楕円曲線上の離散対数問題に関する指数計算法

篠原 直行 [†]　野呂 正行 [‡]　横山 和弘 [‡]

[†]NICT　[‡] 立教大学

概 要

楕円曲線暗号は現在, 実際に使用されている代表的な公開鍵暗号方式である. また, 実用化が進められている公開鍵暗号方式としてペアリング暗号が挙げられ, ペアリング暗号を基盤技術として様々な高度な暗号技術を利用できることが知られている. これらの公開鍵暗号方式は有限体上の楕円曲線を利用しており, その楕円曲線において与えられる離散対数問題 (ECDLP) が解かれると解読されてしまう.

ECDLP だけではなく一般的に, 有限巡回群上の離散対数問題 (DLP) を解くアルゴリズムとして指数計算法がある. 例えば, 有限体上の DLP を効率良く解く指数計算法として数体篩法や関数体篩法などが挙げられる. 近年, summation polynomial やグレブナー基底などを利用して, ECDLP に対して有効な指数計算法を構築する研究が進められている.

本稿では, 楕円曲線暗号やペアリング暗号に対するこれらの新たな攻撃方法を紹介し, それらの影響について述べる. また, 現時点ではこれらの新たな攻撃方法より, Pollard の ρ 法等の既存の攻撃方法の方が計算効率が良いと結論づける. しかし, ECDLP に関する指数計算法の研究の動向は注視する必要がある.

1　はじめに

楕円曲線暗号は現時点で広く使用されている代表的な公開鍵暗号方式の一つであり, 楕円曲線暗号で利用する楕円曲線において与えられる離散対数問題 (ECDLP) が解かれると解読されてしまう. また, 高度な暗号技術を実現するための基盤技術として, ペアリング暗号とよばれる公開鍵暗号方式があり, その実用化に向けて研究が進められている. ペアリング暗号は, 有限体上の離散対数問題 (DLP) と ECDLP の双方を解く計算の困難性をその安全性の基盤としている. 従ってこれらの暗号の安全性を維持する上で, ECDLP は重要な研究課題である.

DLP を解くアルゴリズムは群の固有の性質を利用するか否かで大きく二つの方法に分類される. 群の固有の性質に依存しないものは generic algorithm とよばれ, DLP を定義できる任意の有限群に対して適用できる. 代表的な generic algorithm として Shanks の Baby-step-giant-step や, Pollard の ρ

法, λ 法 (kangaroo-algorithm) が挙げられる. DLP が定義されている有限群 G の位数を $\#G$ としたときに, これらのアルゴリズムの計算量は $O(\sqrt{\#G})$ である. 群の固有の性質を利用することで, DLP を解くために必要な計算量を $O(\sqrt{\#G})$ より小さくすることに成功しているアルゴリズムが存在する. 例えば, 標数の大きい有限体上の DLP に対しては数体篩法, 標数の小さい有限体上の DLP に対しては関数体篩法や Frobenius representation discrete logarithm algorithm 等が挙げられる. これらのアルゴリズムは指数計算法とよばれる枠組みに属している.

上記のように指数計算法は有限体上の DLP を解く場合において, それを解く計算コストの削減に成功しているが, ECDLP に対する効率の良い指数計算法の研究はまだ模索の段階にある. これまでは, ECDLP を最も効率よく解く方法は generic algorithm であったことから, 楕円曲線暗号やペアリング暗号で利用する楕円曲線等の暗号パラメータの設定には, generic algorithm の計算量とそれを使用した数値実験の結果が利用されてきた.

近年, ECDLP に対する指数計算法の研究において, Semaev の summation polynomial やグレブナー基底等を利用した新たな指数計算法が多数提案されている. その中には generic algorithm よりも計算効率が良いことを主張する文献が存在し, また一方で逆の主張をする文献も存在している. そのため, generic algorithm との比較を考慮して, これらの新たな指数計算法の効率性を議論する必要が生じている.

この比較について本稿は, generic algorithm よりも効率よく ECDLP を解くアルゴリズムが現時点では提案されていないと結論づける. その理由として以下の二つの事実を挙げる: 一つは, 新たな指数計算法の計算量評価において導入されている仮定 (frist fall degree assumption など) について, それらの仮定の正当性が理論的にも数値実験的にも十分に示されているとは言えないことである [15]. もう一つは適切な暗号パラメータ (十分大きな有限巡回群等) において, 新たな指数計算法の有効性を示す数値実験的な結果が現時点では報告されていないことである.

本稿の構成は以下のとおりである: 第 2 節では楕円曲線や ECDLP の定義など, 基本的な内容を説明する. 第 3 節では generic algorithm 及び ECDLP を解く計算の世界記録を紹介する. 第 4 節では ECDLP に対する基本的な指数計算法について述べ, 第 5 節では, それらのアルゴリズムの計算量を理解する上で必要となる, 連立代数方程式を解くアルゴリズムとその計算量について説明する. 第 6 節で近年の成果について紹介し, 第 7 節で ECDLP に対する新たな指数計算法の影響についてまとめる.

2 楕円曲線上の離散対数問題 (ECDLP)

この節ではまず楕円曲線に関するいくつかの定義及びその性質について紹介する. さらに一般の群上の離散対数問題 (DLP) 及び楕円曲線の有理点のなす群上で与えられる離散対数問題 (ECDLP) について説明する. (参考文献として [5] を挙げる.)

体 K の代数閉包を $\overline{K}$ で表す. 本稿では楕円曲線の以下の定義を採用する:

定義 2.1. K を体として $a_1, a_2, a_3, a_4, a_6 \in K$ とする. このとき等式 E を以下のように定義する:

$$E: \ f(x,y) := y^2 + a_1xy + a_3y - (x^3 + a_2x^2 + a_4x + a_6) = 0.$$

E を満たす任意の $(x_1, y_1) \in \overline{K}^2$ において, $(\partial f/\partial x, \partial f/\partial y) \neq (0,0)$ が成り立つとき, E を K 上の楕円曲線とよぶ[1].

集合 $E(K)$ を次のように定める:

$$E(K) := \{(x,y) \in K^2 : \ f(x,y) = 0\} \cup \{\mathcal{O}\}.$$

(但し, $\mathcal{O}$ は無限遠点とする.) 本稿では特に断りのない限り, $\mathcal{P} \in E(K)$ と書いた場合は $\mathcal{P} \neq \mathcal{O}$ とする. 下記のように演算等を定義することで $E(K)$ は加法群を成す (但し $\mathcal{P}_i := (x_i, y_i) \in E(K)$ とする):

- $\mathcal{O}$ を単位元とする.
- $\mathcal{P}_1$ の逆元を $-\mathcal{P}_1 := (x_1, -y_1 - a_1x_1 - a_3)$ とする.
- $\mathcal{P}_1 \neq -\mathcal{P}_2$ のとき $\mathcal{P}_3 := \mathcal{P}_1 + \mathcal{P}_2$ を以下のように定める:

$$\begin{aligned}
\lambda &= \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} & \text{if } x_1 \neq x_2, \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & \text{if } x_1 = x_2, \end{cases} \\
x_3 &= \lambda^2 + a_1\lambda - a_2 - x_1 - x_2, \\
y_3 &= \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3.
\end{aligned}$$

正整数 m に対して m 個の $\mathcal{P} \in E(K)$ の和を $[m]\mathcal{P}$ で表す. さらに $[0]\mathcal{P} := \mathcal{O}$ とし, $[-m]\mathcal{P} := -[m]\mathcal{P}$ とする. 体 K が有限体であるとき, 即ちある素数べき q に対して $K = \mathbb{F}_{q^n}$ であるとき, $E(\mathbb{F}_{q^n})$ は有限群である. 従って $\mathcal{P} \in E(\mathbb{F}_{q^n})$ を生成元として有限巡回群 $\langle \mathcal{P} \rangle$ を構成できる.

次に離散対数問題 (DLP) について説明する. 群 G の位数を $\#G$ で表す. 有限群 G 上の DLP とは, 与えられた $g, T \in G$ に対して以下の条件を満たす $X \in \mathbb{Z}/\#G\mathbb{Z}$ が存在するならばそれを求める問題である:

$$T = g^X. \tag{1}$$

[1]特にこの定義の曲線を Weierstrass model とよぶ.

((1) が解 X を持つならば, X を $\log_g T$ とかく.) 有限群 G が $E(\mathbb{F}_{q^n})$ であるとき, その離散対数問題は "楕円曲線上の離散対数問題 (ECDLP) " とよばれる. この場合, 楕円曲線から与えられる加法群の表記方法に合わせて, 一般的に以下のように ECDLP を表す. 即ち, ECDLP とは $\mathcal{T}, \mathcal{P} \in G$ に対して以下の条件を満たす $X \in \mathbb{Z}/\#G\mathbb{Z}$ が存在するならばそれを計算する問題である:

$$\mathcal{T} = [X]\mathcal{P}. \tag{2}$$

本稿では, "有限体上の離散対数問題[2]" 等の一般的によく使用される呼び方に適した, 上述の DLP の定義を採用した. しかし, 特に暗号の分野では解が存在する DLP について考える理由等から, 一般的には有限群 G は有限巡回群で定義され, 議論される. 例えば (2) で与えられる ECDLP の場合, G は $E(\mathbb{F}_{q^n})$ ではなく巡回群 $\langle \mathcal{P} \rangle$ で考える.

楕円曲線暗号やペアリング暗号は ECDLP が解かれると解読されてしまう. 従ってこれらの暗号を安全に運用するには, ECDLP を解く計算に十分な計算時間が必要となるように適切な有限体 $\mathbb{F}_{q^n}$, 曲線 E, 及び巡回群 $\langle \mathcal{P} \rangle$ 等を選択する必要がある. その選択の例として, DLP を与える巡回群 G の位数 $\#G$ を十分大きな素数にすることや, $E(\mathbb{F}_{2^n})$ 上での ECDLP を利用する場合は n は素数となるように選ぶことなどが挙げられる. (参考文献として [15] を挙げる.) 以降の節で ECDLP を解くアルゴリズムについて説明していく.

