Bit Coincidence Mining Algorithm (Draft)

Koh-ichi Nagao (nagao@kanto-gakuin.ac.jp)

Faculty of Science and Engineering, Kanto Gakuin Univ.,

Abstract. Here, we propose new algorithm for solving ECDLP named "Bit Coincidence Mining Algorithm!", from which ECDLP is reduced to solving some quadratic equations system. In this algorithm, ECDLP of an elliptic curve E defined over $\mathbb{F}_q$ (q is prime or power of primes) reduces to solving quadratic equations system of $d-1$ variables and $d+C_0-1$ equations where C_0 is small natural number and $d \sim C_0 \log_2 q$. This equations system is too large and it can not be solved by computer. However, we can show theoritically the cost for solving this equations system by xL algorithm is subexponential under the reasonable assumption of xL algorithm.

1 Introduction and Notation

Let q be a power of prime and

$$E/\mathbb{F}_q : y^2 + \tilde{a_1}xy + \tilde{a_3}y - x^3 - \tilde{a_2}x^2 - \tilde{a_4}x - \tilde{a_6} = 0$$

be an elliptic curve. Here, we mainly consider the case q being large prime.

Problem 1 ((ECDLP)) *Let $P, Q \in E(\mathbb{F}_q)$ such that $< P > \ni Q$. ECDLP is the problem finding integer N satisfying $Q = NP$.*

Here, we propose a new algorithm solving this problem, named "Bit Coincidence Mining Algorithm"! From this algorithm, ECDLP reduced to solving equations system which consists of d quadratic equations over $\mathbb{F}_q$ of $d-1$ variable where $d = \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$. Let C_0 be a constant or the small value in $O((\log_2 \#E(\mathbb{F}_q))^\alpha)$ for some $0 \leq \alpha \leq 1$. This algorithm can be generalized that ECDLP is essentially reduced to to solving larger equations system which consists of $d + C_0 - 1$ quadratic equations over $\mathbb{F}_q$ of $d-1$ variable where $d = \lfloor \log_2 q^{C_0 - 1} \#E(\mathbb{F}_q) \rfloor + 1 \sim C_0 \log_2 q$. From the observation of Courtois et al., solving this equations system by xL algorithm is subexponential. And so, under the assumption of the complexity of xL algorithm, ECDLP is also subexponential. It is only the theoretical result and since the equations system is very large, it is not computable.

Moreover, we show the computable version of this algorithm. Suppose d'bit of binary expansion of N is known. So, ECDLP reduced to solving equations system which consists of d quadratic equations over $\mathbb{F}_q$ of $d-1$ variable where $d = \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1 - d'$. Considering the case that d' is large, obtained equations system is small and so we have easily obtain the solution of ECDLP.

2 Riemann Roch space $L((d+1)\infty - P_0)$

Let $P_0, P_1, .., P_d$ be the $d+1$ point in $E(\mathbb{F}_q)\backslash\{\infty\}$. In this section, we further fix P_0. Put $(x_i, y_i) := P_i$. Then the space of function field $L((d+1)\infty - P_0)$, which means the set of the elements of function field that has pole only at ∞, the order of the pole at ∞ is $\leq d+1$ and has zero at P_0, is spanned by

$$(x - x_0), (x - x_0)x, ..., (x - x_0)x^{\lfloor (d-1)/2 \rfloor}, (y - y_0), (y - y_0)x, ..., (y - y_0)x^{\lfloor (d-2)/2 \rfloor}.$$

Let $A_1, ..., A_{d-1}$ be variables and $\overrightarrow{A} := (A_1, ..., A_{d-1})$. Put $\phi_{1,\overrightarrow{A}}(x), \phi_{2,\overrightarrow{A}}(x) (\in \mathbb{F}_q[A_1, ..., A_{d-1}, x])$ as follows;

1) In the case d is odd,

$$\phi_{1,\overrightarrow{A}}(x) := \sum_{i=1}^{(d-1)/2} A_i x^{i-1}, \quad \phi_{2,\overrightarrow{A}}(x) := \sum_{i=(d+1)/2}^{d-1} A_i x^{i-(d+1)/2} + x^{(d-1)/2}$$

2) In the case d is even,

$$\phi_{2,\overrightarrow{A}}(x) := \sum_{i=1}^{d/2} A_i x^{i-1}, \quad \phi_{1,\overrightarrow{A}}(x) := \sum_{i=d/2+1}^{d-1} A_i x^{i-d/2-1} + x^{d/2-1}.$$

Also put

$$\phi_{\overrightarrow{A}}(x,y) := (y-y_0)\phi_{1,\overrightarrow{A}}(x) - (x-x_0)\phi_{2,\overrightarrow{A}}(x) (\in \mathbb{F}_q[A_1,...,A_{d-1},x,y]),$$

$$\phi_{3,\overrightarrow{A}}(x) := y_0\phi_{1,\overrightarrow{A}}(x) + (x-x_0)\phi_{2,\overrightarrow{A}}(x) (\in \mathbb{F}_q[A_1,...,A_{d-1},x]).$$

Note that $\phi_{\overrightarrow{A}}(x,y) = y\phi_{1,\overrightarrow{A}}(x) - \phi_{3,\overrightarrow{A}}(x)$.

Lemma 1. *Let $f \in L((d+1)\infty - P_0)$. Suppose $ord_\infty f = -d-1$, there exists $\overrightarrow{a} = (a_1,...,a_{d-1}) \in \mathbb{A}^{d-1}(\overline{\mathbb{F}_q})$, such that $f = \phi_{\overrightarrow{a}}(x,y)$.*

Remark Suppose $ord_\infty f = -d-1$, $div(f)$ is written by $P_0+P_1+...+P_d-(d+1)\infty$, where the property $\{P_0,...,P_d\} \cap \{-P_0,...,-P_d\} = \emptyset$ holds.

