに行われている。実際、Cantor アルゴリズムを用いた超楕円曲線暗号は同一の安全性を有する楕円曲線暗号と比較し数倍低速であることが知られていたが、多くの研究の

[4] これまでの処偶数次の超楕円曲線を（攻撃以外に）暗号利用する利点は見出されていない

結果、楕円曲線暗号とほぼ同一の速度を達成可能なアルゴリズムが知られるようになった。この種のアルゴリズムは「Harley アルゴリズム」と呼ばれる。Harley アルゴリズムについては [8, Chapter 14] とそこに挙げられている文献等を参照されたい。また、[3], [8, Section 14.7] 等、より一般の曲線の Jacobian 上の加算アルゴリズムの研究も盛んに行われていることを付記する。

6 超楕円曲線上の離散対数問題に対する指数計算法

超楕円曲線暗号が提案されて暫くの後、[2] が超楕円曲線上の離散対数問題に対する準指数時間アルゴリズムを示した。このアルゴリズムは、s より小さい素数に代えて次数が s より小さい多項式を因子基底とし、Mumford 表現に現れる多項式 U が因子基底の要素に分解される場合に対して関係式を得るものである。このアルゴリズムの計算量は $\log p < (2g+1)^{0.98}$, $g \to \infty$ に対し $O(L_{p^{2g+1}}(1/2, c < 2.181))$ である。またこのアルゴリズムの改良が研究され、計算量が $p^g \to \infty$ に対し $O(L_{p^g}(1/2, \cdot)$ のアルゴリズムが得られている [14]。これらのアルゴリズムの出現により種数が 2 桁以上の曲線を暗号に利用することは難しくなった。しかし、これらは超楕円曲線暗号にとっての脅威とは考えられてこなかった。何故ならば、(暗号応用に対して適切な設定である) g を固定した場合には、これらはいずれも指数時間計算量のアルゴリズムであり、実際に効果が現れる種数が暗号に利用される曲線の種数より大きいと考えられたからである。しかし、Gaudry [16] によって上記アルゴリズム低種数曲線に対する変形が示された。この Gaudry アルゴリズムは、指数時間計算量アルゴリズムであるものの、ある範囲の種数の超楕円曲線上の離散対数問題に対する計算量が square-root 法より小さいアルゴリズムであり、超楕円曲線の暗号応用に対し現実的な脅威となりうるものである。

Algorithm 3 に Gaudry アルゴリズムを示す。Algorithm 3 に示したアルゴリズムは、Algorithm 2 との対応を見やすくするために、[16] に示されたアルゴリズムに修正を施したものであることに注意されたい。Algorithm 3 を Algorithm 2 と比較すると Gaudry アルゴリズムが因子基底の選択を除き有限体上の離散対数問題に対する指数計算法とほぼ同一のアルゴリズムであることが理解される。以下に Algorithm 3 による離散対数問題解法の具体例を示す。

例 6.1 与えられた $p = 47$, $a = 40$, $b = 11$ に対し $a \equiv b^x \bmod p$ を満足する x を求める。

$p = 7$ とし、$\mathbb{F}_p$ 上の種数 *6* の超楕円曲線

$$(2)\quad C/\mathbb{F}_p : Y^2 = X^{13} + 5X^{12} + 4X^{11} + 6X^9 + 2X^8 + 6X^7 + 5X^4 + 5X^3 + X^2 + 2X + 6$$

を選ぶ。ここで $N := \#\mathcal{J}_C(\mathbb{F}_p) = 208697$ であり、これは素数である。以下では

$$\begin{aligned} \mathcal{D}_a &= (X^6 + 2X^5 + 4X^4 + X^3 + 5X^2 + 3, 4X^5 + 5X^3 + 2X^2 + 5X + 4), \\ \mathcal{D}_b &= (X^5 + 6X^3 + 3X^2 + 1, 3X^4 + X^3 + 4X^2 + X + 3) \in \mathcal{J}_C(\mathbb{F}_p) \end{aligned}$$