3 Generic algorithm による DLP の計算

この節では任意の有限群上で定義されている DLP を解くことに適用可能なアルゴリズムである generic algorithm について説明する. また, その代表的なアルゴリズムである Shanks の baby-step-giant-step [28], Pollard の ρ 法及び λ 法 [25] について簡単に述べる. これらのアルゴリズムの計算量は, 与えらた有限群を G としたときに, 大まかには $O(\sqrt{\#G})$ であることが知られている[3].

3.1 Generic algorithm

本稿では [5] の定義 19.1 に基づいた以下の generic algorithm アルゴリズムの定義を採用する:

定義 3.1. 有限群 G が与えられているとして, G における以下の演算のみを行うアルゴリズムを generic algorithm とよぶ:

- 二項演算,

[2]有限体 $\mathbb{F}_{q^n}$ 上の DLP とは, 乗法群 $G = \mathbb{F}_{q^n}^*$ 上の DLP のことである.
[3]Shanks の baby-step-giant-step の計算量は正確には $O(\sqrt{\#G}\log \#G)$ である.

- 逆元の演算,
- 二つの元が等しいか否かの確認.

Generic algorithm では, 与えらえた有限群が固有の性質[4]を持っていたとしても, それを利用した計算が行われない.

3.2 Shanks の baby-step-giant-step

Shanks の baby-step-giant-step [28] について簡単に説明する. 式 (1) で表される, 巡回群 $G = \langle g \rangle$ 上の DLP を解くことを考える. まず $M := \lceil \#G \rceil$ として $h = (g^{-1})^M$ を計算し, 以下のリスト BL, GL を計算する:

$$\begin{aligned} BL &= \{g^i : \ i = 0, \cdots M-1\}, \\ GL &= \{Th^j : \ j = 0, \cdots M-1\}. \end{aligned}$$

(BL の計算は baby-step, GL の計算は giant-step とよばれる.) リスト BL, GL を並べ替え, 重複箇所 $g^i = Th^j$ を探し, $X = i + jM \pmod{\#G}$ を返す. このアルゴリズムは, リストの生成で $O(\sqrt{\#G})$ 回の群演算及び $O(\sqrt{\#G})$ のデータ保存空間を必要とし, さらにリストの並び替えと重複箇所の探索で $O(\sqrt{\#G} \log \#G)$ 回の比較を実行する. このアルゴリズムの様々な改良が提案されているがオーダーとしての計算量は変わらない. (参考文献として [15] を挙げる.)

3.3 Pollard の ρ 法と λ 法

第 3.2 節で紹介した Shanks の baby-step-giant-step は確定的なアルゴリズムであるが多くのデータ保存空間を必要とする. 一方で Pollard の ρ 法や λ 法は birthday paradox を利用する確率的なアルゴリズムである. しかし, データ保存空間は baby-step-giant-step に比べてずっと小さく, 巡回群 G 上の DLP を解くために必要な群演算の回数は $O(\sqrt{\#G})$ である. この節では基本的な ρ 法を説明する [7]. (詳しくは [5], [15] を参照されたい.)

ここでも第 3.2 節と同じ DLP (1) を解くとする. 基本的な ρ 法では, まず巡回群 G を三つのほぼ同じ位数を持つ互いに交わりのない集合 G_1, G_2, G_3 に分割する:

$$G = G_1 \cup G_2 \cup G_3.$$

[4]例えば群の位数が小さな素数の積で表される場合等が挙げられる.

表 1: ECDLP に関する計算の記録

曲線の種類	サイズ (bit)	年	著者
素体	112	2009	Bos et al. [2]
標数 2 の拡大体	118	2016	Bernstein et al. [1]
Koblitz	113	2014	Wenger and Wolfger[29]

次に $(a_i, b_i) \in (\mathbb{Z}/\#G\mathbb{Z})^2$ に対して, $h_i := T^{a_i} g^{b_i} \in G$ なる数列 $\{h_i\}$ を考える. 但し, $a_0 = b_0 = 0, h_0 = e$ (e は G の単位元) とし,

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i + 1, b_i) & (h_i \in G_1), \\ (2a_i, 2b_i) & (h_i \in G_2), \\ (a_i, b_i + 1) & (h_i \in G_3) \end{cases}$$

とする. このとき次が成り立つ:

$$h_{i+1} = \begin{cases} Th_i & (h_i \in G_1), \\ h_i^2 & (h_i \in G_2), \\ gh_i & (h_i \in G_3). \end{cases}$$

実際の計算では $(h_i, a_i, b_i, h_{2i}, a_{2i}, b_{2i})$ のみを保持し, $h_i = h_{2i}$ なる i を計算する. このとき $T^{a_i} g^{b_i} = T^{a_{2i}} g^{b_{2i}}$ であることから次が成り立つ:

$$T^{a_i - a_{2i}} = g^{b_{2i} - b_i}.$$

暗号では $\#G$ は十分大きな素数となるように設定されるため, 高い確率で $\gcd(a_i - a_{2i}, \#G) = 1$ が期待できることから,

$$X = (b_{2i} - b_i)(a_i - a_{2i})^{-1} \pmod{\#G}$$

を計算することで解 X が得られる.

ρ 法の計算量について説明する. $\{h_i\}$ がランダムな数列であれば birthday paradox により, $i = O(\sqrt{\#G})$ で $h_i = h_{2i}$ となる確率が 1/2 以上になると期待できる. また, $(h_i, a_i, b_i, h_{2i}, a_{2i}, b_{2i})$ から $(h_{i+1}, a_{i+1}, b_{i+1}, h_{2(i+1)}, a_{2(i+1)}, b_{2(i+1)})$ を計算するには 3 回の群演算を必要とするだけであることから, 合計で $O(\sqrt{\#G})$ 回の群演算を必要とする.

ここで ECDLP に関する近年の代表的な計算の記録を紹介する. 表 1 の結果は ρ 法を改良した generic algorithm によるものである [18]. このことは, 楕円曲線の有理点のなす群が持つ固有の性質を利用して, ECDLP を効率良く解くアルゴリズムがまだ発見されていないことを意味する.

4　ECDLP に関する指数計算法

第 3 節で述べたように, generic algorithm を用いることで, 任意に与えられた有限群 G 上の DLP は $O(\sqrt{\#G})$ 回の群演算, 即ち指数時間で解くことがで

きる. 一方で有限体上の DLP は, 数体篩法や関数体篩法など, 指数計算法とよばれる枠組みに属する方法を使用することで指数時間より小さい計算量, 即ち準指数時間や quasi-polynomial time で解かれることが知られている. (詳しくは [5] [8] を参照.) 近年, 指数計算法を ECDLP に導入した研究が進められている. この節では, 指数計算法について簡単に説明し, それを ECDLP に導入する際に道具として使われる Semaev の summation polynomial と Weil descent について述べる. (参考文献として [15] を挙げる.)

4.1 指数計算法

指数計算法は大きく二つの種類に分けられるため, これら二つの違いについて説明する. 第 4.1.1 節で述べる指数計算法は ECDLP を解く場合によく議論されており, 第 4.1.2 節で紹介する指数計算法は有限体上の DLP を解く場合によく利用されている. 本稿では前者を指数計算法 1, 後者を指数計算法 2 とよぶことにする. また本稿では指数計算法 1 を中心に議論をする.

以下二つの注意を紹介する. 指数計算法は任意の有限群に適用することができるが, 計算効率を上げるために与えられた群が持つ固有の性質を利用するため, 一般的には generic algorithm に分類されない. 第 4.1 節では (1) で表される, 有限巡回群 $G = \langle g \rangle$ 上の DLP が与えられているとする.

4.1.1 指数計算法 1

ECDLP を考える場合によく扱われる指数計算法 1 の概要を以下に与える;

初期設定段階: 因子基底とよばれる G の部分集合 $FB := \{\pi_1, \cdots \pi_s\}$ を設定する.

関係 (relation) 探索段階:

(i) $a, b \in \mathbb{N}$ を選び $R = g^a T^b \in G$ を計算する.

(ii) 下記の等式を満たす非負整数 e_ℓ の組 $(e_1, \cdots, e_s)$ が存在するかを判定し[5], 存在するならばそれを計算する:

$$R = \prod_{\ell=1}^{s} \pi_\ell^{e_\ell}. \tag{3}$$

等式 (3) は relation とよばる. この relation から, 因子基底の元および T の 離散対数を解とする線形方程式

$$a + bX \equiv \sum_{\ell=1}^{s} e_\ell \log_g \pi_\ell \pmod{\#G}$$

[5] 一般的に e_ℓ には上界が与えらているため, 常に (3) を満たす $(e_1, \cdots, e_s)$ が存在する保証はない.

が生成される．実際の計算では (a,b) と $(e_1,\cdots,e_s)$ を結合したベクトルを行列の行 (または列) として保存する．

(iii) (i), (ii) の計算を十分な個数の relation が得られるまで繰り返す．

線形代数段階: 関係探索段階で得られた行列に対して, $\#G$ を法とする行列操作を行うことで以下を満たす X_1, X_2 を計算する:

$$e = g^{X_1} T^{X_2}.$$

(但し e は G の単位元とする.)