Lemma 2. *Let $f \in L((d+1)\infty - P_0)$. Suppose $div(f)$ is written by the form $P_0+P_1+...+P_d-(d+1)\infty$, where $P_0,...,P_d$ is some element in $E(\mathbb{F}_q)\backslash\{\infty\}$ satisfying $\{P_0,...,P_d\} \cap \{-P_0,...,-P_d\} = \emptyset$, then, there exists $\overrightarrow{a} = (a_1,...,a_{d-1}) \in \mathbb{A}^{d-1}(\mathbb{F}_q)$, such that $f = \phi_{\overrightarrow{a}}(x,y)$.*

Put

$$f_{\overrightarrow{A}}(x) := \phi^2_{3,\overrightarrow{A}}(x) + \tilde{a_1}x\phi_{1,\overrightarrow{A}}(x)\phi_{3,\overrightarrow{A}}(x) + \tilde{a_3}\phi_{1,\overrightarrow{A}}(x)\phi_{3,\overrightarrow{A}}(x) - \phi_{1,\overrightarrow{A}}(x)^2(x^3 + \tilde{a_2}x^2 + \tilde{a_4}x + \tilde{a_6})).$$

Then we see $\deg_x f_{\overrightarrow{A}}(x) = d+1$ and $(x-x_0) \mid f_{\overrightarrow{A}}(x)$.

Put

$$g_{\overrightarrow{A}}(x) := \begin{cases} f_{\overrightarrow{A}}(x)/(x-x_0) & (d \text{ is odd}) \\ -f_{\overrightarrow{A}}(x)/(x-x_0) & (d \text{ is even}) \end{cases}.$$

Then $g_{\overrightarrow{A}}(x)$ is monic degree d polynomial associate to variable x. Let $B_{d-1},...,B_0 \in \mathbb{F}_q[A_1,...,A_{d-1}]$ by

$$g_{\overrightarrow{A}}(x) = x^d + B_{d-1}x^{d-1} + ... + B_1x + B_0.$$

Then we see that B_i $(i = 0,...,d-1)$ are quadratic polynomials in $\mathbb{F}_q[A_1,...,A_{d-1}]$.

Lemma 3. *Suppose there exists some $P_1,...,P_d \in E(\overline{\mathbb{F}}_q)\backslash\{\infty\}$ such that*
1) $P_0+P_1+...+P_d = 0$
2) $\{P_0,...,P_d\} \cap \{-P_0,...,-P_d\} = \emptyset$.
Then we have
there exists some $\overrightarrow{a} = (a_1,...,a_{d-1}) \in \mathbb{A}^{d-1}(\overline{\mathbb{F}}_q)$, such that $g_{\overrightarrow{a}}(x) = \prod_{i=1}^d (x - x(P_i)))$.

Proof. From the condition 2), $\phi(x,y) \in \overline{\mathbb{F}}_q(E)$ such that $P_0+P_1+...+P_d-(d+1)\infty = div(\phi(x,y))$, has the property $\mathrm{ord}_\infty\phi(x,y) = -d-1$. Then from the definition of $\phi_{\overrightarrow{A}}(x,y)$, we see the existence of $\overrightarrow{a} = (a_1,...,a_{d-1}) \in \mathbb{A}^{d-1}(\overline{\mathbb{F}}_q)$, such that $\phi(x,y)$ is written by $\phi_{\overrightarrow{a}}(x,y)$.

Proposition 1. *Let* $x_1, ..., x_d \in \{x(P) \in \mathbb{F}_q \mid P \in E(\mathbb{F}_q)\backslash\{\infty\}\}$. *Suppose there exists some* $\overrightarrow{a} = (a_1, ..., a_{d-1}) \in \mathbb{A}^{d-1}(\overline{\mathbb{F}}_q)$ *such that* $g_{\overrightarrow{a}}(x) = \prod_{i=1}^{d}(x - x_i)$.
Then we have
1) there exists some $P_1, ..., P_d \in E(\mathbb{F}_q)\backslash\{\infty\}$, *satisfying*
1-a) $x(P_i) = x_i$ $(i = 1, ..., d)$,
1-b) $P_0 + P_1 + ... + P_d = 0$, *and*
1-c) $\{P_0, ..., P_d\} \cap \{-P_0, ..., -P_d\} = \emptyset$.
2) $\overrightarrow{a} = (a_1, ..., a_{d-1}) \in \mathbb{A}^{d-1}(\mathbb{F}_q)$.

Proof. From the definition of $\phi_{\overrightarrow{A}}(x, y)$, $\phi_{\overrightarrow{a}} = \phi_{\overrightarrow{a}}(x, y)$ satisfies $div(\phi_{\overrightarrow{a}}) = P_0 + ... + P_d - (d+1)\infty$, $ord_\infty \phi_{\overrightarrow{a}}(x, y) = -d-1$. Then we have 1-a),1-b) and 1-3). Since $P_0, P_1, ..., P_d\infty \in E(\mathbb{F}_q)$, we see $\phi_{\overrightarrow{a}}(x, y) \in \mathbb{F}_q(E)$. Thus we have 2).

Definition 1 *Let* $P_0, ..., P_d \in E(\mathbb{F}_q)\backslash\{\infty\}$. *Put* $c_i (\in \mathbb{F}_q)$ $(i = 0, ..., d-1)$ *by*

$$x^d + \sum_{i=0}^{d-1} c_i x^i = \prod_{i=1}^{d}(x - x(P_i)).$$

We define the equation system $EQS1(P_0, P_1, ..., P_d)$ *consists of* $d-1$ *variable* $A_1, ..., A_{d-1}$ *and* d *quadratic equations* $B_0 = c_0$, $B_1 = c_1$,...,$B_{d-1} = c_{d-1}$.

From this definition, Proposition 1 can be expressed as follows

Theorem 1. *Let* $P_0, ..., P_d \in E(\mathbb{F}_q)\backslash\{\infty\}$. *Suppose* $EQS1(P_0, ..., P_d)$ *has solution* $A_1 = a_1, ..., A_{d-1} = a_{d-1}$ *in* $\overline{\mathbb{F}}_q$. *Then we have*
1) There exists some $l'_i \in \{1, -1\}$ *such that* $P_0 + \sum_{i=1}^{d} l'_i P_i = 0$,
2) $(a_1, ..., a_{d-1}) \in \mathbb{A}^{d-1}(\mathbb{F}_q)$.