に対し、$\mathcal{D}_a = [\mathrm{Ind}_{\mathcal{D}_b}\mathcal{D}_a]\mathcal{D}_b$ を満足する $\mathrm{Ind}_{\mathcal{D}_b}\mathcal{D}_a$ を求める。

Algorithm 3 Gaudry アルゴリズム

Input: $C/\mathbb{F}_p$: 超楕円曲線, $\mathcal{D}_a, \mathcal{D}_b \in \mathcal{J}_C(\mathbb{F}_p)$ s.t. $\mathcal{D}_a \in \langle \mathcal{D}_b \rangle$, $N := \#Jc(\mathbb{F}_p)$
Output: $x \in \mathbb{Z}/N\mathbb{Z}$ s.t. $\mathcal{D}_a = [x]\mathcal{D}_b$

1: $B := \{P_j \in C(\mathbb{F}_p) \setminus P_\infty \mid X(P_j) \neq X(P_i) \text{ for } i \neq j\}, n := \#B$ /*因子基底*/
2: $i := 1$
3: **repeat** /*STAGE 1: 対数表の作成*/
4: Choose $r_i \in \mathbb{Z}/N\mathbb{Z}$ randomly
5: **if** $[r_i]\mathcal{D}_b = \sum_{j=1}^{n} e_j P_j^{e_j} - mP_\infty$ **then** /*via factorization of U*/
6: $e_{ij} := e_j$ for $j = 1 \ldots n$
7: $i = i + 1$
8: **until** $\mathrm{rank}(e_{ij}) = n$ (over $\mathbb{Z}/N\mathbb{Z}$)
9: Compute $\mathrm{Ind}_{\mathcal{D}_b} P_i$ for $i = 1 \ldots n$
10: **repeat** /*STAGE 2: $\mathrm{Ind}_{\mathcal{D}_b}\mathcal{D}_a$ の求解*/
11: Choose $r \in \mathbb{Z}/N\mathbb{Z}$ randomly
12: **until** $\mathcal{D}_a + [r]\mathcal{D}_b = \prod_{j=1}^{n} [s_j]P_j - mP_\infty \in \mathbb{Z}$ /*via factorization of U*/
13: Output $\sum_{j=1}^{n} s_j \mathrm{Ind}_{\mathcal{D}_b} P_j - r \bmod N$ as x and terminate

まず、$C(\mathbb{F}_p) = \{P_\infty, (1,1), (1,6), (2,1), (2,6), (4,1), (4,6)(5,3), (5,4), (6,3), (6,4)\}$ から因子基底

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

を選択する。そして、*Algorithm 3* のステップ *4* に従い $r_1 \in \mathbb{Z}/N\mathbb{Z}$ をランダムに選択する。例えば $r_1 = 9343$ を選択すると、ステップ *5* に現れる $[r_1]\mathcal{D}_b$ は

$$[9343]\mathcal{D}_b = (X^5 + 6X^4 + 6X^3 + 5X^2 + 6X + 4, X^4 + X^3 + X^2 + 4X + 6)$$

となる。これの第一多項式は $\mathbb{F}_p$ 上で $X^5+6X^4+6X^3+5X^2+6X+4 = (X-1)^2(X-4)^2(X-5)$ と *1* 次式に素因子分解される。従ってこの r_1 からは関係式が得られる。実際、$X^4 + X^3 + X^2 + 4X + 6 \mid_{X=1} = 6$, $X^4 + X^3 + X^2 + 4X + 6 \mid_{X=4} = 1$, $X^4 + X^3 + X^2 + 4X + 6 \mid_{X=5} = 3$ より、

$$[9343]\mathcal{D}_b = -[2](1,1) + [2](4,1) + (5,3)$$

が得られる。これを繰り返すことで、線形方程式系

$$\begin{pmatrix} [9343]\mathcal{D}_b \\ [120243]\mathcal{D}_b \\ [121571]\mathcal{D}_b \\ [120688]\mathcal{D}_b \\ [151649]\mathcal{D}_b \end{pmatrix} = \begin{pmatrix} -2 & 0 & 2 & 1 & 0 \\ 0 & -2 & 1 & 1 & -2 \\ -1 & 0 & 2 & -1 & -1 \\ 2 & 1 & 0 & 2 & 0 \\ 1 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} (1,1) \\ (2,1) \\ (4,1) \\ (5,3) \\ (6,3) \end{pmatrix}$$