離散対数計算段階: $X_2^{-1} \pmod{\#G}$ が存在するならば以下を返す:

$$X = -X_1 X_2^{-1} \pmod{\#G}.$$

上記のように, 指数計算法は四つの計算段階から構成される．初期設定段階では因子基底を設定するとしているが, 他にも G の表現に使用する数値等 (例えば有限次拡大体を表す多項式など) を設定することもある．この段階に必要な計算コストは一般的に無視できるほど小さい．従って, 文献によっては初期設定段階を一つの計算段階として数えず, 指数計算法は他の三つの計算段階から構成されていると定義する場合もある．

因子基底の設定は指数計算法の効率を決定する重要な要素である．以下に因子基底に求められる性質について紹介する:

- 因子基底の個数 s に対して, 線形代数段階で扱われる行列の大きさは $O(s)$ である．よって, この段階での行列操作の計算量は, ある定数 $2 < \omega \leq 3$ に対して $O(s^{\omega})$ であるため, 因子基底の個数 s は可能な限り小さいことが望ましい．

- Relation が得られる確率が可能な限り高くなるように因子基底を設定する必要がある．その理由は, この確率が低いと関係探索段階での計算を繰り返す回数が大きくなってしまうことである．因子基底の個数 s が小さいほどその確率は小さくなる．従って関係探索段階と線形代数段階の計算コストはトレードオフの関係にある．

- Relation を計算するコストが小さくなるように因子基底を設定する必要がある．

第 4.4 節で説明するように, 指数計算法 1 で ECDLP を解く場合, relation (3) を生成するために有理点 R を因子基底の元である有理点の和で表す計算を効率良く実行する必要がある．この計算を point decomposition とよぶ．近年では Semaev の summation polynomial (第 4.2 節) に Weil descent (第 4.3) を適用することで連立代数方程式を生成し, それをグレブナー基底を計算するアルゴリズムなどを利用して解くこと (第 5 節) で point decomposition を実行する研究が進められている．

4.1.2 指数計算法 2

有限体上の DLP を解く場合に数体篩法や関数体篩法などおいて, 上述の指数計算法 1 を少し変更した指数計算法 2 の枠組みがよく利用される. 本稿を理解する上でこの節を読み飛ばしても問題ないが, 指数計算法 2 を ECDLP に適用する議論もあるため, この節で簡単に紹介する. その主な変更内容は関係探索段階において, 与えられた DLP を定義する T を含まない relation を生成することと, T を因子基底の元で表す計算を離散対数計算段階に追加することである. 以下に指数計算法 2 の概要を与える;

初期設定段階: 因子基底 $FB := \{\pi_1, \cdots \pi_s\}$ を設定する.

関係探索段階:

(i) 下記の等式を満たす非負整数 e_ℓ の組 $(e_1, \cdots, e_L, e_{L+1}, \cdots e_s)$ が存在するかを判定し, 存在するならばそれを計算する:

$$\prod_{\ell=1}^{L} \pi_\ell^{e_\ell} = \prod_{\ell=L+1}^{s} \pi_\ell^{e_\ell}. \tag{4}$$

この relation は以下の線形方程式に対応する:

$$\sum_{\ell=1}^{L} e_\ell \log_g \pi_\ell \equiv \sum_{\ell=L+1}^{s} e_\ell \log_g \pi_\ell \pmod{\#G}.$$

ベクトル $(e_1, \cdots, e_s)$ を行列の行 (または列) として保存する.

(ii) (i) の計算を十分な個数の relation が得られるまで繰り返す.

線形代数段階: 関係探索段階で得られた線形方程式の解 $\log_g \pi_1, \cdots, \log_g \pi_s$ を求める.

離散対数計算段階: $a, b \in \mathbb{N}$ を選び $R = g^a T^b$ を因子基底の元で表す:

$$R = \prod_{\ell=1}^{s} \pi_\ell^{t_\ell}.$$

このとき $b^{-1} \pmod{\#G}$ が存在するならば

$$X = (\sum_{\ell=1}^{s} t_\ell \log_g \pi_\ell - a) b^{-1} \pmod{\#G}$$

が成り立つことから, 線形代数段階で計算した解を上記の等式に代入することで X を得る.

4.2 Semaev の Summation polynomial

第 4.2 節以降では ECDLP について議論するため, 式 (2) で表される等式について考える. 指数計算法 1 で ECDLP を解く場合, 関係探索段階において選んだ $a, b \in \mathbb{N}$ に対して

$$\mathcal{R} = [a]\mathcal{P} + [b]\mathcal{T} \tag{5}$$

を計算する. 次に $\mathcal{R}$ を因子基底 $FB = \{\pi_1, \cdots, \pi_s\}$ の和

$$\mathcal{R} = \sum_{\ell=1}^{s} [e_\ell]\pi_\ell$$

として表現する計算, 即ち point decomposition を試み, それが可能であれば

$$\log_{\mathcal{P}} \mathcal{R} \equiv \sum_{\ell=1}^{s} [e_\ell] \log_{\mathcal{P}} \pi_\ell \pmod{\#\langle\mathcal{P}\rangle}$$

を得る.

Semaev は, 標数が 2 でも 3 でもない有限体 $\mathbb{F}_{q^n}$ 上の楕円曲線 E が与えられた場合に, point decomposition の計算に使用する道具として summation polynomial を提案した [26]. この場合, E は以下の式で表すことができる:

$$y^2 = x^3 + Ax + B. \tag{6}$$

(但し A, B は $\mathbb{F}_{q^n}$ の元で $\Delta := 4A^3 + 27B^2 \neq 0$ を満たすものとする.)

定理 4.1. q は 5 以上の奇数素数のべきとし, $E: \ y^2 = x^3 + Ax + B$ は $\mathbb{F}_{q^n}$ 上の 楕円曲線とする. このとき $2 \leq m \in \mathbb{N}$ に対して第 m-summation polynomial S_m を以下のように定義する:

$$\begin{aligned}
S_2(x_1, x_2) &= x_1 - x_2, \\
S_3(x_1, x_2, x_3) &= (x_1 - x_2)^2 x_3^2 - 2((x_1 + x_2)(x_1 x_2 + A) + 2B)x_3 \\
&\quad + (x_1 x_2 - A)^2 - 4B(x_1 + x_2), \\
S_m(x_1, \cdots, x_m) &= \mathrm{Res}_x(S_{m-M}(x_1, \cdots, x_{m-M-1}, x), S_{M+2}(x_{m-M}, \cdots, x_m, x)) \\
&\quad (\text{if } m \geq 4, 1 \leq M \leq m-3).
\end{aligned}$$

ただし, Res は終結式とする [4]. $m \geq 3$ のとき, S_m は絶対既約で対称な多項式であり, さらに各変数 x_i に対して $\deg_{x_i}(S_m) = 2^{m-2}$ である.

Summation polynomial S_m は標数が 2 または 3 の場合にも自然に拡張できる [14] [15] [26].

ここで point decomposition に利用する S_m の性質を紹介する: 即ち $\overline{x_1}, \cdots, \overline{x_m} \in \overline{\mathbb{F}_{q^n}}$ が

$$S_m(\overline{x_1}, \cdots, \overline{x_m}) = 0 \tag{7}$$

を満たすことと,

$$\mathcal{P}_1 + \cdots + \mathcal{P}_m = \mathcal{O}$$

なる $\mathcal{P}_i = (\overline{x_i}, \overline{y_i}) \in E(\overline{\mathbb{F}_{q^n}})$ が存在することは同値である. この S_m の性質を利用して, 関係探索段階で与えられた (5) の $\mathcal{R}$ を因子基底 $FB = \{\pi_1, \cdots, \pi_s\}$ の元の和で

$$\mathcal{R} = \pi_{\ell_1} + \cdots + \pi_{\ell_m} \tag{8}$$

のように表すことを考える. まず $\mathcal{Q} := (x, y) \in E(\overline{\mathbb{F}_{q^n}})$ の x 座標を以下のように表記する:

$$x(\mathcal{Q}) := x.$$

S_{m+1} の x_{m+1} に $x(\mathcal{R})$ を代入した等式

$$S_{m+1}(x_1, \cdots, x_m, x(\mathcal{R})) = 0 \tag{9}$$

を解くことを試みたとして, その解 $(\overline{x_1}, \cdots, \overline{x_m}) \in (\mathbb{F}_{q^n})^m$ が存在したとする[6]. このとき, 各 $\overline{x_i}$ に対して $\overline{x_i} = x(\pi_{\ell_i})$ なる $\pi_{\ell_i} \in FB$ が存在するならば relation が得られる[7].

代数方程式 (9) を効率良く解くために, Frobenius 写像を利用した等式と (9) から構成される下記の連立代数方程式を解く方法が考えられる:

$$\begin{cases} 0 = S_{m+1}(x_1, \cdots, x_m, x(\mathcal{R})), \\ 0 = x_1^{q^n} - x_1, \\ \vdots \\ 0 = x_m^{q^n} - x_m. \end{cases} \tag{10}$$

次の第 4.3 節では連立代数方程式 (10) を解くために Weil descent を導入した方法について説明する. またグレブナー基底を利用した連立代数方程式の解法については第 5 節で説明する.

4.3 Weil descent

Semaev によって導入された summation polynomial を利用して ECDLP を解く指数計算法は, Weil descent を導入することによって, Diem や Gaudry らによって改良されていった [9] [16]. この節では Weil descent を紹介し, それを連立代数方程式 (10) を解くためにどのように利用するかを説明する.

まず Weil descent に関する以下の補題を紹介する [15]:

[6] 存在しない場合は relation が得られないため, $\mathcal{R}$ をとりなおして同様の計算を行うことになる.

[7] この場合も relation が得られないため, $\mathcal{R}$ をとりなおして同様の計算を行うことになる.