Remark l'_is can be computed as follows:
Put $x_i, y_i, y'_i \in \mathbb{F}_q$ by $(x_i, y_i) = P_i$, $(x_i, y'_i) = -P_i$ and put $\overrightarrow{a} = (a_1, ..., a_{d-1})$. Then

$$l'_i := \begin{cases} 1 & \phi_{\overrightarrow{a}}(x_i, y_i) = 0 \\ -1 & \text{otherwise (in the case } \phi_{\overrightarrow{a}}(x_i, y'_i) = 0 \text{ holds)} \end{cases}.$$

3 ECDLP

Here, we give the algorithm for solving discrete logarithm problem for elliptic curve.

For a technical reason, we consider the group $E[2] \cap < P >$ and consider the following "Modified ECDLP". Let $2^k \cdot m$ (m being odd natural number) be the order of $E(\mathbb{F}_q)$. In the case $mP = 0$, $E[2] \cap < P >= \{\infty\}$ and otherwise, $E[2] \cap < P >$ is written by the form $\{T_0, \infty\}$ for some 2-torsion point T_0. Let k_0 be the (minimal) natural number such that $(2^{k_0}m)P \neq 0$ and $(2^{k_0+1}m)P = 0$. Then T_0 $(\in E[2])$ is expressed by $T_0 = (2^{k_0}m)P$.

Problem 2 ((Modifyied ECDLP)) *Let* $E/\mathbb{F}_q$ *be an elliptic curve and let* $P, Q \in E(\mathbb{F}_q)$ *such that* $< P > \ni Q$. *We call the problem finding* N *such that* $Q - NP \in E[2] \cap < P >$ *by "Modified ECDLP".*

In the case $E[2] \cap < P >= \{\infty\}$, "Modified ECDLP" is the same as "ECDLP". Otherwise, suppose we have the relation $Q - NP = T_0 \in \{T_0, \infty\} = E[2] \cap < P >$, we have $Q = (N + 2^{k_0}m)P$ and ECDLP is solved. Here, we remark that "Modified ECDLP" is essentially the same as "ECDLP".

Now, we will solve "Modified ECDLP".
Put $d = \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$, $P_i := 2^{i-1}P$ $(i = 1, 2, ..., d)$. Also put $P_0 := 2Q - \sum_{i=1}^{d} P_i$ and solving the equation system $EQS1(P_0, ..., P_d)$. Assume $EQS1$ has a solution. From the

discussion of previous section $l'_i \in \{\pm 1\}$ $(i = 1, ...d)$ satisfying $P_0 + \sum_{i=1}^{d}(l'_i P_i) = 0$ can be computable. Also put

$$l_i := (-l'_i + 1)/2 \ \in \{0, 1\}, \quad (i = 1, ..., d).$$

Since $2\sum l_i P_i = -\sum l'_i P_i + \sum P_i$, we have

$$0 = 2Q - \sum_{i=1}^{d} P_i + \sum_{i=1}^{d} l'_i P_i = 2(Q - \sum_{i=1}^{d} l_i P_i).$$

Thus we have

$$Q - \sum_{i=1}^{d} l_i P_i \in E[2] \cap < P > .$$

Assume $EQS1$ has no solution. we see similarly that such N does not exists. Thus we have the algorithm for solving "ECDLP".

Algorithm 1 Bit coincidence mining algorithm

Input: $E/\mathbb{F}_q$ elliptic curve, $P, Q \in E(\mathbb{F}_q)$ st. $< P > \ni Q$
Output: Integer N satisfying $NP = Q$
Put $d := \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$
Put $P_i := 2^{i-1}P$ $(i = 1, 2, ..., d)$
Put $P_0 := 2Q - \sum_{i=1}^{d} P_i$
Solve $EQS1(P_0, ..., P_d)$
if $EQS1(P_0, ..., P_d)$ has no solution **then**
  Such N does not exists and return **False**.
Compute $l'_i \in \{\pm 1\}$ $(i = 1, ..., d)$ st. $P_0 + \sum_{i=1}^{d} l'_i P_i = 0$ from the solution of $EQS1$
Put $l_i := (-l'_i + 1)/2 \ \in \{0, 1\}$ $(i = 1, ..., d)$
if $Q - \sum_{i=1}^{d} l_i P_i = 0$ **then**
  Return $\sum_{i=1}^{d} l_i 2^{i-1}$
else
  Compute $T_0 (\in E[2] \backslash \{\infty\} \cap < P >)$
  Compute Integer $\log_P T_0$ satisfying $(\log_P T_0)P = T_0$
  Return $\sum_{i=1}^{d} l_i 2^{i-1} + (\log_P T_0)$

Remark Algorithm 1 returns **False** only in the case that there are no N such that $Q - NP \in E[2] \cap < P >$.

Remark In Algorithm 1, the value d is taken $\lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$. So, general ECDLP reduced to solving equations system consists of $\lfloor \log_2 \#E(\mathbb{F}_q) \rfloor$ variables, $\lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$ quadratic equations over $\mathbb{F}_q$.

4 Toy Example

Here, we compute toy example. Let $E/\mathbb{F}_{1021} : y^2 = x^3 + x + 6$. We have $\#E(\mathbb{F}_{1021}) = 991$ and it is prime order. Let $P = (2, 4), Q = (101, 250) \in E(\mathbb{F}_{1021})$. Put $P_1 := P = (2, 4), P_2 := 2P = (557, 498), P_3 := 2^2 P = (93, 425), P_4 := 2^3 P = (629, 471), P_5 := 2^4 P = (632, 880), P_6 := 2^5 P = (660, 38), P_7 := 2^6 P = (182, 513), P_8 := 2^7 P = (27, 714), P_9 := 2^8 P = (549, 777), P_{10} := 2^9 P = (156, 982), P_0 := 2Q - \sum_{i=1}^{10} P_i = (662, 1016)$.

