УДК 681.3.06

ШВИДКІ АЛГОРИТМИ ДЛЯ ОБЧИСЛЕННЯ ІЗОГЕНІЙ НА ЕЛІПТИЧНИХ КРИВИХ

В. А. ПОНОМАР, О. Г. БЕРЕЖНИЙ

Розглядаються та аналізуються алгоритми обчислення ізогеній еліптичних кривих над скінченним полем. Аналізується алгоритм, обчислення ізогенії ступеня L, що грунтується на швидких алгоритмах розкладання β-функції і пов'язані з ними функції в ряд Вейерштрасса. Наводяться рекомендації та пропозиції відносно оцінки складності алгоритмів обчислення ізогеній ступеню L.

Ключові слова: алгоритми обчислення ізогеній, ізогенії еліптичних кривих, електронні підписи, постквантовий період, швидкі алгоритми обчислення ізогеній еліптичних кривих.

ВСТУП

У 2014 – 2016 роках отримані суттєві результати в побудуванні квантового комп'ютера [1,4]. Ще раніше розроблені та практичні готові до використання методи квантового криптоаналізу. Їх реалізація на квантовому комп'ютерові дозволить успішно атакувати більшість асиметричних криптосистем. Підтвердженням цьому є спочатку поява і Internet статі «A RIDDLE WRAPPED IN AN ENIGMA [1], в якій зазначається, що в серпні 2015 року агентство національної безпеки (АНБ) уряду США виступило з заявою про слабкість існуючих асиметричних криптосистем відносно квантового криптоаналізу. В 2016 році опубліковано звіт «Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT) [2]», в якому повністю підтверджено можливості успішного квантового криптоаналізу асиметричних криптосистем, а також визначені основні проблеми та можливості і етапи їх вирішеня.

Серед можливих методів та алгоритмів побудови асиметричних криптосистем називається метод, що грунтується на використанні криптографічних перетворень на ізогеніях еліптичних кривих. В [3,4] наведено ряд даних, що підтеджують перспективність криптографічних перетворень з використанням математичного апарату ізогеній еліптичних кривих. При цьому, в першу чергу ставляться задачі побудування криптографічних механізмів електронного підпису(ЕП). Вирішення вказаної, на наш погляд суттєво проблемної задачі, пов'язане з доведенням криптографічної стійкості, забезпеченням необхідної швидкодії (складності) перетворень та обгрунтуванням і побудуванням загальних параметрів та ключів. Зважаючи на стан досліджень в указаному напряму, на наш погляд, актуальними є задачі аналізу та оптимізації криптографічних перетворень на ізогеніях еліптичних кривих.Тому метою цієї статі є обгрунтування та формулювання основних задач аналізу складності побудування параметрів і ключів, а також прямих та зворотних асиметричних криптоперетворень. Зрозуміло, що суттєво актуальною є проблема доведення криптографічної стійкості, на наш погляд вона розв'язуватиметься на світовому рівні ще декілька років.

1.Сутність криптографічних перетворень на ізогеніях еліптичних кривих

Спочатку розглянемо основні складові механізму та алгоритмів криптографічних перетворень на ізогеніях еліптичних кривих[3 – 5].

Ізогенія – це раціональне відображення $\varphi : E_1(K) \to E_2(K)$, де $E_1(K)$ та $E_2(K)$є еліптичними кривими, а $\varphi(P_\infty) = P_\infty$. Нульова ізогенія – це ізогенія, що відображає усі точки однієї кривої, у точку на нескінченності іншої. Ядром ізогенії є

$$\mathrm{Ker}(\varphi) = \{ K_i \in E_1 \};\ \varphi(K_i) = P_\infty .$$

У цілому ізогенії ініціюють відображення полів функцій на кривих. Степінь розширення $(K(E_1):\ \varphi^*K(E_2))$ називається степінню ізогенії.

Відносно певної ізогенії $\varphi : E_1(K) \to E_2(K)$ існує дуальна ізогенія $\hat{\varphi} : E_2(K) \to E_1(K)$, така, що $\hat{\varphi}\varphi = [l]$, де l – множення точки кривої E_1 на число l, аналогічно $\varphi\hat{\varphi} = [l]$, де l – множення точки кривої E_2 на число l. При цьому дуальні ізогенії мають однакову степінь.

Для випадку алгебраїчно замкненого поля операція множення точки на число l задає ендоморфізм еліптичної кривої з ядром l^2 точок. Оскільки ізогенія відповідає квадратному кореню з операції множення на l, то ядро ізогенії складається з l точок порядку l, що створюють циклічну групу (однією з них є точка P_∞).

Ізогенії складних степенів можуть використовуватися, як композиція ізогеній простих степеней.

Властивості ізогеній, що використовуються в ході створення криптосистем:

$$1)\ \gamma\left(\varphi\left(A\right)\right)=\varphi\left(\gamma\left(A\right)\right)=\gamma\varphi\left(A\right);$$

$$2)\ k^{*}A_{\varphi}=\varphi\left(k^{*}A\right).$$

Знаходження ізогеній по ядру. Для знаходження ізогенії $\varphi:E_1\left(K\right)\rightarrow E_2\left(K\right)$, з заданим ядром використовується формула Велу [6,7,14-15].

Для еліптичної кривої, що задана формулою Вейєрштрасса:

$$E_1:\ y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6,$$

узагальнений алгоритм знаходження ізогенії по ядру наведено в [9]. З урахуванням того, що в криптографії еліптичних кривих прийнято використовувати спрощену формулу Вейєрштрасса, то розглянемо алгоритм знаходження ізогенії по ядру й зведемо до прийнятого вигляду, використовуючи формулу

$$E_1:\ y^2=x^3+a_4x+a_6.$$

Нехай C – група точок еліптичної кривої, що буде ядром ізогенії. Тоді:

1. Формуємо набір точок S:

а) Виключаємо з C точку на нескінченності.

б) Нехай C_2 – усі точки C, в яких координата y дорівнює нулю, а R – усі інші точки C.