補題 4.2. 素数べき q と自然数 n に対して, $\mathbb{F}_{q^n}$ を n 次元の $\mathbb{F}_q$ ベクトル空間としてみたときの基底を $\{\theta_1, ..., \theta_n\}$ とする. さらに $f(x_1, ..., x_m) \in \mathbb{F}_{q^n}[x_1, ..., x_m]$ とする. このとき $Z := \{z_{i,j} : 1 \leq i \leq m,\ 1 \leq j \leq n\}$ に対して, 以下の等式を満たす $f_k(Z) \in \mathbb{F}_q[Z]$ がただ一つ存在する:

$$f(z_{1,1}\theta_1 + \cdots + z_{1,n}\theta_n, \cdots, z_{m,1}\theta_1 + \cdots + z_{m,n}\theta_n) = \sum_{k=1}^{n} \theta_k f_k(Z).$$

さらに, ある $(\overline{x_1}, \cdots, \overline{x_m}) \in (\mathbb{F}_{q^n})^m$ に対して $f(\overline{x_1}, \cdots, \overline{x_m}) = 0$ であるならば, ある $\overline{z_{i,j}} \in \mathbb{F}_q$ が存在して以下の条件を満たす:

$$\begin{aligned} \overline{x_i} &= \sum_{j=1}^{n} \overline{z_{i,j}}\theta_j, \\ f_k(\overline{Z}) &= 0\ (1 \leq k \leq n). \end{aligned}$$

補題 4.2 により, $\mathbb{F}_{q^n}$ 係数の m 変数代数方程式は $\mathbb{F}_q$ 係数の mn 変数の n 個の代数方程式に変換される. さらに補題 4.2 における $f(x_1, \cdots, x_m)$ を (9) の左辺の多項式としたときに, Weil descent によって連立代数方程式 (10) は $\mathbb{F}_q$ 係数の mn 変数の $n + mn$ 個の等式で構成される以下の形の連立代数方程式に変換される:

$$\begin{cases} 0 = f_1(Z), \\ \vdots \\ 0 = f_n(Z), \\ 0 = z_{1,1}^q - z_1, \\ \vdots \\ 0 = z_{m,n}^q - z_{m,n}. \end{cases} \tag{11}$$

Weil descent を行う前に比べて, 変数と等式の個数はともに増加するが, 連立代数方程式を解く際に扱う多項式の各変数の次数を q より小さくできることが利点である.

因子基底の設定を工夫することで, Weil descent によって生成される連立代数方程式の変数の個数を削減することができる. まず $\mathbb{F}_{q^n}$ のある $\mathbb{F}_q$ ベクトル部分空間を V とし, その次元を $1 \leq n' < n$ とする. このとき因子基底 FB を次のように定める:

$$FB = \{\pi_\ell \in E(\mathbb{F}_{q^n}) \mid x(\pi_\ell) \in V\}. \tag{12}$$

この因子基底の設定により x_i は Weil descent によって n' 個の変数 $z_{i,j}$ $(1 \leq j \leq n')$ で表されるため, 方程式 (9) は $\mathbb{F}_q$ 係数の $mn'(< mn)$ 変数の n 個の代数方程式に変換される. 最終的にはそれらの代数方程式に Frobenius map に対応する等式 $z_{i,j}^q - z_{i,j} = 0$ を加えた以下の形の連立代数方程式を解

く (但し $Z' := \{z_{i,j} :\ 1 \leq i \leq m,\ 1 \leq j \leq n'\} \subsetneq Z$ で $z_{i,j} \in Z'$ とする.):

$$\begin{cases} 0 = f_1(Z'), \\ \vdots \\ 0 = f_n(Z'), \\ 0 = z_{1,1}^q - z_1, \\ \vdots \\ 0 = z_{m,n'}^q - z_{m,n'}. \end{cases} \tag{13}$$

n' を小さくすると (13) の変数が少なくなることによりそれを解くために必要な計算コストは削減されるが, (13) が解をもつ確率も下がってしまう.

4.4 ECDLP に関する指数計算法の概要

ECDLP に指数計算法を適用する様々な方法が提案されている [15]. その中で近年, 議論が進められている代表的な方法として, 指数計算法 1 に Semaev の summation polynomial と Weil descent を導入する方法が挙げられる. この節ではその方法について簡単に紹介し, その計算量評価について説明する. またその説明のため, 式 (2) で表される $E(\mathbb{F}_{q^n})$ 上の ECDLP が与えられているとする:

$$\mathcal{T} = [X]\mathcal{P} \in G = \langle \mathcal{P} \rangle \subset E(\mathbb{F}_{q^n}).$$

ECDLP に関する指数計算法;

初期設定段階: 因子基底 FB を (12) のように設定する:

$$FB = \{\pi_\ell \in E(\mathbb{F}_{q^n}) \mid x(\pi_\ell) \in V\}$$

関係探索段階:

(i) $a, b \in \mathbb{N}$ を選び $\mathcal{R} = [a]\mathcal{P} + [b]\mathcal{T}$ を計算する.

(ii) Semaev の summation polynomial S_{m+1} の x_{m+1} に $x(\mathcal{R})$ を代入し[8], それに対して Weil descent を実行することで連立代数方程式 (13) を計算する:

$$\begin{cases} 0 = f_1(Z'), \\ \vdots \\ 0 = f_n(Z'), \\ 0 = z_{1,1}^q - z_1, \\ \vdots \\ 0 = z_{m,n'}^q - z_{m,n'}. \end{cases}$$

[8] m は固定された自然数である.

この連立代数方程式を解き, その解

$$(\overline{z_{1,1}}, \cdots, \overline{z_{m,n'}}) \in (\mathbb{F}_q)^{mn'}$$

が存在するならば[9], その解に対応する $(\overline{x_1}, \cdots, \overline{x_m}) \in (\mathbb{F}_{q^n})^m$ を求める. 即ち V の基底 $\theta_1, \cdots, \theta_{n'} \in \mathbb{F}_{q^n}$ に対して

$$\overline{x_i} = \sum_{j=1}^{n'} \overline{z_{i,j}} \theta_j \in V$$

を計算する. そのような一つの組 $(\overline{x_1}, \cdots, \overline{x_m})$ に対して

$$(\overline{x_1}, \cdots, \overline{x_m}) = (x(\pi_{\ell_1}), \cdots, x(\pi_{\ell_m})) \tag{14}$$

を満たす因子基底の元の組 $(\pi_{\ell_1}, \cdots, \pi_{\ell_m})$ は高々 2^m 個存在する. その中から

$$\mathcal{R} = \sum_{i=1}^{m} \pi_{\ell_i} \tag{15}$$

を満たすものを求め, 下記の等式を満たす非負整数 e_ℓ の組み $(e_1, \cdots, e_s)$ を決定する:

$$\mathcal{R} = \sum_{\ell=1}^{s} [e_\ell] \pi_\ell. \tag{16}$$

この relation (16) は下記の線形方程式に対応する:

$$a + bX \equiv \sum_{\ell=1}^{s} e_\ell \log_{\mathcal{P}} \pi_\ell \pmod{\#G}.$$

実際の計算では (a, b) と $(e_1, \cdots, e_s)$ を結合したベクトルを行列の行 (または列) として保存する.

(iii) (i), (ii) の計算を十分な個数の relation が得られるまで繰り返す.

線形代数段階: 関係探索段階で得られた行列に対して, $\#G$ を法とする行列操作を行うことで以下を満たす X_1, X_2 を計算する:

$$\mathcal{O} = [X_1]\mathcal{P} + [X_2]\mathcal{T}.$$

離散対数計算段階: $X_2^{-1} \pmod{\#G}$ が存在するならば[10] 以下を返す:

$$X = -X_1 X_2^{-1} \pmod{\#G}.$$

[9] 存在しないならば, 新しく a, b を選び直して $\mathcal{R}$ を生成して, 同様の計算を続ける.

[10] 暗号で利用する ECDLP を考えた場合, $\#G$ は十分大きな素数となるように選ばれるため, $X_2^{-1} \pmod{\#G}$ が存在しない確率は無視できる.

ここから上記のアルゴリズムの計算量について議論する．この計算量の基本的な評価方法の概要は [15] でまとめられている．さらに [15] では, $E(\mathbb{F}_{q^n})$ 上の ECDLP が与えらえているとして, q が n に比べて十分小さい場合及びその逆の場合について分類して説明している．これは暗号で利用される楕円曲線が, $\mathbb{F}_{2^n}$ (n は素数) または $\mathbb{F}_{q^n}$ (q は素数で n は十分小さい) 上で定義されたものが多いためである．

有限体に関する上記の分類によらず, S_{m+1} の m と $n'(=\dim V)$ は一般的に

$$mn' \approx n \tag{17}$$

となるように設定される [14]．(このように設定しない方法もある [15].) また, S_{m+1} の生成に必要な体演算は

$$O(2^{m^2}) \tag{18}$$

であることが知られている [14].