が得られる。この方程式系を解き対数表

$$\begin{pmatrix} \mathrm{Ind}_{\mathcal{D}_b}(1,1) & \mathrm{Ind}_{\mathcal{D}_b}(2,1) & \mathrm{Ind}_{\mathcal{D}_b}(4,1) & \mathrm{Ind}_{\mathcal{D}_b}(5,3) & \mathrm{Ind}_{\mathcal{D}_b}(6,3) \end{pmatrix}$$
$$\equiv \begin{pmatrix} 85159 & 114347 & 182999 & 22360 & 136908 \end{pmatrix} \bmod N$$

を得る。

従って、

$$\mathcal{D}_a + [105454]\mathcal{D}_b = (1,1) + [2](2,1) + (4,1) - (6,3)$$

であり、

$$\mathrm{Ind}_{\mathcal{D}_b}\mathcal{D}_a \equiv \mathrm{Ind}_{\mathcal{D}_b}(1,1) + 2\mathrm{Ind}_{\mathcal{D}_b}(2,1) + \mathrm{Ind}_{\mathcal{D}_b}(4,1) - \mathrm{Ind}_{\mathcal{D}_b}(6,3) - 105454 \equiv 45793 \bmod N$$

を得る。

以下では Algorithm 3 の計算量を評価する。評価を必要とするのは $\#B + 1 = O(p)$ 個の関係式を得るために必要な計算量と線形方程式系を解くために必要な計算量である。

Algorithm 3 のステップ 5, 12 は因子類の整数倍算と g 次多項式の素因子分解を必要する。因子類の加算は $O(g^2(\log p)^2)$ ビット演算を必要とするので整数倍算に必要な計算量は $O(g^2(\log p)^3)$ である。また、g 次多項式の素因子分解に必要な計算量は $O(g^3(\log p)^3)$ ビット演算である[1]。従って、これらのステップの実行には $O(g^3(\log p)^3)$ ビット演算を必要とする[1]。$\mathbb{F}_p$ 上のモニック g 次多項式の数は p^g であり、1 次式の積に分解するモニック g 次多項式の数は $p^g/g!$ であるので、$O(p)$ 個の関係式を得るために必要な計算量は $O(g!g^3p(\log p)^3)$ となる。一方、対数表の作成過程で得られる行列は各行に高々 g 個の要素を持つ疎行列である。疎行列に対しては効率的なアルゴリズムが知られており [37, Section 19.4]、これを用いることでステップ 9 に必要な計算量は $O(gp^2(\log N)^2) = O(g^3p^2(\log p)^2)$ となる。以上より、Algorithm 3 の計算量は

$$O\left(g!g^3p(\log p)^3 + g^3p^2(\log p)^2\right) \tag{3}$$

ビット演算であり、種数 g が十分に小さい範囲で $g \geq 5$ に対し rho 法より漸近的に計算量が小さい。これはあくまでも「漸近的」な振舞について述べたものであり、暗号に利用される範囲のサイズの p に対して実際に効果があるとはいえないことに注意されたい。

7 Gaudry アルゴリズムの改良

[16] は Algorithm 3 の計算量削減手法についても言及している。この手法は (3) の 2 項を（p に関して）バランスをとり全体の計算量を削減するものである。実際、因子基底を B の代わりに $\#B_0 = O(p^r)$, $0 < r < 1$ を満足する $B_0 \subset B$ とすれば、Algorithm 3 の計算量は、（g や $\log p$ を無視して）

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

となる。従って、$r = g/(g+1)$ とすれば、その計算量が $\tilde{O}(p^{2g/(g+1)})$ となり、種数 $g \geq 4$ に対し rho 法より漸近的に計算量が小さいアルゴリズムが得られる。

また、Thériault[40] によってこのアルゴリズムのさらなる改良が行われた。この改良は素因数分解の標準的な高速化手法として知られる larg prime 手法を Gaudry アルゴリズム