в) Розділимо точки набору R на R_+ та R_-, але так, що для кожної точки P, що належить R_+, -P належить R_-.

г) $S=R_+\cup C_2$.

2. Для кожної точки $Q\in S$ виконуємо такі обчислення:

а) $g_Q^x=3x_Q^2+a_4$;

б) $g_Q^y=-2y_Q$;

в) $v_Q=\begin{cases}g_Q^x, & 2Q=\infty\\ 2g_Q^x, & 2Q\neq\infty\end{cases}$;

г) $u_Q=\left(g_Q^y\right)^2$;

г) $v=\sum_{Q\in S}v_Q$;

д) $w=\sum_{Q\in S}\left(u_Q+x_Qv_Q\right)$.

3. Розраховуємо формулу еліптичної кривої $E_2(K)$:

а) $A_4=a_4-5v$;

б) $A_6=a_6-7w$;

в) $E_2:\ y^2=x^2+A_4x+A_6$.

4. Розраховуємо координати точки

$$\left(x_\varphi,y_\varphi\right)=\varphi\left(x,y\right):$$

а) $x_\varphi=x+\sum_{Q\in S}\left(\frac{v_Q}{x-x_Q}+\frac{u_Q}{\left(x-x_Q\right)^2}\right)$;

$$y_\varphi=y-\sum_{Q\in S}$$

б) $\left(u_Q\frac{2y}{\left(x-x_Q\right)^3}+v_Q\frac{y-y_Q}{\left(x-x_Q\right)^2}-\frac{g_Q^xg_Q^y}{\left(x-x_Q\right)^2}\right)$

2.Вимоги до загальних параметрів та можливості і методи їх побудування

Важливим будівельним блоком у роботі Elkies є алгоритм, який обчислює криві, які є ізогеніями на заданій кривій E. Цей блок використовує модульні поліноми, щоб отримати список ізогеній кривих і формул V'elu, щоб отримати явний вигляд ізогеній $I:E\rightarrow\tilde{E}$, де $\tilde{E}$ знаходиться у відповідній формі Вейерштрасса. У цій роботі ми концентруємося на алгоритмах, які будують ступінь L ізогенії I від E і $\tilde{E}$. Ми могли б обмежити далі до випадку, коли L непарне просте, оскільки ізогенії можна записати у вигляді композицій ізогеній простого ступеня. До того ж, непарний простий випадок є найбільш важливим в SEA. Проте, наші результати стоять для довільного L. Ми вимагаємо, щоб характеристика ρ основного поля **K** було 0 або ρ>>L. Це обмеження задовольняється в разі зацікавленості в застосуванні до алгоритму SEA, оскільки в іншому випадку ρ-адичні методи набагато швидше і простіше у використанні.

З нашого припущення про ρ випливає, що рівняння наших кривих можна записати у формі Вейєрштрасса

$$y^2 = x^3 + A*x + B. \quad (1)$$

У нульовій характеристиці, крива (1) може бути параметризована $(x, y) = \left(\rho(z), \frac{p'(z)}{2}\right)$ зважаючи на те, що класичне диференціальне рівняння

$$p'(z)^2 = 4(p(z)^3 + A*p(z) + B) \quad (2)$$

відповідає ρ–функції Вейєрштрасса. Це є основою для нашого обчислення ізогеній. Таким чином, ми доведемо два результати, спочатку на обчисленні Вейєрштрасса ρ-функції, а потім на обчисленні самих ізогеній.

Тут особливість полягає у використанні при розрахунках класичних швидких алгоритмів для степеневих рядів і демонструється, як їх можна застьосувати до обчислення ізогеній. Позначимо $M : N \to N$ функцію таку, що многочлени степеня менше n можуть бути мультиплыкативні в операціях, де M (n) основне поле. За допомогою швидкого перетворення Фур'є [8, 16], можна знайти

$$M(n) \in O(n \log n \log\log n)$$

над полями, що містять примітивні об'єднанні корені, причому $M(n) \in O(n \log n)$.

3.Вимоги до алгоритмів генерування асиметритчнизх пар ключів та оцінка їх властивостей.

Квадратичний алгоритм Elkies. Далі розглядатимемо дві криві E і Ẽ, через які вейєрштрассове рівняння допускатиме нормалізування ізогенія I: E → Ẽ степеня l. Подамо ці криві у вигляді

$$E : y^2 = x^3 + Ax + B \quad \text{і} \quad \tilde{E} : y^2 = x^3 + \tilde{A}x + \tilde{B}. \quad (3)$$

Визначимо залежно від вхідних даних ізогенії I, які подаватимемо у вигляді

$$I(x, y) = \left(\frac{N(x)}{D(x)},\ y\left(\frac{N(x)}{D(x)}\right)'\right). \quad (4)$$

Спочатку розглянемо алгоритм Elkies [11], чия складність квадратична за ступенем l. Далі з метою порівняння розглянемо два швидких варіанти алгоритму Elkies1998, так звані fastElkies і fastElkies', складність яких відповідно $O(M(l))$ і $O(M(l)\log l)$.

Як показав аналіз алгоритм Elkies1998 був введений для простого випадку ступеня в роботі [11], але він може застосовуватися для будь-яких великих l. Перша частина алгоритму спрямована на обчислювальні розкладання $\frac{N(x)}{D(x)}$ на нескінченності; друга частина становить відновлення сум коренів $D(x)$ з цього розкладу.