関係探索段階で生成した $\mathcal{R}$ が (15) のように表される確率 $\mathrm{Prob}_{\mathrm{sum}}$ について考える．ここからの議論の準備として, 因子基底の個数 s は

$$s \approx \#V \tag{19}$$

を満たすとして良い．その理由は, ランダムに選んだ $\overline{x_i} \in \mathbb{F}_{q^n}$ を与えられた楕円曲線の x 座標として持つ有理点が存在する確率が約 1/2 であることと, 同じ x 座標を持つ有理点の個数の期待値が約 2 であることである．

さらに, この節の目的は上述の指数計算法とその計算量の評価方法を大まかに理解することであるため, $q << n$ の場合について議論する．(その逆の場合, 即ち $q >> n$ の場合も計算量の評価方法はほぼ同様である．参考文献として [15] を挙げる.) この場合, (19) より

$$s \approx q^{n'} \tag{20}$$

が成り立つ．

確率 $\mathrm{Prob}_{\mathrm{sum}}$ は, 因子基底 FB に属する m 個の元の和でかける有理点 $\mathcal{R} \in E(\mathbb{F}_{q^n})$ の割合で見積もる．そのため, そのような m 個の元の和において生じる重複を無視できると仮定している．和の対称性を考慮して, 因子基底 FB から m 個の元を選ぶ組み合わせの数は大まかに $s^m/m!$ である．また, $E(\mathbb{F}_{q^n})$ の位数は約 q^n であることと, (20) より

$$\mathrm{Prob}_{\mathrm{sum}} \approx \frac{s^m}{m! \cdot q^n} \approx \frac{1}{m!} \tag{21}$$

が成り立つ．

線形代数段階で扱う行列のランクが $O(\#V) = O(s)$ であることと (21) より, 関係探索段階で実行される point decomposition の回数は

$$O(m!s) = O(m!q^{n'})$$

と評価される．従って, point decomposition の計算コストを $\mathrm{C}_{\mathrm{dcmp}}$ とすると, 関係探索段階の計算量は

$$O(q^{n'} m! \mathrm{C}_{\mathrm{dcmp}}) \tag{22}$$

となる．線形代数段階の計算量は, 実行可能行列乗算指数 $2 < \omega \leq 3$ に対して

$$O(s^{\omega}) = O(q^{n'\omega}) \tag{23}$$

である．よって, (18), (22), (23) より, 全体の計算量は

$$O(2^{m^2} + q^{n'} m! \mathrm{C}_{\mathrm{dcmp}} + q^{n'\omega}) \tag{24}$$

である．

ここで問題となるのが point decomposition の計算量 $\mathrm{C}_{\mathrm{dcmp}}$ の評価である．この計算量は連立代数方程式 (13) を解くために必要な計算量である．連立代数方程式を効率よく解く方法については第 5 節で説明する．

5　有限体における連立代数方程式の解法

連立代数方程式 (13) を更生する多項式集合で生成されるイデアルは 0 次元であるため, この節で扱う多項式集合 F も同様の性質を持つとする．そのような F で表現される連立代数方程式を効率よく解く方法として以下の計算を組み合わせる方法が知られている:

- F_4-style のアルゴリズム (第 5.2 節) によって多項式集合 F の全次数逆辞書式順序のグレブナー基底 GB_{DRL} を計算する．
- FGLM (第 5.3 節) を利用して GB_{DRL} を辞書式順序のグレブナー基底 GB_{LEX} に変換する．

GB_{LEX} は, F で与えられる連立代数方程式と同じ解の集合を持つ連立代数方程式を構成する多項式集合である．また, F で生成されるイデアルが 0 次元であるとき, GB_{LEX} はある変数に関する一変数多項式を含む．さらに, その一変数多項式の解を GB_{LEX} の他の多項式に代入することで新たな一変数多項式を得る．この計算を繰り返すことで連立代数方程式の全ての解を解の個数の多項式時間で計算することができる．しかし, GB_{LEX} の計算コストは GB_{DRL} のそれより大きいことが経験的に知られている．そこでまずは GB_{DRL} を計算し, FGLM を利用して GB_{DRL} を GB_{LEX} に変換する．

この節では F_4-style のアルゴリズムと FGLM の計算量について簡単に説明する．また K を体, X を $\{x_1, \cdots, x_m\}$ なる変数の集合とし, $F := \{f_1, \cdots, f_k\} \subset K[X]$ とする．さらに X で生成される項全体の集合を $\mathrm{T}(X)$ で表す．

5.1 連立代数方程式とグレブナー基底の計算

第 5 節で述べたように, 多項式集合 F で与えられる連立代数方程式を解くために, まず F_4-style のアルゴリズムで F の GB_{DRL} 計算する必要がある. この節では F_4-style のアルゴリズムを理解する準備として, グレブナー基底の基本的な計算方法である Buchberger アルゴリズムのキーポイントを説明する.

定義 5.1. 多項式 $f \in K[X]$ と $K[X]$ における項順序 $\prec$ が与えられているとする. この項順序に関して f で最も大きい項 $HT_{\prec}(f)$ を頭項とよび, その係数 $HC_{\prec}(f)$ を頭係数とよぶ. さらに $HC_{\prec}(f)HT_{\prec}(f)$ を頭単項式とよび $HM_{\prec}(f)$ で表す.

Buchberger アルゴリズムでは, S 多項式の計算と多項式集合による多項式の簡約の計算を繰り返すことでグレブナー基底を計算する. 多項式 $f_1, f_2 \in K[X]$ の S 多項式 $\mathrm{Spoly}(f_1, f_2)$ は次のように定義される:

$$\mathrm{Spoly}(f_1, f_2) := \frac{\mathrm{lcm}(\mathrm{HT}_{\prec}(f_1), \mathrm{HT}_{\prec}(f_2))}{\mathrm{HM}_{\prec}(f_1)} f_1 - \frac{\mathrm{lcm}(\mathrm{HT}_{\prec}(f_1), \mathrm{HT}_{\prec}(f_2))}{\mathrm{HM}_{\prec}(f_2)} f_2. \tag{25}$$

ただし, 単項式 m_1, m_2 に対して $\mathrm{lcm}(m_1, m_2)$ はそれらの最小公倍単項式とし, その係数は 1 とする. F_4-style のアルゴリズムにおいて, (25) に現れる f_1 の倍多項式は left-side とよばれ, 同様に f_2 の倍多項式は right-side とよばれる. $\mathrm{Spoly}(f_1, f_2)$ の計算では, f_1, f_2 をそれぞれ単項式倍したものの集合 $\{m_i f_i : m_i \in \mathrm{T}(X)\}$ の中から頭項が一致する項順序が最小の組 $(m_1 f_1, m_2 f_2)$ を選び, その差を計算することでその頭項を消去している. これは互除法において最大の項を削除する計算の一般化である.

次に多項式集合による多項式の簡約について説明する.

定義 5.2. $f_1, f_2 \in K[X]$ としたときに, $\mathrm{HT}_{\prec}(f_2)$ で割り切れる f_1 の項 M が存在し, その係数を C_M とする. このとき

$$f_3 := f_1 - \frac{C_M M}{\mathrm{HM}(f_2)} f_2$$

とする. この操作を f_1 の f_2 による単項簡約とよび,

$$f_1 \underset{f_2}{\longrightarrow} f_3 \tag{26}$$

と書く.

多項式による単項簡約 (26) は多項式集合 F の元による 0 回以上の単項簡約に拡張することができ,

$$f_1 \xrightarrow[F]{*} f_3 \tag{27}$$

のように表す．この操作は単に F による f_1 の簡約とよぶ．また, (27) の f_3 に対して F による単項簡約を 1 回以上実行できないとき, f_3 は f_1 の F による剰余とよぶ．

Buchberger アルゴリズムで実行される多項式の簡約操作の計算効率を上げるアルゴリズムとして F_4-style のアルゴリズムが挙げられる．第 5.2 節で F_4-style のアルゴリズムについて述べる．Buchberger アルゴリズムの詳細については [6] を参照されたい．

5.2 F_4-style のアルゴリズムとその計算量

グレブナー基底を効率よく解くアルゴリズムとして, J.-C. Faugére によって提案された F_4 アルゴリズム [10] 及び F_5 アルゴリズム [11] が存在する．(以後, それぞれを単純に F_4, F_5 とよぶ.) これら二つのアルゴリズムはグレブナー基底の計算の高速化を図ったものである．F_4 は Macaulay 行列 (定義 5.3) の性質を利用することで多項式の簡約操作の効率化を行う．このようなアルゴリズムは F_4-style のアルゴリズムとよばれる．F_5 では F_4-style のアルゴリズムに signature という概念を導入して不要な S 多項式の生成を排除している．

F_4-style のアルゴリズムの計算量は, 基本的に入力とする多項式集合 F が生成するイデアルが斉次の場合で評価されている．非斉次の場合には, 新たに変数を一つ追加して F の各多項式を斉次化した上で計算量を評価する．そのため F の生成するイデアルが 0 次元のとき, 斉次化した多項式が生成するイデアルは一般に 1 次元となる．

以下でも特に断らない限り, 入力とする多項式集合 F が生成するイデアルは 0 次元, 即ち零点が有限個の場合に限定して説明する．イデアルのグレブナー基底計算は Macaulay 行列の行簡約化が基本となる．

5.2.1 F_4-style のアルゴリズム

F_4-style のアルゴリズムでは Macaulay 行列の部分行列を利用する:

定義 5.3. $F = \{f_1, \cdots, f_k\} \subset K[X]$ はある $d \in \mathbb{N}$ に対して $\deg(f_1), \cdots, \deg(f_k) \leq d$ [11] を満たすとする．$m_1, m_2, \cdots \in \mathrm{T}(X)$ に対しては $m_1 \succ m_2 \succ \cdots$ が成り立つとする．さらに $t_{i,j}$ は $\deg(t_{i,j} f_i) \leq d$ を満たす全ての $t_{i,j} \in \mathrm{T}(X)$ とし, $t_{i,j} f_i = \sum_\ell c_{i,j,\ell} m_\ell$ $(c_{i,j,\ell} \in K)$ のように表現するとする．このとき F の

[11] $\deg(f_i)$ は f_i の全次数である．

d 次の Macaulay 行列 $M_d(F)$ を以下のように定義する:

$$M_d(F) := \begin{array}{c} \\ \\ t_{i,j}f_i \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} m_1 & m_2 & \cdots \end{array} \\ \begin{pmatrix} \vdots & \vdots & \\ c_{i,j,1} & c_{i,j,2} & \cdots \\ \vdots & \vdots & \end{pmatrix} \end{array}.$$

F_4-style のアルゴリズムでは各次数 d ごとに, S 多項式の left-side, right-side, 及び多項式集合の簡約で利用される多項式から構成される Macaulay 行列の部分行列を生成してグレブナー基底を計算する. 即ち, その部分行列に対して行簡約操作 (ガウス消去, 掃き出し法など) を行うことで階段形を計算し, この形からグレブナー基底の元を抽出する [21].