[1]暗号応用考慮し、標準的な乗算アルゴリズムを利用することを仮定している。

に応用したものである。これは、B_0 の要素による関係式が得られる確率が $O(p^{g(r-1)})$ であるのに対して、1 個だけ $B \setminus B_0$ の要素[5]を含み他は B_0 の要素による関係式が得られる確率が $O(p^{(g-1)(r-1)})$ と高確率であることを利用している。実際、Algorithm 3 のステップ 3 のループを p^s 回繰り返した後には B_0 の要素による関係式が $O(p^{sg(r-1)})$ 個得られるが、さらに 1 個だけ $B \setminus B_0$ の要素を含み他は B_0 の要素による関係式が $O(p^{s(g-1)(r-1)})$ 個得られると期待される。これらの関係式の中には $B \setminus B_0$ の同一要素を含む組が $O(p^{2s(g-1)(r-1)-1})$ 組あると期待されるので、それぞれの組から $B \setminus B_0$ の要素を消去することで B_0 の要素のみによる新たな関係式が得られる。そしてその数は元々得られていた関係式の数より多い。そこで、r と s を最適に選択することで Gaudry アルゴリズムの計算量が削減される。

さらに、Nagao [30] と Gaudry, Thomḿe, Thériault, Diem [19] によって独立に 2 個の large prime を利用する改良が示された。このアルゴリズムの計算量は

$$\tilde{O}\left(q^{2-\frac{2}{g}}\right)$$

であり、これが低種数の超楕円曲線上の離散対数問題に対する現在までの最良計算量アルゴリズムである。この計算量から種数 g が十分に小さい範囲で $g \geq 3$ に対し rho 法より漸近的に計算量が小さいアルゴリズムであることがわかる。

このように種数が 3 以上の代数曲線上の離散対数問題に対しては漸近計算量が rho 法より小さいアルゴリズムが存在するため、これらを暗号利用する際にはその安全性に際し詳細な議論を必要とするようになった。特に種数が大きい代数曲線については暗号速度を維持しつつ安全性を確保することが困難であり、事実上暗号利用は不可能である。

注意 7.1 上記のような改良は素因数分解では漸近計算量削減手法ではなく高速化手法であるのに対し、超楕円曲線上の離散対数問題に対しては漸近計算量削減手法として働く。更に *large prime* を *3* 個以上含む関係式を利用しても漸近計算量は削減されない。

注意 7.2 ここで紹介したアルゴリズムは超楕円曲線以外の代数曲線上の離散対数問題に対しても適用可能である。更に、最近になって *Diem [11]* によって同一種数の超楕円曲線と比較して次数が小さい平面代数曲線に対するより効率的なアルゴリズムが示された。このアルゴリズムの計算量は次数 $d \geq 4$ の平面代数曲線に対して

$$\tilde{O}\left(p^{2-\frac{2}{d-2}}\right)$$

である。このアルゴリズムの出現により、超楕円曲線以外の高種数代数曲線を暗号に利用することは困難になった。

8 楕円曲線上の離散対数問題への応用

前節で紹介した攻撃法を拡大体上定義された楕円曲線（や超楕円曲線等）の上の離散対数問題に適用可能な場合があることが Frey, Gangle [15] によって指摘された。これは、楕

[5] このような要素を large prime という。

円曲線の $\mathbb{F}_{p^k}$ 有理点群 $E(\mathbb{F}_{p^k})$ を種数 $g \geq k$ の代数数曲線 C の Jacobian の有理点群 $\mathcal{J}_C(\mathbb{F}_p)$ に埋め込み、$\mathcal{J}_C(\mathbb{F}_p)$ 上で前節のアルゴリズムによって離散対数問題を解くものである。この攻撃は「Weil descent 攻撃」と呼ばれる。その後、Gaudry, Hess, Smart [18] によって、この攻撃の陽な（GHS-Weil descent と呼ばれる）アルゴリズムが示され、これについて多くの研究がなされてきた。これらについては [6, Chapter VIII], [8, Section 22.3] 等を参照されたい。この攻撃が効率的であるためには C の種数 g が（$g = k$ を満足する等）k に十分に近い必要があり、どの程度の数の曲線に対し効果があるか等研究課題が多い。このような状況において、例えば、$\mathbb{F}_{p^4}$ 上の（暗号に適した）楕円曲線の全てに対し Weil descent 攻撃の計算量が漸近的に rho 法より小さいこと等が示され始めている [4]。さらに、高種数代数曲線 C を介さない方法が最近提案された [17], [31]。これは、因子基底に $\mathcal{J}_C(\mathbb{F}_p)$ を用いる代わりに