Для того, щоб провести аналіз, розглянемо раціональні функції N (x) / D (x), які задовольняють нелінійне диференціальне рівняння

$$\left(x^3 + Ax + B\right) + \left(\frac{N(x)}{D(x)}\right)'^2 = \left(\frac{N(x)}{D(x)}\right)^3 + \tilde{A}\left(\frac{N(x)}{D(x)}\right) + \tilde{B}. \quad (5)$$

Це випливає з того, що I відображає E в $\tilde{E}$. Диференційне рівняння (5) перебудовується до наступного рівняння другого порядку

$$(3x^3 + A)\left(\frac{N(x)}{D(x)}\right)' + 2\left(x^3 + Ax + B\right)\left(\frac{N(x)}{D(x)}\right)'' = 3\left(\frac{N(x)}{D(x)}\right)^2 + \tilde{A}. \quad (6)$$

Записуючи розкладання раціональної функції N (x) / D (x) на нескінченності

$$\frac{N(x)}{D(x)} = x + \sum_{i \ge 1}\left(\frac{h_i}{x^i}\right)$$

і визначення коефіцієнтів x^{-i} з обох сторін рівняння (6) дає можливість подати

$$h_k = \frac{3}{(k-2)(2k+3)}\sum_{i=1}^{k-2} h_i h_{k-1-i} - \frac{2k-3}{2k+3}Ah_{k-2} - \frac{2(k-3)}{2k+3}Bh_{k-3}, \quad (7)$$

для всіх $k \ge 3$, з початковими умовами

$$h_1 = \frac{A - \tilde{A}}{5} \quad \text{і} \quad h_2 = \frac{B - \tilde{B}}{7}.$$

Подання (7) є основою алгоритму Elkies1998; використовуючи його, можна обчислити $h_3,\ldots, h_{l-2}$ з використанням $O\left(l^2\right)$ операцій в K.

Також в алгоритмі 'Elkie 1998 передбачається, що $p_1 = \sigma$. Подання коефіцієнів в рівнянні дає, що

$$h_i = (2i+1)p_{i+1} + (2i-1)Ap_{i-1} + (2i+2)Bp_{i-2}, \text{ для всіх } i \geq 1. \quad (8)$$

Оскільки $h_1,\ldots, h_{l-2}$ відомі, $p_2,\ldots, p_{l-1}$ можна вивести, використовуючи $O(l)$ операцій. Потім отримати неоліном D (x), або шляхом квадратичного алгоритму або з використанням швидкого алгоритму .2, а N (x) можна отримати за допомогою формули (4), в $O\left(M(l)\right)$ операцій.

У такому алгоритмі необхідно, щоб в K було одиниць $2,\ldots,2l-1$. Його складність $O\left(l^2\right)$, причому вузьким місцем є обчислення коефіцієнтів $h - h_1,\ldots, h_{l-2}$. Для цього можна застосовувати паралельні обчислення згідно з [7], де диференційні рівняння Вейєрштрасса дають (7), , якщо взяти A=B= =0 в останньому).

4. Швидкі алгоритми.

Важливим є те, що необхідно покращити в алгоритмі Elkies1998 обчислення коефіцієнтів h_i, а решта є незмінною. Але, як показав аналіз, розкладання N (X) / D (x) на нескінченності можна зробити використовуючи диференціальне рівняння (5), за умови, що рівняння, отримано заміною змінних x → 1/x. Для того, щоб уникнути ускладнення, ми переважно можна застосовувати ступеневі ряди вигляду

$$S(x) = x + \frac{\tilde{A} - A}{10}x^5 + \frac{B - \tilde{B}}{14}x^7 + O\left(x^9\right) \in x + x^3 K\left[\left[x^2\right]\right]$$

за умови, що

$$\frac{N(x)}{D(x)} = \frac{1}{S\left(\frac{1}{\sqrt{x}}\right)^2}.$$

Також необхідно враховувати, що для S справедливе співвідношення $\bar{R}$=SoR, де

$$R(z) = 1/\sqrt{p(z)} \text{ і } \bar{R} = 1/\sqrt{p^{\sim}(z)}.$$

У подальшому, застосовуючи правило ланцюга, можна отримати диференціальне рівняння першого порядку, яке якому задовольняє S (x), тобто

$$\left(Bx^6 + Ax^4 + 1\right)S'(x)^2 = 1 + \tilde{A}S(x)^4 + \tilde{B}S(x)^6. \quad (9)$$

За допомогою (9) для обчислення N (X) / D (X) можна застосувати два алгоритми), така можливість залежить від того, чи відомо коефіцієнт σ чи ні. Для цих алгоритмів маємо

$S(x) = xT\left(x^2\right)$ і $U(x) = \frac{1}{T(x)^2} \in 1 + x^2 K\left[\left[x\right]\right]$ так що $\frac{N(x)}{D(x)} = xU\left(\frac{1}{x}\right)$.

У першому алгоритмі, що називається як fastElkies, передбачається, що σ відоме, тому

1) Обчислюється

$$C(x) = \left(Bx^6 + Ax^4 + 1\right)^{-1} \bmod x^{2l-1} \in K\left[\left[x\right]\right];$$

2) Обчислюється $S(x) \bmod x^{21}$ з використанням того, що $G(x,t) = C(x)\left(1 + \tilde{A}t^4 + \tilde{B}t^6\right)$, і далі знаходиться T (x) $\bmod x^{1}$ `;

3) Обчислюється $U(x) = 1/T(x)^2 \bmod x^1$;

4) Знаходяться коефіцієнти $h ,\ldots, h_{l-2}$ N (x) / D (x), використовуючи N (x) / D (x) = xU (1 / x);

5) Робиться підрахунок сум $p_2,\ldots, p_{l-1}$ з D (x), з використанням лінійного повторення(8);

6) Відновлюється D (x) з використанням сум, як описано в п.2.

7) Обчислюється N (x) з використанням рівняння також (4).

Етапи 1) і 5) мають складність порядку $O(l)$. Етапи 2), 3), 6) і 7) можуть бути виконані за $O\left(M(l)\right)$ операцій, а етап (4) не вимагає складних операцій.

У другому алгоритмі, який називають як fastElkies', не ставиться вимога знання σ. Його кроки

1 ') – 3') є лише результат невеликої зміни кроків (1) – (3). Їх складність має з точністю до констант порядок $O\left(M\left(l\right)\right)$.