F_4-style のアルゴリズムによる計算の特徴として, その計算は一般に大量のメモリを必要とすることが挙げられる. F_4-style のアルゴリズムでは, そのアルゴリズムの性質上, スパース（疎）な行列を扱うことになる[12]. （これは, Macaulay 行列の部分行列である. ）巨大なスパース行列の処理を必要とする別の代表的なアルゴリズムとして数体篩法がある. 数体篩法ではスパース行列で表される線形方程式の解を求めることが目的であるため, 行列のスパース性を保持したまま効率良く計算するアルゴリズム（Lanczos 法など）を効果的に利用できる. しかし, グレブナー基底を計算する F_4-style のアルゴリズムでは簡約した結果の行列が必要であるため, スパース性を保持したまま効率良く計算することは一般的に難しい. F_4-style のアルゴリズムでは前処理として, 行列のスパース性を利用して行列のサイズを小さくするアルゴリズム（structured Gaussian elimination）を利用することが推奨されている [10]. Magma などでの F_4 の実装はブラックボックスであるため, その詳細は不明であるが, 簡約した行列を高速に計算するために, 最終的にはスパース性を犠牲にして実メモリ上で掃き出し法を実行すると考えられる. そのため, Magma で実装されている F_4 のような効率的な実装でさえ使用するメモリ量は結果として膨大になると考えられる.

5.2.2 F_4-style のアルゴリズムの計算量

F_4-style のアルゴリズムで実際に使用する部分行列の大きさは, 入力される多項式集合の多項式の個数や次数のみからでは精密には評価できない. 従って, F_4-style のアルゴリズムの計算量は, Macaulay 行列の簡約操作の計算量と同じオーダーで上から評価される[13]. (これは最悪計算量を見積もっている

[12] スパースな行列を扱う理由として, 例えば, d を初期値からいくつか大きくしたときに, 簡約で利用される多項式に対応する行が一般に疎になる傾向があることが挙げられる.

[13] F_5 の計算量評価では, F_5 による不要な計算の効果を考慮しない Macaulay 行列の行簡約操作の計算量を見積もることになる. 一方, F_4-style のアルゴリズムの場合でも, ECDLP の特殊性を考慮して部分行列のサイズをより詳細に評価する研究もある [14].

ことを意味する.) よって, グレブナー基底の計算量はグレブナー基底の元の最大次数を D とすると, D 次までの Macaulay 行列の簡約操作の計算量となる.

D 次の Macaulay 行列 $M_D(F)$ のサイズは斉次イデアルの場合には D 次の単項式の個数以下になり, それは, m 個の変数から重複を許して D 個を選ぶ組み合わせの個数である. 行列のサイズを N とするとき, 掃き出しの計算量は N^ω であることから, F_4-style の計算量は以下のように見積もられる:

$$O\left(\binom{m+D}{m}^{\omega}\right).$$

一方, 既約なグレブナー基底の元の最大次数 D は生成元の次数を用いて評価されている. 現在の方針ではイデアルによって定まる Hilbert 多項式の degree of regularity D_{reg} が D の上からの評価を与えるため, D_{reg} の大きさを評価することになる.

多項式集合 F が regular sequence とよばれる形になっていれば, D_{reg} は F に属する多項式の次数の和で抑えられるが, そうでない場合にはそれらの次数の積等で抑えることになる. F が regular sequence でないときは, そのイデアルの元で regular sequence になるものを抽出し, その差分を考えることで次数 D_{reg} を評価する.

実際に変数を $x_1, \cdots, x_m$ とし, 1 次元斉次イデアル I が $f_1, \cdots, f_k (k \geq m-1)$ で生成されているとする. ここで, d_i は f_i の次数で $d_1 \geq d_2 \geq \cdots \geq d_k$ とする. $k = m-1$ で $f_1, \cdots, f_k$ が regular sequence であれば,

$$D_{\mathrm{reg}} \leq d_1 + \cdots + d_{m-1} - m$$

である [20], [21]. しかし, regular sequence でない場合でも, 射影次元が 0 以下であれば同様のことが成り立つ [21]. また, 非斉次の場合には斉次化を行うことで, グレブナー基底の元の最大次数が評価される.

一方で, イデアルの生成系を斉次化したものがイデアル自体の斉次化を生成する場合には, 斉次での評価がそのまま使える. また, F の多項式をランダムにとった場合には, ほとんどの場合に最初の $m-1$ 個が regular sequence になる. そこで, 実際的な計算量として斉次化との計算量のギャップがないものと仮定し, かつ regular sequence の場合を想定して計算量の評価をする方向性もある. 詳細は異なるが本質的には [12], [14], などがこれに対応するものと考える.

以上より, 任意に与えられた F に対して, そのグレブナー基底を計算することなく, その D_{reg} を厳密に評価する方法は現時点では知られていない. そこで, F 自身の持つ代数的な性質を利用して D_{reg} を可能な限り厳密に評価する研究が進められている. このような背景から, 第 6.1 節で説明する first fall degree assumption (FFDA) の導入が議論されている.

5.3 FGLM とその計算量

5.3.1 グレブナー基底の項順序変換

体 K 上の多項式環 $R = K[X]$ のイデアル I の零点をグレブナー基底を用いて求める方法として, I の辞書式順序に関するグレブナー基底を求めて, 変数の少ない多項式から順に零点を求めて代入していくという方法がある. この場合, 一般に辞書式順序グレブナー基底を I の生成系から直接 Buchberger アルゴリズムや F_4 などで求めるのは効率がよくない. よって, 全次数逆辞書式順序など, グレブナー基底が比較的求めやすい項順序に関するグレブナー基底を求めておき, 辞書式順序など, 他の項順序に関するグレブナー基底を求める, 項順序変換 (Change of Ordering) と呼ばれる方法がいくつか提案されている. FGLM [13] は 0 次元イデアルに対して線形代数を応用して項順序変換を行うアルゴリズムである.

5.3.2 FGLM アルゴリズム

イデアル I が 0 次元イデアルのときは, 剰余環 R/I が K 上の線形空間として有限次元であり, I の $\overline{K}$ における零点の個数が有限個である. 以下で, $\prec$ に関するグレブナー基底 G による f の剰余を $\mathrm{NF}_\prec(f, G)$ と書くことにする. FGLM アルゴリズムは Algorithm 1 で与えられる.

FGLM アルゴリズムの原理は単項式 h を $\prec_1$ に関する頭項とする多項式がイデアル I の中に含まれるかを, 未定係数法により, h を $\prec_1$ に関する昇順で取り替えながら調べていくというものである. 調べる多項式は $f_h = h + \sum_{t \in B} \lambda_t t$ である. ここで B は, それまでに得られた G_1 の元の頭項 (これは H に格納されている) のどれでも割り切れない単項式が格納されている. $f_h|_{\lambda_t = a_t} \in I$ となるような $a_t \in K (t \in B)$ が存在するとき, $f_h|_{\lambda_t = a_t}$ が G_1 に追加され, 存在しないとき h が B に追加される. $\prec_1$ に関する簡約グレブナー基底を求めるには, h としてそれまでに得られた H のどの元でも割り切れないもののみを考えればよい. これが Algorithm 1 の 11 行目の意味である.

G が $\prec$ に関する I のグレブナー基底であることから

$$f_h|_{\lambda_t = a_t} \in I \Leftrightarrow \mathrm{NF}_\prec(f_h|_{\lambda_t = a_t}, G) = 0$$

である. NF の線形性により $\mathrm{NF}_\prec(f_h|_{\lambda_t = a_t}, G) = E|_{\lambda_t = a_t}$ を得る. E を R の単項式について整理すると

$$E = \sum_{s \in S} c_s(\lambda_t; t \in B) s$$

と書ける. ここで S は G に関する標準単項式集合 (G のどの先頭単項式でも割れないような単項式の集合) である. よって $E = 0$ は

$$c_s(\lambda_t) = 0 \quad (\forall s \in S)$$

Algorithm 1 FGLM アルゴリズム

Input : 0 次元イデアル I の $\prec$ に関するグレブナー基底 G
Output : I の $\prec_1$ に関する簡約グレブナー基底

$G_1 \leftarrow \emptyset$; $B \leftarrow \emptyset$; $N \leftarrow \emptyset$; $H \leftarrow \emptyset$; $h \leftarrow 1$
loop
  $E \leftarrow \mathrm{NF}_{\prec}(h, G) + \sum_{t \in B} \lambda_t \mathrm{NF}_{\prec}(t, G)$
  if $E = 0$ を満たす $\lambda_t = a_t \in K$ $(t \in B)$ が存在する **then**
    $G_1 \leftarrow G_1 \cup \{h + \sum_{t \in B} a_t t\}$
    $H \leftarrow H \cup \{h\}$
  else
    $B \leftarrow B \cup \{h\}$
    $N \leftarrow N \cup \{x_1 h, \ldots, x_m h\}$
  end if
  $N \leftarrow N \cap \{t \mid t$ は単項式で, すべての $s \in H$ に対し $s \nmid t\}$
  if $N = \emptyset$ **then**
    return G_1
  else
    $h \leftarrow N$ 中で $\prec_1$ に関して最小の単項式
    $N \leftarrow N \setminus \{h\}$
  end if
end loop

となる. $c_s(\lambda_t)$ は λ_t の一次式なので, $E=0$ をみたす $\lambda_t = a_t$ を探すことは線形方程式系の求解に帰着される.

5.3.3 FGLM アルゴリズムの計算量

FGLM における主な計算は, $\mathrm{NF}_{\prec}(t,G)$ の計算と, $f_h|_{\lambda_t=a_t} \in I$ をみたす $\lambda_t = a_t$ が存在するかどうか線形方程式系を解いて調べる計算である. これ以外の手間は, 単項式のリスト操作などであり無視できる. 以下で $\dim_K R/I$ を γ とおく. γ は I の零点の個数と等しい.