$$B = \left\{ P \in E(\overline{\mathbb{F}}_p) | X(P) \in \mathbb{F}_p \right\}$$

を用いて（従って、一般には $B \not\subset E(\mathbb{F}_{p^k})$）、「関係式」を得るために 1 変数多項式の素因子分解を行う代わりに多変数代数方程式系の求解を行うものである。

暗号に利用されるサイズの離散対数問題に対してこれらのアルゴリズムがどの程度の効果を持つのかについては、現在迄殆ど知見がない状態である。

9 おわりに

本稿では触れることができなかったが、（楕円曲線暗号を含む）代数曲線暗号に関する最近の研究成果の多くが「ペアリング暗号」に関するものであることを付記する。この分野は、[45] や [28] 等、日本人の研究結果を端緒としている。また、多くの（計算）数論的な問題を残している分野である。この分野の研究状況については [6, Part 4] や [41, 8] の他に本スクールの直前に東京で開催された国際会議の proceedings [38] 等を参照されるとよいだろう。

参考文献

[1] L. Adleman. A subexponential algorithm for the discrete logarithm problem with applications. In *Proc. 20th Ann. IEEE Symp. on Foundations of Computer Science*, pp. 55–60, 1979.

[2] L. Adleman, J. DeMarrais, and M. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobian of large genus hyperelliptic curves over finite fields. In *ANTS-I*, LNCS 877, pp. 28–40. Springer, 1994.

[3] S. Arita. An addition algorithm in Jacobian of C_{34} curve. In *Information Security and Privacy, ACISP 2003*, LNCS 2727, pp. 248–258. Springer, 2003.

[4] S. Arita, K. Matsuo, K. Nagao, and M. Shimura. A Weil descent attack against elliptic curve cryptosystems over quartic extension fields. *IEICE Trans.*, Vol. E89-A, No. 5, May 2006. 1246-1254.

[5] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography.* LMS 265. Cambridge U. P., 1999.

[6] I. Blake, G. Seroussi, and N. Smart, editors. *Advances in Elliptic Curves Cryptography.* LMS 317. Cambridge U. P., 2005.

[7] D. G. Cantor. Computing in the Jacobian of hyperelliptic curve. *Math. Comp.*, Vol. 48, No. 177, pp. 95–101, 1987.

[8] H. Cohen, G. Frey, C. Doche, editors. *Handbook of elliptic and hyperelliptic curve cryptography.* Chapman & Hall/CRC, 2005.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 2nd edition, 2001.

[10] R. Crandall and C. Pomerance. *Prime Numbers.* Springer, 2nd edition, 2005.

[11] C. Diem. An index calculus algorithm for plane curves of small degree. In *ANTS-VII*, LNCS 4076, pp. 543–557. Springer, 2006.

[12] W. Diffie and M. Hellman. New direction in cryptography. *IEEE Trans.*, Vol. IT-23, No. 6, pp. 644–654, 1976.

[13] T. ElGamal. A public key cryptosystem and sigunature scheme based on discrete logarithms. *IEEE Trans.*, Vol. IT-31, No. 4, pp. 469–472, 1985.

[14] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, Vol. 102, pp. 83–103, 2002.

[15] G. Frey and H. Gangl. How to disguise an elliptic curve (Weil descent). Talk at ECC '98, The 2nd Workshop on Elliptic Curve Cryptography, U. Waterloo, `http://www.cacr.math.uwaterloo.ca/conferences/1998/ecc98/frey.ps`, 1998.

[16] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, pp. 19–34. Springer, 2000.