Алгоритм вимагає виконання таких етапів:

1) Обчислюється

2)

$$C\left(x\right)=\left(Bx^{6}+Ax^{4}+1\right)^{-1} \bmod x^{8l-5} \in K\left[\left[x\right]\right];$$

обчислюється $S\left(x\right) \bmod x^{8l-4}$ з використанням алгоритму розділу 4, де

$G\left(x,t\right)=C\left(x\right)\left(1+\tilde{A}t^{4}+\tilde{B}t^{6}\right)$, і залишається вивести T (x) $\bmod x^{4l-2}$ ';

3) Обчислюється

$U\left(x\right)=1/T\left(x\right)^{2} \bmod x^{4l-2}$ з використанням алгоритму §2.1;

4) Реконструюється раціональна функція U(x);

5) Знаходиться N(x)/D (x) = xU(1/x).

Виконанння швидкої раціональної реконструкції на кроці 4 ') може бути виконана зі складністю $O\left(M\left(l\right)\log l\right)$ операції в K. Також можна перевірити, що алгоритм fas tElkies вимагає що $2,\ldots,2l-1$ будуть одиниці в K, в той час як алгоритм fastElkies' вимагає, що $2,\ldots,8l-5$ будуть одиниці в K.

У разі непарного l, необхідно замість D(x обчислити $g\left(x\right)$).

Нехай $q_1,q_2,\ldots$ ступеневі суми $g\left(x\right)$ так, що $q_i=p_i/2$. Тоді коефіцієнти h_i і степені суми q_i пов'язані співвідношенням

$$h_i=(4+2)q_{i+1}+(4i-2)Aq_{i-1}+(4i-4)Bq_{i-2}. \qquad (10)$$

Для того, щоб обчислити $g\left(x\right)$ з використанням алгоритму fastElkies, достатньо обчислити $S\left(x\right) \bmod x^{l+1}$; тоді T(x) і U(x) обчислюються за модулем $x^{(l+1)/2}$. Аналогічним чином, в алгоритмі fastElkies' достатньо обчислити $S\left(x\right) \bmod x^{41}$, і T(x) і U(x) за модулем x^{21} `.

5. Метод Старка.

Як показує аналіз, перший subcubic метод для знайдених N і D пов'язано зі Stark [21] і становить розширення $p^{\smile}$, як і раніше фракцію в (19). Фракція N / D апроксимується через $\frac{p_n}{q_n}$ і алгоритм зупиняється, коли ступінь n $l-1$ дає D. Зокрема, це можна застосовувати для будь-якого ступеня ізогенії. Оскільки p і $p^{\smile}$ в $\frac{1}{z^2}+K\left[\left[z^2\right]\right]$, то достатньо працювати з зі значеннями в $Z = z^2$.

В результаті маємо

1) $T=p^{\smile}(Z)+O\left(Z^{1}\right)$;

2) $n=1$;

3) $q_0=1$;

4) $q_1=0$;

Оскільки $\deg\left(q_n\right)<l-1$, обчислюємо

a) $n=n+1$;

b) $a_n=0$;

c) Поки $r\geq 1$ знаходимо

$a_n=a_n+t_{-r}z^{r}$;

$T=T-t_{-r}p^{r}=t_{-s}Z^{-s}+\ldots$;

r=s

d) $q_n=a_nq_{n-1}+q_{n-2}$;

e) T:=1/T;

5) Отримуємо D:= q_n.

Вказаний алгоритм називається як Stark1972, що має $O\left(1\right)$ складність і проходить через стадії 5); а оцінка досягається в загальному випадку, при r = 1 на кожному кроці. Крок, який визначає складність, зводиться до обчислення зворотних значень на стадії (5) з точністю $2l-1-2\deg q_n-2r$. загальна складність цих операцій $O\left(lM\left(1\right)\right)$. Множення на стадії (5.d) може бути зроблено за час $O\left(lM\left(1\right)\right)$, при чому ці множення можна зробити швидше, якщо це необхідно). Оскільки найбільша ступінь многочленів a_n обмежена $l-1$, p на кроці (5.c) також зводиться до складності $O\left(lM\left(1\right)\right)$і. На останок, знаючи, D (X), чисельник N (x) може бути відновлений зі складністю $O\left(M\left(1\right)\right)$ з використанням.

У разі, коли l непарне, то і як у алгоритмів SEA обчислюється зі складністю $O\left(M\left(1\right)\right)$ операцій, наприклад, шляхом обчислення $\exp\left(\log D/2\right)$.

Таким чином, загальна складність алгоритму Stark1972 оцінюється як в $O(lM(l))$. Зауважимо, що порівняно з методами, навединими нижче, алгоритм Stark1972 не вимагає знання σ. p^r на стадії (5.c) може бути амортизоване в контексті алгоритму SEA.

6. Метод Elkies1992.

Розглянемо метод як Elkies1992 , що наведений в [10]. Покладемо, що l непарне так, що

$$D(x) = g(x)^2,$$

хоча незначні зміни призведуть до спільного розв'язання.

Диференціюючи двічі вираз (2), отримаємо

$$\frac{d^4 p(z)}{dz^4} = 120p^3 + 72Ap + 48B.$$

Для більш загального випадку можна застосовувати рівності вигляду

$$\frac{d^{2k} p(z)}{dz^{2k}} = \mu_{k,k+1} p^{k+1} + \ldots + \mu_{k,0},$$

а для деяких констант $\mu_{k,j}$, які задовольняють рекурентне співвідношення, отримаємо

$$\mu^{k+1,j} = (2j-2)(2j-1)\mu_{k,j-1} + (2j+1)(2j+2)A\mu_{k,j+1} + (2j+2)(2j+4)B\mu_{k,j+2};$$

з $\mu_{k,k+1} = (2k+1)!$. [8]

Використовуючи це рекурентне співвідношення, коефіцієнти $\mu_{k,j}$, для $k \le d-1$ і $j \le k+1$, можуть бути бути обчисленими зі складністю $O(l^2)$ операції в K.