The polynomials $\phi_{1,\overrightarrow{A}}(x), \phi_{2,\overrightarrow{A}}(x), \phi_{3,\overrightarrow{A}}(x)$ in $\mathbb{F}_{1021}(A1, ..., A9, x)$ are written by the following;

$\phi_{1,\overrightarrow{A}}(x) = A6 + A7 * x + A8 * x^2 + A9 * x^3 + x^4,$
$\phi_{2,\overrightarrow{A}}(x) = A1 + A2 * x + A3 * x^2 + A4 * x^3 + A5 * x^4,$
$\phi_{3,\overrightarrow{A}}(x) = A1 * x + 359 * A1 + A2 * x^2 + 359 * A2 * x + A3 * x^3 + 359 * A3 * x^2 + A4 * x^4 +$
$359 * A4 * x^3 + A5 * x^5$
$+ 359 * A5 * x^4 + 1016 * A6 + 1016 * A7 * x + 1016 * A8 * x^2 + 1016 * A9 * x^3 + 1016 * x^4.$
Note that $\phi_{3,\overrightarrow{A}}(x)$ depends on the coordinates of P_0.

From this, we can compute $g_{\overrightarrow{A}}(x)$, $\prod_{i=1}^{10}(x - x(P_i))$ and the equations system $EQS1(P_0, ..., P_{10})$ (where A1,...,A9 are variables) is written by
$662 * A1^2 + 10 * A1 * A6 + 236 * A6^2 + 869 = 0,$
$1020 * A1^2 + 303 * A1 * A2 + 10 * A1 * A7 + 10 * A2 * A6 + 662 * A6^2 + 472 * A6 * A7 + 880 = 0,$
$1019 * A1 * A2 + 303 * A1 * A3 + 10 * A1 * A8 + 662 * A2^2 + 10 * A2 * A7 + 10 * A3 * A6 +$
$A6^2 + 303 * A6 * A7 + 472 * A6 * A8 + 236 * A7^2 + 543 = 0,$
$1019 * A1 * A3 + 303 * A1 * A4 + 10 * A1 * A9 + 1020 * A2^2 + 303 * A2 * A3 + 10 * A2 * A8 + 10 * A3 *$
$A7 + 10 * A4 * A6 + 2 * A6 * A7 + 303 * A6 * A8 + 472 * A6 * A9 + 662 * A7^2 + 472 * A7 * A8 + 228 = 0,$
$1019 * A1 * A4 + 303 * A1 * A5 + 10 * A1 + 1019 * A2 * A3 + 303 * A2 * A4 + 10 * A2 * A9 +$
$662 * A3^2 + 10 * A3 * A8 + 10 * A4 * A7 + 10 * A5 * A6 + 2 * A6 * A8 + 303 * A6 * A9 + 472 *$
$A6 + A7^2 + 303 * A7 * A8 + 472 * A7 * A9 + 236 * A8^2 + 574 = 0,$
$1019 * A1 * A5 + 1019 * A2 * A4 + 303 * A2 * A5 + 10 * A2 + 1020 * A3^2 + 303 * A3 * A4 + 10 *$
$A3 * A9 + 10 * A4 * A8 + 10 * A5 * A7 + 2 * A6 * A9 + 303 * A6 + 2 * A7 * A8 + 303 * A7 * A9 +$
$472 * A7 + 662 * A8^2 + 472 * A8 * A9 + 115 = 0,$
$1019 * A2 * A5 + 1019 * A3 * A4 + 303 * A3 * A5 + 10 * A3 + 662 * A4^2 + 10 * A4 * A9 + 10 * A5 *$
$A8 + 2 * A6 + 2 * A7 * A9 + 303 * A7 + A8^2 + 303 * A8 * A9 + 472 * A8 + 236 * A9^2 + 350 = 0,$
$1019 * A3 * A5 + 1020 * A4^2 + 303 * A4 * A5 + 10 * A4 + 10 * A5 * A9 + 2 * A7 + 2 * A8 * A9 +$
$303 * A8 + 662 * A9^2 + 472 * A9 + 132 = 0,$
$1019 * A4 * A5 + 662 * A5^2 + 10 * A5 + 2 * A8 + A9^2 + 303 * A9 + 650 = 0,$
$1020 * A5^2 + 2 * A9 + 65 = 0.$

Solving this equations system, we have a solution

$$(A1, ..., A9) = (372, 568, 115, 683, 111, 739, 673, 739, 2),$$

and we can recover

$$[l_1, l_2, ..., l_{10}] = [0, 1, 1, 0, 0, 0, 1, 0, 1, 1].$$

Put $n := \sum_{i=1}^{10} l_i 2^{i-1}$ and we obtain $n = 838$ which is the discrete logarithm. By a direct calculation, $838P = (101, 250) = Q$ and the result is checked to be true.

5 Complexity using Linearizion

First, we try to estimate the complexity for solving this equations system by xL algorithm [2]. In [2], Courtois et al. treat the only case that the equations are of the form "homogeneous quadratic polynomial=constant", but, one can obtain similar results if general quadratic equations are used.

Let $\mathcal{M}_d$ be the set of all monomials of variables $X_1, ..., X_n$ whose degree $\leq d$.

In [2], Courtois et al. observed as follows;

When $D = O(\sqrt{n})$ and $m \geq n$, the number of the equations obtained by "Multiply Step" is bigger than $\#\mathcal{M}_D$ and so, xL algorithm seems to be work. However, in the case $m = n$, simulation(maybe computer experiments cf [2]) shows the D that xL algorithm works well must be 2^n. (Reason is clear, since the equations system have generally 2^n solutions in $\overline{K}$.) In the case $m = n+1$, simulation(maybe computer experiments) shows the D that xL algorithm works well must be n (in stead of $\sqrt{n}$, Reason is not clear[1]). In the case $m = n + C_0$ (C_0 some small value), D that xL algorithm works well can be taken $O(\sqrt{n})$. The observation of the case $m = n + C_0$ is written by the following Assumption 1 or Assumption 2. First, we consider the strong assumption, that C_0 does not depend on the input value n and can be taken constant.

[1] I think it is a mystery!!!!