- $\mathrm{NF}_{\prec}(h,G)$ の計算

  $h \neq 1$ のとき $h = x_i h'$ と書けるので, $\mathrm{NF}_{\prec}(h,G) = \mathrm{NF}_{\prec}(x_i \mathrm{NF}(h',G),G)$ により, x_i 倍写像 $f \mapsto \mathrm{NF}_{\prec}(x_i f,G)$ の, R/I の線形空間としての基底 S に関する表現行列を求めておけば, 一つの $\mathrm{NF}_{\prec}(h,G)$ は, 既に求めてあるはずの $\mathrm{NF}_{\prec}(h',G)$ から手間 γ^2 で計算できる. この表現行列の計算は, γ 個の単項式 $s \in S$ に関する $\mathrm{NF}_{\prec}(x_i s,G)$ の計算であるが, これを既に計算してある値を再利用しながら行うことで, x_i 倍写像 $(i=1,\ldots,m)$ の表現行列を $O(m\gamma^3)$ で行うことができる.

- 線形方程式の求解

  各ステップにおける λ_t の線形方程式系は高々 $O(\gamma^3)$ で解けるが, それを単純にループの回数 ($O(m\gamma)$ であることが示される) だけ繰り返すと $O(m\gamma^4)$ となってしまう. しかし, 各 $\mathrm{NF}_{\prec}(t,G)$ たちを $s \in S$ の一次式として三角化したものに置き換えて保持しておけば, 新たな $\mathrm{NF}_{\prec}(h,G)$ に対し $E=0$ となる $\lambda_t = a_t$ が存在するかどうかは, この三角基底による剰余計算により判定でき, 1 ステップ $O(\gamma^2)$ となる. さらに, この剰余が 0 でないとき, この剰余を付け加えても三角基底という性質は保たれる. よって, ループの回数と合わせて, 線形方程式求解で必要となる手間は $O(m\gamma^3)$ である.

以上により, FGLM アルゴリズムの計算量は $O(m\gamma^3)$ となる.

Summation polynomial から構成される代数方程式系の場合, 変数は $z_{i,j}$ $(i=1,\ldots,m,\ j=1,\ldots,n')$ の mn' 個であり, 各 $z_{i,j}$ に対し $z_{i,j}^q - z_{i,j} = 0$ がイデアルの生成系に入っているので, 解の個数は高々 $q^{mn'}$ 個である. よって γ は高々 $q^{mn'}$ となり, FGLM による辞書式順序グレブナー基底への項順序変換の最悪計算量は $O(mn'q^{3mn'})$ となる. しかし, ECDLP の場合は一般的に γ は計算量的に無視できるほどに小さいことが知られている[14]. 従って,

[14] この γ は連立代数方程式の解の個数と等しいことから, ECDLP を指数計算法で解く場合は, γ が大きいほど得られる relation の個数が増加する. これは指数計算法で連立代数方程式を解く回数が減ることにつながる. しかし, 現時点では γ を増加させて計算効率を上げるアルゴリズムは発表されていない.

ECDLP を指数計算法で解く場合, F_4-style のアルゴリズムで必要とされる計算コストに比べて, FGLM のそれは無視できるほどに小さい.

6 ECDLP に関する指数計算法の研究動向

ECDLP に関する指数計算法の研究動向については [15] にまとめられており, 特にその 10.2 節では $E(\mathbb{F}_{2^n})$ 上の ECDLP を準指数時間で解く指数計算法の実現可能性について述べられている. その議論のキーワードとなっているのが first fall degree assumption である. この節では first fall degree assumption について説明する. また, 素体 $\mathbb{F}_p$ における ECDLP への指数計算法の適用 [23] についても述べる.

6.1 First fall degree assumption (FFDA)

2012 年, Petit と Quisquater は first fall degree assumption (FFDA) とよばれる仮定を導入することで, $E(\mathbb{F}_{2^n})$ 上の ECDLP を指数計算法で解くために必要な計算量が準指数時間 $O(2^{C' n^{2/3} \log n})$ であることを示した [24]. 但し C' は 2 未満の定数とする. さらに, 拡大次数 n がおよそ 2000 より大きい場合は, 指数計算法の計算コストは generic algorithm のそれより小さくなることを示した. しかし ECDLP における first fall degree assumption の妥当性については議論が分かれている [15]. この節では first fall degree assumption に関する近年の研究動向について述べる.

6.1.1 First fall degree assumption (FFDA) を仮定した計算量評価

ECDLP に対する指数計算法では一般的に FGLM の計算量は F_4-style の計算量 C_{F_4} より小さいため, point decomposition の計算量 C_{dcmp} は C_{F_4} で評価される. さらに C_{F_4} は degree of regularity D_{reg} で決定される.

この D_{reg} を近似する値として, Petit と Quisquater は first fall degree D_{first} を導入した (この節でも $X = \{x_1, \cdots, x_m\}$ であることに注意.):

定義 6.1. 多項式環 $R := \mathbb{F}_{q^n}[X]$ に対して $F := \{f_1, \cdots, f_k\} \subset R$ とする. ある $h_1, \cdots, h_k \in R$ に対して, D_{first} が以下の条件を満たす最小の次数であるとき, D_{first} を F の first fall degree とよぶ:

- $\sum_{i=1}^{k} h_i f_i \neq 0$,
- $\deg(\sum_{i=1}^{k} h_i f_i) < D_{\mathrm{first}}$,
- $D_{\mathrm{first}} = \max_i(\deg(f_i) + \deg(h_i))$.

(この節でも $Z' = \{z_{1,1}, \cdots, z_{m,n'}\}$ であることに注意.) さらに, first fall degree に対して以下の仮定を導入した:

仮定 6.2. $f \in \mathbb{F}_{2^n}[X]$ は各変数 x_i に対して $\deg_{x_i} f \leq 2^t - 1$ を満たすとする. $\mathbb{F}_{2^n}$ を $\mathbb{F}_2$-ベクトル空間としてみたときの部分ベクトル空間 V の次元を n' とする. V を利用した f への Weil descent で生成される連立代数方程式を構成する多項式集合を $F_f \subset \mathbb{F}_2[Z']$ とし, さらに F_f に全ての $z_{i,j}^2 - z_{i,j}$ を加えた集合を $F_{f,\mathrm{Frob}}$ とする. このとき $F_{f,\mathrm{Frob}}$ の D_{reg} ついて以下が成り立つ:

$$D_{\mathrm{reg}} \approx D_{\mathrm{first}}.$$

Summation polynomial S_{m+1} の x_m に $r \in \mathbb{F}_{2^n}$ を代入した多項式 $S_{m+1}|_{x_{m+1}=r}$ は仮定 6.2 の f の条件を満たす. そのためさらに以下の仮定を導入している:

仮定 6.3. f が summation polynomial から生成された多項式であっても仮定 6.2 は成り立つ.

仮定 6.2 の f を $S_{m+1}|_{x_{m+1}=r}$ に対応させたときの $F_{f,\mathrm{Frob}}$ を $F_{S_{m+1},\mathrm{Frob}}$ とする. このとき, Petit と Quisquater は $F_{S_{m+1},\mathrm{Frob}}$ で与えらえる連立代数方程式を解くことに適した方法として, ブロックグレブナー基底アルゴリズムを主張しており, 仮定 6.3 のもとでそれを解くために必要な計算量 C_{dcmp} は $O((n')^{\omega D_{\mathrm{first}}})$ で, $D_{\mathrm{first}} \approx m^2$ と見積もっている:

$$C_{\mathrm{dcmp}} = O((n')^{\omega m^2}). \tag{28}$$

ただし, $2 < \omega \leq 3$ は実行可能行列乗算指数とする.

(24) に (28) を代入して $E(\mathbb{F}_{2^n})$ 上の ECDLP を解くために必要な計算量 $C_{\mathrm{total}}(\mathbb{F}_{2^n})$ を評価する:

$$O(2^{m^2} + 2^{n'} m!(n')^{\omega m^2} + 2^{n'\omega}). \tag{29}$$

ここで $1/2 < \alpha < 1$ に対して $n' = n^\alpha$, $m = n^{1-\alpha}$ とする. このとき

$$m! \approx n^{1-\alpha} \log n^{1-\alpha} \tag{30}$$

が成り立つ [14]. 従って, (29), (30) より

$$\begin{aligned} C_{\mathrm{total}}(\mathbb{F}_{2^n}) &= O(2^{t_1} + 2^{t_3} + 2^{t_2}), \qquad (31)\\ t_1 &:= n^{2(1-\alpha)}, \\ t_2 &:= n^\alpha + n^{1-\alpha}(1-\alpha)\log n + \omega n^{2(1-\alpha)} \alpha \log n, \\ t_3 &:= \omega n^\alpha. \end{aligned}$$

よって最適化することで, (31) で $\alpha = 2/3$ を代入することにより以下を得る:

$$C_{\mathrm{total}}(\mathbb{F}_{2^n}) = O(2^{Cn^{2/3} \log n})$$

ただし定数 C は $C < 2$ を満たす.

6.1.2 First fall degree assumption (FFDA) の妥当性

仮定 6.2, 6.3 は first fall degree assumption (FFDA) とよばれ, いくつかの文献では FFDA を支持する結果や, それを利用して $\mathbb{F}_{2^n}$ 上の ECDLP を解くために必要な計算量を見積もっている [19], [24], [27]. しかしそれらの文献において FFDA は証明されていない. さらにこれらの文献の数値実験で扱われた有限体 $\mathbb{F}_{2^n}$ の拡大次数 n の大きさは, [19] では $n = 26$, [24] では $n = 20$, [27] では $n = 40$ までとなっており, generic algorithm による ECDLP に関する計算の記録 (表 1) に比べてずっと小さい.