Elkies показав [9], як використовувати ці коефіцієнти для відновлення статечних сум $q_2, \ldots, q_d$ в g, за допомогою таких рівностей,отримуючи при k≥1:

$$(2k)!(c\check{}_k - c_k) = 2(\mu_{k,0} q_0 + \ldots + \mu_{k,k+1} q_{k+1}).$$

Використовуючи ці рівності, за умови, що $q_1 = \frac{\sigma}{2}$ і коефіцієнти $c_k, \ldots, c\check{}_k$ і $\mu_{k,j}$ відомі, можна відновити $q_2, \ldots, q_d$ шляхом вирішення системи, в якої складність $O(l^2)$. Це дає можливість відновити g, використовуючи відповідний алгоритм.

Діагональна система $q_2, \ldots, q_d$ може бути зведена до квазілінійної відносно просторової складності. Для цього слід використовувати структуру з діагональної системи, з тим, щоб уникнути явного обчислення зі складністю $O(l^2)$ константи $\mu_{k,j}$.

7. Метод Аткіна.

В роботі [2], Аткін запропонував формулу, що дозволяє обчислення D (x) [16, 18] в разі, коли l непарне. Продовжимо це так, щоб охопити випадок довільної l, йогозначаення повертається D (X). Використаємо

$$D(p(z)) = z^{2-2l} \exp(F(z)),$$

де

$$F(z) = -\sigma z^2 + 2\left(\sum_{k=1}^{\infty} (lc_k - c\check{}_k)\ (z^{2k+2}\right) / (2k+1)(2k+2)). \qquad (11)$$

Оскільки l і коефіцієнти $c_k, c\check{}_k$ передбачаються відомими, можна dbpyfxbnb $F(z) \bmod z^l$, за умови, що σ відоме. Для цього можна скористатись прямим методом визначення D (X) , а потім обчислити експоненти $F(z)$, а також відновити коефіцієнти D (x, але по одному за кожен раз. Це викладено в алгоритмі, що названий як Atkin1992. Тоді, використовуючи серію значень в $Z = z^2$, маємо алгоритм.

1. Підраховуються серії $P_i(Z) = p(Z)^i$ порядку l, для $1 \le i \le l-1$;
2. Обчислюються $G(Z) = \exp_l(F(Z))$;
   T=G;
   D=0;
3. $D=D+tz^i$;
   $T=T-tP_i$.

На кроці 1) складність оцінюється як $O(lM(l))$; на 2) можна знехтувати, використовуючи або класичну або швидку експоненцію. На кроці 5) складність оцінюється як $O(l)$ більше, операції, а в

загальному випадку як $O(l^2)$. Таким чином, загальна складність алгоритму оцінюється як $O(lM(l))$. Якщо цей алгоритм використовується в контексті алгоритму SEA,то крок (1) може бути модифікований, оскільки вона залежить тільки від кривої E.

Таким чином, всі p мають обчислюватися для максимального значення l, які використовуватимуться і зберігатимуться. Тому складність цього алгоритму визначатиметься на кроці (5), тобто як $O(l^2)$.

Аналіз показує, що для обчислення D (X), за умови уникнення обчислення всіх $p(z)$, краще застосовувати рівняння (20), подане у вигляді

$$D\left(\frac{1}{x}\right) = I^{2-2l}\left((\exp \circ F) \circ I\right),$$
$$I(x) = p^{-1}\left(\frac{1}{x}\right), \qquad (12)$$

де p^{-1} є функціонально зворотним p. Розширення з $I(x)$ близьке $\Phi(l)$, може бути обчислене в $O(l)$ операцій з використанням диференційного рівняння

$$I'(x)^2 = \frac{1}{4x\left(1+Ax^2+Bx^3\right)} \quad \text{або}$$
$$I'(x) = \frac{1}{2\sqrt{x}}\ \frac{1}{\sqrt{1+Ax^2+Bx^3}}. \qquad (13)$$

Таким чином, маємо лінійне диференційне рівняння:

$$\frac{I'(x)}{I(x)} = -\frac{1+3Ax^2+4Bx^3}{2x\left(1+Ax^2+Bx^3\right)}.$$

З викладеного випливає, що лінійне диференціальне рівняння може бути поданим як

$$J(x) = x^{-\frac{1}{2}} I(x) = \sum_{i\geq 0} a_i x^i \ . \qquad (14)$$

Розпакування коефіцієнтів в цьому рівнянні, дає лінійну можливість

$$A_{i+1} = -\frac{2i-1}{2(i+1)(2i+3)}\left((2i-3)Ba_{i-2} + 2Aia_{i-1}\right)$$

для $i \geq 2$, з початковими умовами

$$a_0 = 1, a_1 = 0, a_2 = -\frac{A}{10}.$$

Вказане призводить до наступного алгоритму, що називається AtkinModComp.є

Його сутність у тому, що:

1. Обчислюється

$$G(Z) = \exp_1\left(F(Z)\right);$$

2. Обчислюється $I(x)$, використовуючи рівняння (23);

3. Обчислюється $G(Z)$ як модульна композиція;

4. Визначається D, використовуючи рівняння (13).

Складність алгоритму порядку

$$O\left(M(l)\sqrt{l}+l^{\frac{w+1}{2}}\right) \text{ або } O\left(M(l)\sqrt{l\log l}\right)$$

операції в K.

Для того, щоб зробити простіше потрібно переглянути $G(I) = (\exp o\ F)\ o\ I$, що використовується вище.