Algorithm 2 xL algorithm [2]

Notation: K field, $X_1, ..., X_n$ variables, $\overrightarrow{X} := (X_1, ..., X_n)$
$p_i(\overrightarrow{X}) \in K[X_1, ..., X_n]$ $(i = 1, ..., m)$ quadratic polynomials
$\mathcal{M}_d := \{$All monomials of $X_1, ..., X_n$ degree $\leq d\}$
Assumption: $n \leq m$
Input: $p_i(\overrightarrow{X})$ $(i = 1, ..., m)$
Output: $\overrightarrow{x} = (x_1, ..., x_n) \in \mathbb{A}^n(K)$ satisfying $p_i(\overrightarrow{x}) = 0$ $(i = 1, ..., m)$
Set parameter $D = D(n, m)$
Multiply:
for all $m(\overrightarrow{X}) \in \mathcal{M}_{D-2},\ p(\overrightarrow{X}) \in \{p_1(\overrightarrow{X}), ..., p_m(\overrightarrow{X})\}$ **do**
Genera all products $m(\overrightarrow{X})p(\overrightarrow{X})$
Linearize: Consider each monomial in $\mathcal{M}_D$ as new variable and perform Gaussian elimination on the equations obtained in "Multiply". The ordering on the monomial must be such that all the terms containing 1 variable (say X_1) are eliminated last
Solve: Assume that Linearize step yields at least one univariate equation in the powers of X_1, Solve this equation.
Repeat: Simplify the equations and repeat the process to find the values of the other variables.

Assumption 1 (Strong version) *There are positive constants C_0, C_1 satisfying the following:*
Suppose $m = n + C_0$ and $D = C_1\sqrt{n}$, xL algorithm returns the solution in high probability.

Assume Assumption 1 and $D = C_1\sqrt{n}$. we have
$\#\mathcal{M}_D = \binom{n+D}{D} \preceq n^{C_1\sqrt{n}} = O(exp(n^{1/2+o(1)}))$. (Many terms are absorbed into $o(1)$ term. Then $o(1)$ is Huge, although $\lim_{n\to\infty} o(1) = 0$.) In order for performing xL algorithm, the dominant part is Gaussian elimination of the matrix whose size is about $\#\mathcal{M}_D \times \#\mathcal{M}_D$. Its cost is $(\#\mathcal{M}_D)^w$ where $w \sim 2.7$ is the linear algebra constant and it is also written by $O(exp(n^{1/2+o(1)}))$.

Here, we prepare the weaker version of the assumption. [2]

Assumption 2 (Weaker version) *There are positive constants α, β, C_0, C_1 satisfying the following:*
1) $0 < \alpha, \beta < 1$, $(1 + \alpha)\beta < 1$,
2) Suppose $m = n + C_0 n^\alpha$ and $D = C_1 n^\beta$, xL algorithm returns the solution in high probability.

Assume Assumption 2 and $D = C_1 n^\beta$. Similarly, we have that the cost of xL algorithm is $O(exp(n^{(1+\alpha)\beta+o(1)}))$.

In our situation, general ECDLP reduces to solving equations system whose
$n = \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor, m = n + 1$ and so, the required $D = D(n, m)$ is n and the complexity of ECDLP is still exponential. In the next section, we modify the equations system and show ECDLP is subexponential under the Assumption 1 or Assumption 2.

Remark If one uses Gröbner basis computation for solving equations system, the maximal degree of the polynomials appears in the computation is generally larger than the that of xL. At least, from the complexity point of view, for solving low degree equations system, using xL seems to be better than using Gröbner basis.

6 Sub-exponential algorithm

Recall the situation we concern. Let $E/\mathbb{F}_q$ be an elliptic curve and $P, Q \in E(\mathbb{F}_q)$ such that $< P > \ni Q$. We consider the ECDLP which compute integer N satisfying $Q = NP$. Moreover,

[2] If C_0 depend on the input value n, we can show the subexponentiality of ECDLP under the Assumption 2.

we suppose Assumption 1. (We will concern the case $C_0 \geq 2$.)

Put $d := \lfloor \log_2(q^{C_0-1} \cdot \#E(\mathbb{F}_q)) \rfloor + 1$, and consider the $EQS1(P_0, ..., P_d)$, where $P_i = (2^{i-1})P$ $(i = 1, ..., d)$ and $P_0 = 2Q - \sum_{i=1}^{d} P_i$.

Note that $d \sim C_0 \log_2 q \sim C_0 \log_2 \#E(\mathbb{F}_q)$. From the discussion in §2, each solution of $EQS1$ corresponds to the $(l_1, ..., l_d) \in \{0,1\}^d$ such that $Q = \sum_{i=1}^{d} l_i P_i$ (or $Q = T_0 + \sum_{i=1}^{d} l_i P_i$ where $T_0 \in E(\mathbb{F}_q)[2] \cup < P >$). Since $\#\{0,1\}^d = 2^d \sim q^{C_0-1} \cdot \#E(\mathbb{F}_q)$, the number of $(l_1, ..., l_d) \in \{0,1\}^d$ satisfying $Q = \sum_{i=1}^{d} l_i P_i$. is estimated by q^{C_0-1}. Thus also the number of the solution $EQS1$ is estimated by $\geq q^{C_0-1}$.

Let $L_i = L_i(A_1, ..., A_{d-1})$ $(i = 1, ..., C_0 - 1)$ be the random degree 1 polynomials in $\mathbb{F}_q[A_1, ..., A_{d-1}]$ and consider the new equations system

$$EQS2(P_0, ..., P_d) := EQS1(P_0, ..., P_d) \cup \{L_i = 0 \,|\, i = 1, ..., C_0 - 1\}.$$

Since L_i is random degree 1 polynomial, so, for any $(a_1, .., a_{d-1}) \in \mathbb{A}^{d-1}(\mathbb{F}_q)$, the probability $L_i(a_1, ..., a_{d-1}) = 0$ is $1/q$ and the probability $L_i(a_1, ..., a_{d-1}) = 0$ holds for all $i \in [1, ..., C_0 - 1]$ is $1/q^{C_0-1}$. Thus the number of the solution $EQS2$ is estimated by ≥ 1 and from this solution, the value of ECDLP can be recovered.