FFDA が成り立たない場合は存在する. 例えば多項式集合 $F_1 \subset \mathbb{F}[x_1, x_2]$ と $F_2 \subset \mathbb{F}[x_3, x_4]$ が与えられたとして, F_1, F_2 の degree of regularity をそれぞれ $D_{\mathrm{reg},1}$, $D_{\mathrm{reg},2}$ とし, 同様にそれぞれの first fall degree を $D_{\mathrm{first},1}$, $D_{\mathrm{first},2}$ とする. さらに以下が成り立つとする:

$$D_{\mathrm{first},1} \approx D_{\mathrm{reg},1} << D_{\mathrm{reg},2} \approx D_{\mathrm{first},2}.$$

これは F_1, F_2 において FFDA が成り立っていることを意味する. しかし $F = F_1 \cup F_2 \subset \mathbb{F}[x_1, \cdots, x_4]$ について考えたとき, F の degree of regularity は $D_{\mathrm{reg},2}$ であり, また first fall degree は $D_{\mathrm{first},1}$ であるため FFDA は成り立たない.

FFDA の正当性に疑問を示す結果が存在する. 例えば [17] では, いくつかの $n \leq 40$ に対して $E(\mathbb{F}_{2^n})$ 上の ECDLP を S_3 を利用して解く実験を行っており, D_{first} と D_{reg} の差は n に依存する実験結果を与えた. 即ち, これは FFDA が成り立たないことを主張している.

上述のように FFDA は成り立つことも成り立たないことも厳密にはまだ証明されておらず, また十分大きな拡大体での数値実験の検証も実行されていない. しかし, summation polynomial と Weil descent を利用した ECDLP に関する指数計算法によって, 十分大きな拡大体での数値実験が現時点で成功していないことを考慮すると, FFDA は有効でないと推測される [15].

6.2 素体 $\mathbb{F}_p$ に関する ECDLP と指数計算法

ECDLP だけではなく, (1) のような一般の DLP を解く場合において, 巡回群 G の位数が小さな素数の積で表される場合, 即ち相異なる素数 p_i によって

$$\#G = \prod_{i=1}^{k} p_i^{e_i}$$

のように表されるとき, この G 上の DLP は Pohlig-Hellman のアルゴリズムで

$$O\left(\sum_{i=1}^{k}(e_i(\log \#G + \sqrt{p_i}))\right)$$

回の群演算で解かれることが知られている. 従って, 暗号で利用する巡回群の位数は素数となるように設定している.

2016 年, Petit らは素体上の楕円曲線 $E(\mathbb{F}_p)$ における ECDLP を解く場合に, 巡回群の位数ではなく, 標数 p について以下の条件が成り立つときに有効と思われる指数計算法の因子基底の設定を提案した [23]:

$$p-1 =: ST,\ T := \prod_{j=1}^{k} p_j \approx p^{1/m}.$$

ただし, p_i は与えられた定数 B 以下の素数とし[15], m は利用する summation polynomal S_{m+1} で与えられるとする. $\mathbb{F}_p^*$ の乗法部分群で位数が T であるものを V として因子基底 F を次のように設定する:

$$F = \{\pi_\ell \in E(\mathbb{F}_p) \mid x(\pi_\ell) \in V\}.$$

このとき, $x(\pi_\ell)$ は $\mathbb{F}_p$ における

$$L(x) = 1 - x^T \tag{32}$$

の根である. この L は以下の関数の合成関数として表すことができる,

$$\begin{aligned} L_j(x) &= x^{p_j}\ (j = 1, \cdots, k-1), \\ L_k(x) &= 1 - x^{p_k}, \\ L(x) &= (L_k \circ \cdots \circ L_1)(x). \end{aligned}$$

このとき以下の多項式で与えられる連立代数方程式を解くことで relation を得ることができる:

$$\begin{aligned} 0 &= S_{m+1}(x_{1,1}, \cdots, x_{m,1}, x(\mathcal{R})),\ (\mathcal{R} = [a]\mathcal{P} + [b]\mathcal{T} \in E(\mathbb{F}_p)), \\ x_{i,j+1} &= L_j(x_{i,j})\ (i = 1, \cdots, m; j = 1, \cdots, k-1), \\ 0 &= L_k(x_{i,k})\ (i = 1, \cdots, m). \end{aligned}$$

この提案方法の計算量の評価は与えられていない. また [23] ではこの提案方法を基にしたいくつかの工夫について述べられているが, 20-bit 程度の大きさの p に対する実験結果が報告されているだけで, 現時点では楕円曲線暗号の脅威とはなっていない.

7 まとめ

本稿では研究が近年盛んに行なわれている, summation polynomial と Weil descent を利用した, ECDLP に関する指数計算法について議論した. その内

[15] $i \neq i'$ に対して $p_i \neq p_{i'}$ である必要はない.

容は, サーベイ論文 [15] を基に, この種の計算方法の概要を整理したものである. [15] では連立代数方程式を解く方法に関する記述が少ないが, この部分は first fall degree assumption などの理解に必要な部分であるため, その主な計算方法として F_4-style のアルゴリズムと FGLM を組み合わせた方法に関する節を設けた. また [15] が公開された後に発表された, 素体上の楕円曲線 $E(\mathbb{F}_p)$ 上の ECDLP を考慮した指数計算法 [23] についても説明した.

第 6 節で述べたように $E(\mathbb{F}_{2^n})$ 上の ECDLP を解く指数計算法で, その計算量が準指数時間になると主張している文献がいくつか存在する. しかし first fall degree assumption など, 利用している仮定の正当性は必ずしも保証されているとは限らない. Generic algorithm と ECDLP を解く指数計算法の比較で重要なのは, [15] でも述べられているように, 現時点で実際にどれくらいの大きさの有限体における ECDLP が解けているかである. 指数計算法の場合は, 限られた小さな有限体上の楕円曲線における実験しか報告例がないことから, 現時点では ECDLP を利用した暗号の安全性は generic algorithm の計算量によって評価されるべきである. また, 十分大きな有限体上における ECDLP を指数計算法で解く場合に, 第 5.2.1 節で述べたように, F_4-style のアルゴリズムが膨大な量のメモリを必要とすることが障害となっている. このことを, ECDLP に関する指数計算法が現時点で有効でない一因として挙げる. しかしながら, ECDLP に関する指数計算法の研究動向は今後も注視する必要がある.

参考文献

[1] D. J. Bernstein, S. Engels, T. Lange, R. Niederhagen, C. Paar, P. Schwabe, and R. Zimmermann. Faster discrete logarithms on fpgas. *IACR Cryptology ePrint Archive*, Vol. 2016, p. 382, 2016.

[2] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. *IJACT*, Vol. 2, No. 3, pp. 212–228, 2012.

[3] Certicom Research. Certicom ECC challenge (latest update: November 10, 2009). https://www.certicom.com/images/pdfs/challenge-2009.pdf, 2009.

[4] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[5] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, 2005.

[6] D. A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Springer, 1998.

[7] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective (2nd Edition)*. Springer, 2005.

[8] CRYPTREC. CRYPTREC Report 2014, 2014. `http://www.cryptrec.go.jp/report/c14\_eval\_web.pdf`.

[9] C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, Vol. 147, pp. 75–104, 2011.

[10] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure and Applied Algebra*, Vol. 139, No. 1–3, pp. 61–88, 1999.

[11] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *ISSAC 2002, Proceedings*, pp. 75–83, 2002.

[12] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Using symmetries in the index calculus for elliptic curves discrete logarithm. *J. Cryptology*, Vol. 27, No. 4, pp. 595–635, 2014.

[13] J.-C. Faugère, P. M. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *J. Symb. Comput.*, Vol. 16, No. 4, pp. 329–344, 1993.

[14] J.-C. Faugère, L. Perret, C. Petit, and G. Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In *EUROCRYPT 2012, Proceedings*, pp. 27–44, 2012.

[15] S. D. Galbraith and P. Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Des. Codes Cryptography*, Vol. 78, No. 1, pp. 51–72, 2016.

[16] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *J. Symb. Comput.*, Vol. 44, No. 12, pp. 1690–1702, 2009.

[17] M.-D. A. Huang, M. Kosters, and S. L. Yeo. Last fall degree, hfe, and weil descent attacks on ECDLP. In *CRYPTO 2015, Proceedings, Part I*, pp. 581–600, 2015.

[18] T. Izu. Current status on solving ECDLP. In *SCIS 2017, Proceedings*, 2017.

[19] K. Karabina. Point decomposition problem in binary elliptic curves. *IACR Cryptology ePrint Archive*, 2015. `http://eprint.iacr.org/2015/319`.

[20] M. Kreuzer and L. Robbiano. *Computational Commutative Algebra 2*. Spromger, 2005.

[21] D. Lazard. Gröbner-bases, gaussian elimination and resolution of systems of algebraic equations. In *EUROCAL '83, Proceedings*, pp. 146–156, 1983.

[22] E. W. Mayr and S. Ritscher. Dimension-dependent bounds for gröbner bases of polynomial ideals. *J. Symb. Comput.*, Vol. 49, pp. 78–94, 2013.

[23] C. Petit, M. Kosters, and A. Messeng. Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields. In *PKC 2016, Proceedings, Part II*, pp. 3–18, 2016.

[24] C. Petit and J.J. Quisquater. On polynomial systems arising from a Weil descent. In *ASIACRYPT 2012, Proceedings*, pp. 451–466, 2012.

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

[26] I. A. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. *IACR Cryptology ePrint Archive*, 2004. `http://eprint.iacr.org/2004/031`.

[27] I. A. Semaev. New algorithm for the discrete logarithm problem on elliptic curves. *IACR Cryptology ePrint Archive*, 2015. `http://eprint.iacr.org/2015/310`.

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

[29] E. Wenger and P. Wolfger. Solving the discrete logarithm of a 113-bit koblitz curve with an fpga cluster. In *SAC 2014, Proceedings*, pp. 363–379, 2014.

[30] E. Wenger and P. Wolfger. Harder, better, faster, stronger: elliptic curve discrete logarithm computations on fpgas. *J. Cryptographic Engineering*, Vol. 6, No. 4, pp. 287–297, 2016.