Рівняння Аткіна (11) можна також переписати у вигляді

$$D\left(p(x)\right) = \exp\left(-\sigma^2 + 2\int\int lp(x) - p\breve{}(x)\right).$$

Потім можна отримати D (1 / x) в наступному експонентному вигляді:

$$D\left(\frac{1}{x}\right) = \exp\left(-\sigma I^2 + 2\int I'\int I'\left(\frac{1}{x} - \left(p\breve{}\ \circ I\right)(x)\right)\right) \qquad (15)$$

$$= \exp\left(-\sigma I^2 + 2\int I'\int I'\left(\frac{1}{x} - \frac{N\left(\frac{1}{x}\right)}{D\left(\frac{1}{x}\right)}\right)\right) \qquad (16)$$

$$= \exp\left(-\sigma I^2 + 2\int I'\int I'\left(\frac{1}{x} - \frac{1}{S\left(\sqrt{x}\right)^2}\right)\right) \qquad (17)$$

Далі, уточнюючи деталі та послідовність операцій, отримаємо, що складність алгоритму буде такою ж як і для алгоритму fastElkies. Це зрозуміло, оскільки

$$\frac{N(x)}{D(x)} = lx - \sigma - 2\sqrt{x^3 + Ax + B}\left(\sqrt{x^3 + Ax + B}\,\frac{D'(x)}{D(x)}\right)'.$$

Тоді рівняння (16) не що інше, як інтегральне подання останнього рівняння.

8. Аналіз алгоритмів ЕП за критеріями криптографічної стійкості та складності.

Дослідження алгоритмів проведено, використовуючи бібліотеку NTL C ++ [19, 20] та AMD 64 з процесором 3400 (2,4 ГГц). На рис. 1 наведені результати порівняння складності для розширення p, отриманих над кінцевим полем $F_{10}^{2004} + 4683$.

Вигляд обох кривих вказує на те, що теоретичні складності майже квадратичні і майже лінійні, отже добре дотримуються в нашій реалізації. При цьому стрибки за ступенями 2 відображають особливість реалізації FFT NTL арифметику, що використовується в бібліотеці.

algorithm	complexity	need of σ
linear algebra	$O(\ell^{\omega})$	no
Stark1972	$O(\ell M(\ell))$	no
Atkin1992	$O(\ell M(\ell))$	yes
AtkinModComp	$O(M(\ell)\sqrt{\ell} + \ell^{\frac{\omega+1}{2}})$ or $O(M(\ell)\sqrt{\ell \log \ell})$	yes
Elkies1992	$O(\ell^2)$	yes
Elkies1998	$O(\ell^2)$	yes
fastElkies	$O(M(\ell))$	yes
fastElkies′	$O(M(\ell)\log \ell)$	no

Рис. 1. Порівняння алгоритмів

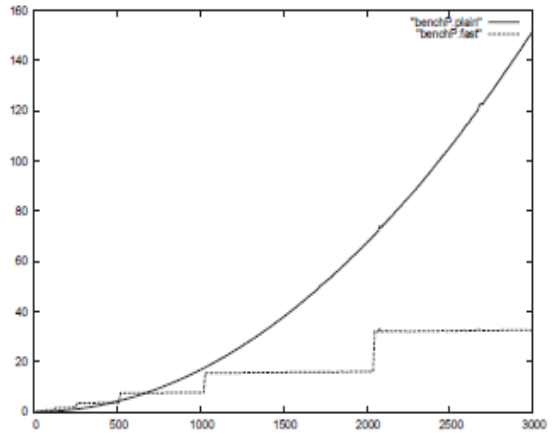

Рис. 2. Затримки для обчислень p на

$E: y^2 = x^3 + 4589x + 91128$ над $F_{10}^{2004} + 4683$

Крім того, існує поріг, за яким алгоритм стає ефективним (корисним), це робить його цікавим на практиці.

Тепер звернемо увагу на частини ізогеній, концентруючись на виразі де l первинна, в контексті алгоритму SEA. Отже, в цьому випадку, достатньо обчислити поліном $g(x)$, що $D(x) = g(x)^2$. Всі алгоритми можуть бути адаптовані, як скористатися цим спрощенням показано в пункті 4.3 для наших алгоритмів fastElkies і fastElkies'.

Перша серія таймингів належить обчисленню ізогенії над невеликим полем,

$$K = F_{10}^{19} + 51,$$

для кривої $E: y^2 = x^3 + 4589x + 91128$. Ми порівнюємо в рис. 2 виступи алгоритмів Elkies1992 з §6.3 і Elkies1998 з § 4.2 для ізогеніїв помірного ступеня $l \le 400$. Рис. 3 порівнює тимчасові діаграми, отримані з алгоритмом Elkies1998 і нашої швидкої версії fastElkies з §4.3, для ізогенії ступеня до 6000.

Далі ми порівняємо на рис. 4 тайминги, отримані $O(M(l))$ алгоритмом fastElkies, що вимагає знання σ, до результатів, отриманих за допомогою його $O(M(l)\log l)$ аналогу fastElkies', який не вимагає цієї інформації.

У всіх фігурах, градуси l з ізогеніями наведені на горизонтальній осі, а тайминги (в секундах) – на вертикальній осі. Знову ж таки, форма обидвох кривих на рис. 3 показує, що теоретичні складності добре дотримуються в нашій реалізації. Криві на рис. 4, показують, що теоретичне відношення $\log l$ між алгоритмами fastElkies і fastElkies' має подальший практичний вплив.

Далі, в таблицях 2 до 8, ми даємо докладні тайминги на обчисленні l -ізогенії для кривої

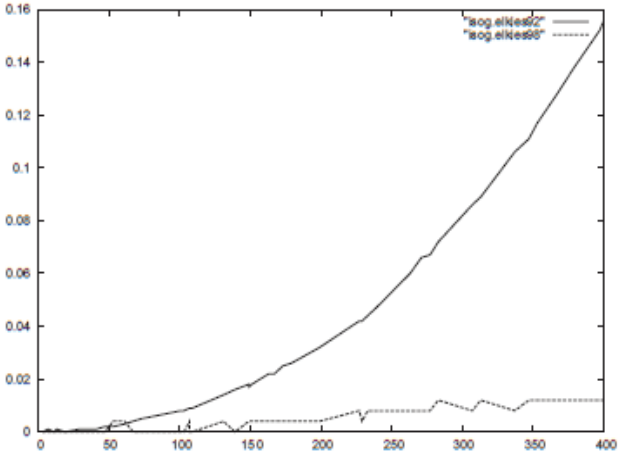

Рис. 3. Elkie1992 проти Elkie1998.