Note that $EQS2$ consists of $n = d-1$ variables and $m = d+C_0-1$ quadratic equations (including $C_0 - 1$ linear equations), So from Assumption 1 and the discussion of the previous section, the cost of solving $EQS2$ is in $O(exp(d^{1/2+o(1)}))$. Since $d \sim C_0 \log_2 q \sim C_0 \log_2 \#E(\mathbb{F}_q)$ and C_0 is a constant, we have the following theorem;

Theorem 2. *Suppose the Assumption1, the complexity of ECDLP is estimated by* $O(exp((\log_2 \#E(\mathbb{F}_q))^{1/2+o(1)}))$ *where* $\lim_{\#E(\mathbb{F}_q)\to\infty} o(1) = 0$. [3]

Algorithm 3 Subexponential algorithm (strong assumption version)

Input: $E/\mathbb{F}_q$ elliptic curve, $P, Q \in E(\mathbb{F}_q)$ st. $< P > \ni Q$
Output: Integer N satisfying $NP = Q$

```
Assume "Assumption 1"
Set parameter C_0
Put d := ⌊q^{C_0-1} log_2 #E(F_q)⌋ + 1
Put P_i := 2^{i-1} P (i = 1, 2, ..., d)
Put P_0 := 2Q − Σ_{i=1}^{d} P_i
Solve EQS2(P_0, ..., P_d) (Note:Cost for solving is O(exp(q^{1/2+o(1)})))
if EQS2(P_0, ..., P_d) has no solution then
   return False.
Compute l'_i ∈ {±1} (i = 1, ..., d) st. P_0 + Σ_{i=1}^{d} l'_i P_i = 0 from the solution of EQS2
Put l_i := (−l'_i + 1)/2 ∈ {0, 1} (i = 1, ..., d)
if Q − Σ_{i=1}^{d} l_i P_i = 0 then
   Return Σ_{i=1}^{d} l_i 2^{i-1}
else
   Compute T_0(∈ E[2]\{∞} ∩ < P >)
   Compute Integer log_P T_0 satisfying (log_P T_0)P = T_0
   Return Σ_{i=1}^{d} l_i 2^{i-1} + (log_P T_0)
```

Similarly we can obtain the following theorem and Algorithm 4; [4]

Theorem 3. *Suppose the Assumption2, the complexity of ECDLP is estimated by* $O(exp((\log_2 \#E(\mathbb{F}_q))^{(1+\alpha)\beta+o(1)}))$ *where* $\lim_{\#E(\mathbb{F}_q)\to\infty} o(1) = 0$.

[3] The terms of C_0 can be absorbed into $o(1)$ term
[4] Using $C_0 n^\alpha$ instead of C_0, we have this result.

Algorithm 4 Subexponential algorithm (weak assumption version)

Input: $E/\mathbb{F}_q$ elliptic curve, $P, Q \in E(\mathbb{F}_q)$ st. $< P > \ni Q$
Output: Integer N satisfying $NP = Q$
Assume "Assumption 2"(Note α, β is the constant of this assumption)
Set parameter C_0
Put $d := \lfloor q^{C_0 n^{\alpha}-1} \log_2 \#E(\mathbb{F}_q) \rfloor + 1$
Put $P_i := 2^{i-1}P$ $(i = 1, 2, ..., d)$
Put $P_0 := 2Q - \sum_{i=1}^{d} P_i$
Solve $EQS2(P_0, ..., P_d)$ (Note:Cost for solving is $O(exp(q^{(1+\alpha)\beta+o(1)}))$)
if $EQS2(P_0, ..., P_d)$ has no solution **then**
  return **False**.
Compute $l'_i \in \{\pm 1\}$ $(i = 1, ..., d)$ st. $P_0 + \sum_{i=1}^{d} l'_i P_i = 0$ from the solution of $EQS2$
Put $l_i := (-l'_i + 1)/2 \ \in \{0, 1\}$ $(i = 1, ..., d)$
if $Q - \sum_{i=1}^{d} l_i P_i = 0$ **then**
  Return $\sum_{i=1}^{d} l_i 2^{i-1}$
else
  Compute $T_0 (\in E[2] \backslash \{\infty\} \cap < P >)$
  Compute Integer $\log_P T_0$ satisfying $(\log_P T_0)P = T_0$
  Return $\sum_{i=1}^{d} l_i 2^{i-1} + (\log_P T_0)$

7 Some bits known ECDLP

Here, we give the algorithm for solving discrete logarithm problem for elliptic curve in which some bits of the value of the discrete logarithm is known.

Problem 3 ((Some bits known ECDLP)) *Let $E/\mathbb{F}_q$ be an elliptic curve and let $P, Q \in E(\mathbb{F}_q)$ such that $< P > \ni Q$. Let $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_{unknown}$ are the disjoint division of $\{0, 1, 2, ..., \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor\}$. Assume there exists some N written by the form $\sum_{i \in \mathcal{I}_1} 2^i + \sum_{i \in \mathcal{I}_{unknown}} l_i 2^i$ (where $l_i \in \{0, 1\}$) such that $Q = NP$. We call the problem finding N (which is equivalent to finding l'_i for $i \in \mathcal{I}_{unknown}$) by "Some bits known ECDLP".*

For a technical reason, we also consider the group $E[2] \cap < P >$ and consider the following "Modified some bits known ECDLP".

Problem 4 ((Modifyied some bits known ECDLP)) *Let $E/\mathbb{F}_q$ be an elliptic curve and let $P, Q \in E(\mathbb{F}_q)$ such that $< P > \ni Q$. Let $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_{unknown}$ are the disjoint division of $\{0, 1, 2, ..., \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor\}$. Assume there exists some N written by the form $\sum_{i \in \mathcal{I}_1} 2^i + \sum_{i \in \mathcal{I}_{unknown}} l_i 2^i$ (where $l_i \in \{0, 1\}$) such that $Q - NP \in E[2] \cap < P >$. We call the problem finding N (which is equivalent to finding l'_i for $i \in \mathcal{I}_{unknown}$) by "Modified some bits known ECDLP".*

In the case $E[2] \cap < P >= \{\infty\}$, "Modified some bits known ECDLP" is the same as "Some bits known ECDLP". Otherwise, suppose we have the relation $Q - NP = T_0 \in \{T_0, \infty\} = E[2] \cap < P >$, we have $Q = (N + \log_P T_0)P$ and ECDLP is solved. Here, we also remark that "Modified some bits known ECDLP" is essentially the same as "Some bits known ECDLP".