[17] P. Gaudry. Index calculus for abelian varieties and the elliptic curve discrete logarithm problem. Cryptology ePrint Archive, Report 2004/073, 2004. `http://eprint.iacr.org/`.

[18] P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, Vol. 15, No. 1, pp. 19–46, 2002.

[19] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, Vol. 76, No. 257, pp. 475–492, 2007.

[20] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, Vol. 48, pp. 203–209, 1987.

[21] N. Koblitz. Hyperelliptic curve cryptosystems. *J. Cryptology*, Vol. 1, No. 3, pp. 139–150, 1989.

[22] N. Koblitz. *A course in number theory and cryptography.* GTM 114. Springer, 2nd edition, 1994.

[23] N. Koblitz. *Algebraic Aspects of Cryptography*, Vol. 3 of *Algorithms and Computation in Mathematics.* Springer, 1998.

[24] A. Menezes. *Elliptic Curve Public Key Cryptosystems.* Kluwer Academic Pub., 1993.

[25] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of applied cryptography.* CRC Press, 1997.

[26] A. Menezes, Y. Wu, and J. Zuccherato. An elementary introduction to hyperelliptic curves. Appendix to [23], 1998.

[27] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology - CRYPTO '85*, LNCS 218, pp. 417–426. Springer, 1986.

[28] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans.*, Vol. E84-A, No. 5, pp. 1234–1243, 2001.

[29] D. Mumford. *Tata Lectures on Theta II.* PM 43. Birkhäuser, 1984.

[30] K. Nagao. Index calculus attack for Jacobian of hyperelliptic curves of small genus using two large primes. *Japan J. of Industrial and Applied Math.*, Vol. 24, No. 3, 2007.

[31] K. Nagao. On the decomposition of an element of jacobian of a hyperelliptic curve. Cryptology ePrint Archive, Report 2007/112, 2007. `http://eprint.iacr.org/`.

[32] S. Paulus and A. Stein. Comparing real and imaginary arithmetics for divisor class groups of hyperelliptic curves. In *ANTS-III*, LNCS 1423, pp. 576–591. Springer, 1998.

[33] G. C. Pohlig and M. E. Hellman. An improved algorithm for comuting logarithms over GF(p) and its cryptographic significance. *IEEE Trans. on Info. Theory*, Vol. IT-24, pp. 106–110, 1978.

[34] J. M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, Vol. 32, pp. 918–924, 1978.

[35] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Com. of the ACM*, Vol. 21, No. 2, pp. 120–126, 1978.

[36] D. Shanks. Class number, a theory of factorization, and genera. In *Proc. of Symp. Math. Soc. 20*, pp. 415–440, 1971.

[37] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge U. P., 2005.

[38] T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto, editors. *Pairing-based cryptography - Pairing 2007*. LNCS 4575. Springer, 2007.

[39] E. Teske. Square-root algorithms for the discrete logarithm problem (A survey). In *Public-Key Cryptography and Computational Number Theory*, pp. 283–301. Walter de Gruyter, 2001.

[40] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology - ASIACRYPT 2003*, LNCS 2894, pp. 75–92. Springer, 2003.

[41] L. Washington. *Elliptic curves: number theory and cryptography*. Chapman & Hall/CRC, 2003.

[42] 青木. 楕円曲線暗号はどこまで速くなるか? — ソフトウェア実装の到達点. 仙台数論小研究集会 2000, 東北大, 2000. `http://staff.miyakyo-u.ac.jp/~taya/sendaiNT/2000/aoki_m.pdf`.

[43] 松尾. 代数曲線暗号とその安全性. 2007 年度 整数論サマースクール講演資料, `http://http://lab.iisec.ac.jp/~matsuo_lab/pub/pdf/sss_mats.pdf`, 2007.

[44] 入海, 松尾, 趙, 辻井. 超楕円曲線上の Harley アルゴリズムにおける resultant 計算について. Technical Report ISEC2006-5, 電子情報通信学会, 2006.

[45] 大岸, 境, 笠原. 楕円曲線上の ID 鍵共有方式の基礎的考察. Technical Report ISEC99-57, 電子情報通信学会, 1999.