$$E : y^2 = x^3 + Ax + B$$

де

$$A = \left\lfloor 10^{1990} \pi \right\rfloor = 31415926 \ldots 58133904,$$

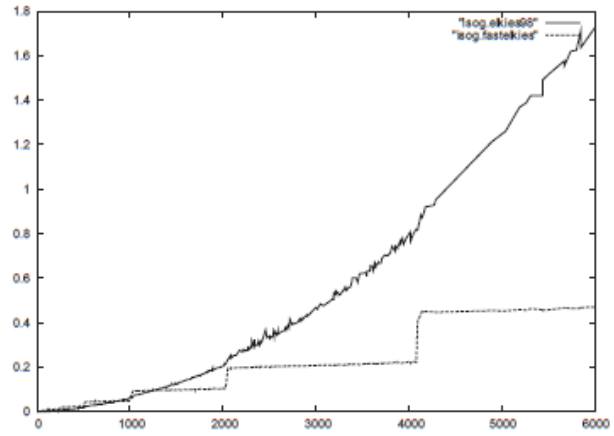

Рис. 4. Elkies1998 проти fastElkies

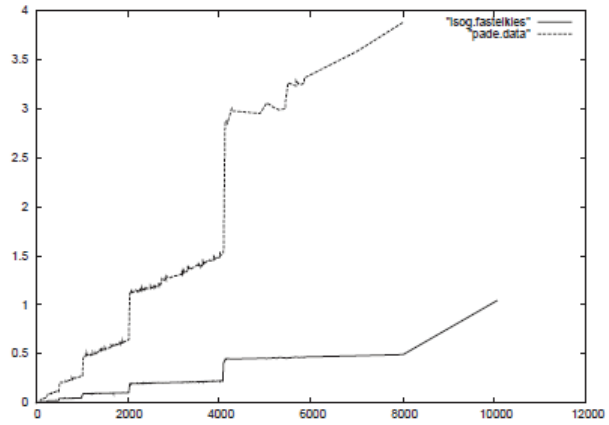

Рис. 5. FastElkies проти FastElkies'

для декількох значень l, над великим кінцевим полем $F_{10}^{2004} + 4683$, а також з використанням різних методів: алгоритми Elkies1992, Elkies1998, а також швидкий варіант fastElkies, алгоритм Stark, Stark1972 і дві версії Atkin1992 і AtkinModComp з алгоритмом Аткін. Отже

$$B = \left\lfloor 10^{1990} e \right\rfloor = 27182818 \ldots 94787610,$$

На рисунках 6 і 7 наведено таблиці значень складності для для основних підпрограм бібліотеки. У таблиці 2 наведено тимчасові значення, що необхідні для обчислення розкладання p і p', використовуючи або класичний алгоритм або розглянутий вище швидкий варіант. Зазначене, використовується в усіх алгоритмах, за винятком алгоритму fastElkies. У таблиці рис. 8 також наведені значення часової складності для відновлення g відносно сум, спочатку з використанням класичного квадратичного алгоритму, а потім за допомогою швидкого експотенціювання. Вказана можливість реалізована в алгоритмах Elkies1992, і Elkies1998 і його модифікаціях.

ℓ	Computing $\wp$ and $\tilde{\wp}$		
	order	quadratic	fast
1013	511	8.6	7.0
2039	1024	34.6	29.9
3019	1514	75.7	30.3
4001	2005	132.7	31
5021	2515	209.3	64.4

Рис. 6. Таблиця складності обчислення p і $p^{\smile}$

ℓ	Computing $\wp$ and $\tilde{\wp}$		
	order	quadratic	fast
1013	511	8.6	7.0
2039	1024	34.6	29.9
3019	1514	75.7	30.3
4001	2005	132.7	31
5021	2515	209.3	64.4

Рис. 7. Таблиця відновлення g від своїх степеневих сум

У таблицях рисунків 8 і 9 наведені тимчасові значення складності ля алгоритмів Elkies1992 і Elkies1998 та варіації fastElkies.. У таблиці рис. 8, стовпці μ і p_i вказують на час, що необхідний для обчислення коефіцієнтів $\mu_{i,j}$ і сум p_i. У таблиці рис. 9, стовпець h_i вказує час, що трититься для обчислення коефіцієнтів h_i з раціональною функцією N/D, та з використанням квадратичного алгоритму Elkies1998 та алгоритму fastElkies. Наступна колонка дає час, який використовується для обчислення статечні суми p_i від h_i з використанням повторення (10).

В таблицях рис. 10 і 11 наведено значення часової для реалізації оригінального алгоритму Atkin в Atkin1992, а також більш швидку версію AtkinModComp з використанням модульного складання.

У таблиці рис. 11, стовпець "експонентний" стосується обчислення exp(F), з використанням швидшо-

го алгоритму; стовпець p^k містить значення часу для обчислення всіх серій $p(z)^k$, а стовпець g – для відновлення коефіцієнтів g на основів сум.

У таблиці рис. 11 наведено дані, що отримані з використанням ModComp1 і ModComp2; передостанній стовпець дає час для обчислення exp (F) і

ℓ	Elkies1992 $\wp, \tilde{\wp}$	μ	p_i	g
1013		10.4	4.4	
2039	See	49.1	17.9	See
3019	Table 2	130.6	38.9	Table 3
4001		263	68.4	
5021		496.5	106.6	

Рис. 8. Таблиця складності алгоритму Elkies1992

ℓ	Elkies1998 and fastElkies h_i quadratic	fast	p_i	g
1013	4.4	4.5	0.05	
2039	17.3	9.6	0.1	See
3019	38.0	19.5	0.16	Table 3
4001	67.2	20.0	0.21	
5021	105.0	40.7	0.27	

Рис. 9. Таблиця складності алгоритмів Elkies1998 і fastElkies

для обчислення ступенів I; останній стовпець містить значення часу виконання остаточного множення.