Now, we will solve "Modified some bits known ECDLP".
Put $I_1, ..., I_d$ by $\{I_1, ..., I_d\} = \mathcal{I}_{unknown}$, $N_0 := \sum_{i \in \mathcal{I}_1} 2^i$, and $P_i := 2^{I_i} P$ $(i = 1, 2, ..., d)$. Also put $P_0 := 2(Q - N_0 P) - \sum_{i=1}^{d} P_i$ and solving the equation system $EQS1(P_0, ..., P_d)$. Assume $EQS1$ has a solution. From the discussion of previous section $l'_i \in \{\pm 1\}$ $(i = 1, ...d)$ satisfying $P_0 + \sum_{i=1}^{d} (l'_i P_i) = 0$ can be computable. Also put $l_i := (-l'_i + 1)/2 \ \in \{0, 1\}$ $(i = 1, ..., d)$. Since $2 \sum l_i P_i = -\sum l'_i P_i + \sum P_i$, we have

$$0 = 2(Q - N_0 P) - \sum_{i=1}^{d} P_i + \sum l'_i P_i = 2(Q - N_0 P - \sum_{i=1}^{d} l_i P_i).$$

Thus we have

$$Q - N_0 P - \sum_{i=1}^{d} l_i P_i \in E[2] \cap < P > .$$

Assume $EQS1$ has no solution. we see similarly that such N does not exists.
Thus we have the algorithm for solving "Some bits known ECDLP".

Algorithm 5 Bit coincidence mining algorithm for Some bits known ECDLP

Input: $E/\mathbb{F}_q$ elliptic curve, $P, Q \in E(\mathbb{F}_q)$ st. $< P > \ni Q$, $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_{unknown}$ division of $\{0, .., \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor\}$
Assume $\exists N$ of the form $\sum_{i \in \mathcal{I}_1} 2^i + \sum_{i \in \mathcal{I}_{unknown}} l_i 2^i$ ($l_i \in \{0, 1\}$) such that $Q - NP \in E[2] \cap < P >$
Output: Integer N satisfying $NP = Q$
Put $I_1, ..., I_d$ by $\{I_1, ..., I_d\} = \mathcal{I}_{unknown}$
Put $N_0 := \sum_{i \in \mathcal{I}_1} 2^i$
Put $P_i := 2^{I_i} P$ $(i = 1, 2, ..., d)$
Put $P_0 := 2(Q - N_0 P) - \sum_{i=1}^{d} P_i$
Solve $EQS1(P_0, ..., P_d)$
if $EQS1(P_0, ..., P_d)$ has no solution **then**
Such N does not exists and return **False**.
Compute $l'_i \in \{\pm 1\}$ $(i = 1, ..., d)$ st. $P_0 + \sum_{i=1}^{d} l'_i P_i = 0$ from the solution of $EQS1$
Put $l_i := (-l'_i + 1)/2 \in \{0, 1\}$ $(i = 1, ..., d)$
if $Q - N_0 P - \sum_{i=1}^{d} l_i P_i = 0$ **then**
Return $N = N_0 + \sum_{i=1}^{d} l_i 2^{I_i}$
else
Compute $T_0 (\in E[2] \backslash \{\infty\} \cap < P >)$
Compute Integer $\log_P T_0$ satisfying $(\log_P T_0) P = T_0$
Return $N = N_0 + \sum_{i=1}^{d} l_i 2^{I_i} + (\log_P T_0)$

Remark Algorithm 5 returns **False** only in the case that the assumption "$\exists N$ of the form $\sum_{i \in I_1} 2^i + \sum_{i \in I_{unknown}} l_i 2^i$ ($l_i \in \{0, 1\}$) such that $Q - NP \in E[2] \cap < P >$" is not true.

Now, we will consider the probabilistic version of "Bit coincidence mining algorithm".

Here also let $E/\mathbb{F}_q$ be an elliptic curve and let $P, Q \in E(\mathbb{F}_q)$ such that $< P > \ni Q$. Let $\mathcal{I}'_0, \mathcal{I}'_1, \mathcal{I}'_{unknown}$ are the disjoint division of$\{0, 1, 2, ..., \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor\}$. This is the same notation of Algorithm 5, but the symbols $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_{unknown}$ are replaced $\mathcal{I}'_0, \mathcal{I}'_1, \mathcal{I}'_{unknown}$ respectively.

Assume there exists some N written by the form $\sum_{i \in \mathcal{I}'_1} 2^i + \sum_{i \in \mathcal{I}'_{unknown}} l_i 2^i$ (where $l_i \in \{0, 1\}$) such that $Q - NP \in E[2] \cap < P >$.

Put $\mathcal{I}_{unknown}$, $\mathcal{I}_{random}$ by the division of $\mathcal{I}'_{unknown}$, (i.e.,)

$$\mathcal{I}_{unknown} \cup \mathcal{I}_{random} = \mathcal{I}'_{unknown} \qquad \text{(disjoint division)}$$

and put $\mathcal{I}_{random,0}$, $\mathcal{I}_{random,1}$ by the division of $\mathcal{I}_{random}$, (i.e.,)

$$\mathcal{I}_{random,0} \cup \mathcal{I}_{random,1} = \mathcal{I}_{random,1} \qquad \text{(disjoint division)}.$$

From this notation, N is written by

$$\sum_{i \in \mathcal{I}'_1} 2^i + \sum_{i \in \mathcal{I}_{unknown}} l_i 2^i + \sum_{i \in \mathcal{I}_{random,0}} l_i 2^i + \sum_{i \in \mathcal{I}_{random,1}} l_i 2^i.$$

So, $1/2^{\#\mathcal{I}_{random}}$ probability, we have

$$l_i = 0 \quad (i \in \mathcal{I}_{0,random}) \text{ and } l_i = 1 \quad (i \in \mathcal{I}_{1,random}).$$

Thus, put

$$\mathcal{I}_0 := \mathcal{I}'_0 \cup \mathcal{I}_{0,random}, \quad \mathcal{I}_1 := \mathcal{I}'_1 \cup \mathcal{I}_{1,random}$$

and apply the Algorithm 5, we have the value of ECDLP in $1/2^{\#\mathcal{I}_{random}}$ probability. Otherwise we have the result "**False**" in $1 - 1/2^{\#\mathcal{I}_{random}}$ probability.

Thus, we have the following probabilistic (so that the required equations system is small) algorithm.

Algorithm 6 Probabilistic bit coincidence mining algorithm

Input: $E/\mathbb{F}_q$ elliptic curve, $P, Q \in E(\mathbb{F}_q)$ st. $< P >\ni Q$, $\mathcal{I}'_0, \mathcal{I}'_1, \mathcal{I}'_{unknown}$ division of $\{0, .., \lfloor \log_2 \#E(\mathbb{F}_q) \rfloor\}$
Assume $\exists N$ of the form $\sum_{i \in \mathcal{I}'_1} 2^i + \sum_{i \in \mathcal{I}'_{unknown}} l_i 2^i$ ($l_i \in \{0, 1\}$) such that $Q - NP \in E[2] \cap < P >$
Fix natural number d' ($< \#\mathcal{I}'_{unknown}$) as parameter
Start:
Put $\mathcal{I}_{unknown} \cup \mathcal{I}_{random} = \mathcal{I}'_{unknown}$ by random division of size $\#\mathcal{I}_{random} = d'$
Put $\mathcal{I}_{0,random} \cup \mathcal{I}_{1,random} = \mathcal{I}_{random}$ by random division
Put $\mathcal{I}_0 := \mathcal{I}'_0 \cup \mathcal{I}_{0,random}, \mathcal{I}_1 := \mathcal{I}'_1 \cup \mathcal{I}_{1,random}$
Call Algorithm 5
if Algorithm 5 returns value N **then**
Return N
else `/*Algorithm 5 returns False*/`
Goto **Start**

Remark The average number of the calls of Algorithm 5 in Algorithm 6 is $2^{d'}$ where d' is the parameter in Algorithm 6.

8 Remarks

1. Consider the problem of solving equations system consists of n variables and m degree δ equations by xL method. Here also assume $m \geq n$. Put $D \sim n^{(\delta-1)/\delta}$ and we have "$\#\mathcal{M}_D <$ # of the equation obtained by multiply step". So, it is expected that in the case $m = n + C_0$ for small C_0, the complexity of solving equations system is $O(exp(n^{(\delta-1)/\delta+o(1)}))$ where $\lim_{n\to\infty} o(1) = 0$.

2. This algorithm can be (or may be) generalized to the Jacobian of the curves. In the case of the Jacobian of hyperelliptic curves, degree of the obtained equations system is 2 and sub-exponentially under the Assumption 1 can be also shown. However, in the case of the Jacobian of the general curve, degree of the obtained equations system is > 2. [5]

3. Instead of using Riemann-Roch space, one can use equations system consists of Semaev formal $S_3(x, y, z) = 0$ of 3 variables [5], [13]. Using S_3, ECDLP reduced to solving equations system consists of $d - 1$ variables, d degree 4 equations, where d is taken $\lfloor \log_2 \#E(\mathbb{F}_q) \rfloor + 1$.

4. We can easily construct sub-exponential version of "Some bit known ECDLP" and "Probabilistic some bit known ECDLP" algorithms under the assumption of the complexity of xL algorithm.

Acknowledgement I would like to have great thanks to Professor Kazuto Matsuo in Kanagawa University for useful advices and coments.

[5] When the degrees of the equations is > 2, we must modify the Assumption and sub-exponentially seems to be shown under the modified assumption

References

1. J. Ding, J. Buchmann, M. Mohamed, W. Mohamed and R-P Weinmann, MutantXL, http://www.academia.edu/2863459/Jintai_Ding_Johannes_Buchmann_Mohamed_Saied_Emam_Mohamed
2. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In Proceedings of International Conference on the Theory and Application of Cryptographic Tech- niques(EUROCRYPT), volume 1807 of Lecture Notes in Computer Science, pages 392–407, Bruges, Belgium, May 2000. Springer.
3. J-C. Faugére, L. Perret, C. Petit, and G. Renault, Improving the complexity of index calculus algorithms in elliptic curves over binary fields, EUROCRYPTO 2012, LNCS **7237**, pp.27-44.
4. S. Galbraith and S.Gebregiyorgis, Summation polynomial algorithms for elliptic curves in characteristic two, https://eprint.iacr.org/2014/806
5. Y. Huang, C. Petit, N. Shinohara, and T. Takagi, On Generalized First Fall Degree Assumptions, https://eprint.iacr.org/2015/358
6. M. Kosters, NOTES ON SUMMATION POLYNOMIALS, http://arxiv.org/pdf/1503.08001.pdf 2015.
7. K. Nagao, Index calculus for Jacobian of hyperelliptic curve of small genus using two large primes, Japan Journal of Industrial and Applied Mathematics, **24**, no.3, 2007.
8. K. Nagao, Decomposition Attack for the Jacobian of a Hyperelliptic Curve over an Extension Field, 9th International Symposium,ANTS-IX., Nancy, France, July 2010, Proceedings LNCS 6197,Springer, pp.285–300, 2010.
9. K. Nagao, Decomposition formula of the Jacobian group of plane curve, https://eprint.iacr.org/2013/548
10. K. Nagao, Equations System coming from Weil descent and subexponential attack for algebraic curve cryptosystem, https://eprint.iacr.org/2013/549
11. C. Petit and J-J. Quisquater. On Polynomial Systems Arising from a Weil Descent, Asiacrypt 2012, Springer LNCS **7658**, Springer, pp.451-466.
12. I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. https://eprint.iacr.org/2004/031.pdf
13. I. Semaev, New algorithm for the discrete logarithm problem on elliptic curves, https://eprint.iacr.org/2015/310.pdf