Аналіз показав, що асимптотично алгоритм ModComp2 є більш швидким, ніж алгоритм ModComp1, так що значення часу в таблиці рис. 12, можуть стати несподіванкою. Цей факт можна пояснити тим, що для проблемних розмірів, що нас цікавлять, переважаючі стадії алгоритму Mod-Comp1 засновані на поліноміальних операціях. Водночас крок заснований на лінійній алгебрі займає лише близько 10% операцій від усього часу обчислень. Таким чином, практична складність цього алгоритму в розглянутому діапазоні ($1000 < l < 6000$) пропорційно $M(l)\sqrt{l}$, в той час як алгоритму ModComp2 пропорційна $M(l)\sqrt{l \log l}$. Крім того, коефіцієнт пропорційності менше у вбудованій в NTL функції виконання ModComp1, ніж в реалізації ModComp2.

Також необхідно враховувати, що в таблицях рис. 6 – 11, тайминги відображають вже згадувану поведінку FFT, коли поліном множення в діапазоні 1024 – 2047 приблизно вдвічі швидший, ніж в діапазоні 2047 – 4095 і приблизно в чотири рази швидший, ніж в діапазоні 4096 – 8191.

У таблиці рис. 12 наведені значення часу для алгоритму Stark1972; окремо від загальних обчислень p і $p^{\smile}$, виділено час, необхідний для обчислення всіх зворотних (квадратичний алгоритм та швидкі інверсії), а також визначення многочленів q_n.

ℓ	Algorithm Aktin1992 $\wp, \tilde{\wp}$	exponential naive	fast	$\wp^k$	g
1013		88.4	1.2	72.3	4.4
2039	See	370.1	4.9	304.9	17.7
3019	Table 2	955.9	5.1	755.8	38.9
4001		1503	5.2	1218.9	67.6
5021		3180	10.8	2506.4	108.7

Рис. 10. Таблиця оригінального алгоритму Atkins, варіація для exp(F)

ℓ	Algorithm AtkinModComp $\wp, \tilde{\wp}$	exp(F)	$T^{1-\ell}$	modular composition ModComp1	ModComp2	g
1013		1.2	2.7	14.3	35.6	0.2
2039	See	2.5	6.6	45.8	111.9	0.4
3019	Table 2	5.1	10.4	95.3	241	0.7
4001		5.2	11.6	143.2	338	0.9
5021		10.9	20.9	240	642	1.4

Рис. 11. Таблиця складності алгоритму Atkins з модульним складом

ℓ	$\wp, \tilde{\wp}$	Inverses quadratic	fast	q_n
1013		23542	1222.7	28.0
2039	See	> 100000	5113.4	116.9
3019	Table 2		12182	258
4001			20388	418.6
5021			38910	663.1

Рис. 12. Алгоритм Stark1972

У перспективі важливим є порівняння ЕП з іншими кандидатами на постквантові алгоритми ЕП.

ВИСНОВКИ

Складність аналізу алгоритмів, що розглянуті, для випадків великих простих характеристик і для

досить великої l ізогенії, новий $O(M(l))$ алгоритм кращий, ніж раніше відомі.

Поточну реалізацію алгоритму можна додатково оптимізувати, щоб зробити цей алгоритм більш швидким для менших значень ступеня. Справді, відомо, що алгоритми, засновані на ітераційних операціях Ньютона представляють певні скорочення штатів (коефіцієнти що можна передбачити заздалегідь, повторних множників). Видалення цих залишків здійснено в [4, 12], що дозволяє досягти постійного фактора прискорень. На сьогодні, існуючі реалізації спираються лише частково на ці методи; вважається, що подальші зусилля програмування принесе практичні поліпшення.

Інший напрямок для майбутньої роботи полягає в адаптації наведених методів у разі невеликої характеристики. У зв'язку з цим, зміна останньої фази алгоритму Joux and Lercier [13], є багатообіцяючим шляхом пошуку.

Література

[1] *C. Alonso, J. Gutierrez, and T. Recio*. A rational function decomposition algorithm by nearseparated polynomials. Journal of Symbolic Computation, 19(6):527–544, 1995.

[2] *A. O. L. Atkin*. The number of points on an elliptic curve modulo a prime (II). Available at http://listserv.nodak.edu/archives/nmbrthry.html,.

[3] *D. J. Bernstein*. Composing power series over a finite ring in essentially linear time. Journal of Symbolic Computation, 26(3):339–341, 1998.

[4] *D. J. Bernstein*. Removing redundancy in high-precision Newton iteration, 2000. ффfАvailable on-line at http://cr.yp.to/fastnewton.html.

[5*] I. Blake, G. Seroussi, and N. Smart*. Elliptic curves in cryptography, volume 265 ффfof London Mathematical Society Lecture Notes Series. Cambridge University ффfPress, 1999.

[6] R. *P. Brent*. Multiple-precision zero-finding methods and the complexity of ффfelementary functionevaluation. In Analytic computational complexity, pages ффf151–176. Academic Press, New York, 1976. Proceedings of a Symposium held ффfat Carnegie-Mellon University, Pittsburgh, Pa., 1975.

[7] *R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun*. Fast solution of Toeplitz systems ффfof equations and computation of Pad´e approximants. Journal of Algorithms, ффf1(3):259–295, 1980.

[8] *D. G. Cantor and E. Kaltofen*. On fast multiplication of polynomials over ффfarbitrary algebras. Acta Informatica, 28(7):693–701, 1991.

[9] *J.-M. Couveignes, L. Dewaghe, and F. Morain*. Isogeny cycles and the Schoof-ффfElkies-Atkin algorithm. Research Report LIX/RR/96/03, LIX, April 1996. ффfАvailable at http://www.lix. polytechnique.fr/Labo/Francois.Morain/.

[10] *N. D. Elkies*. Explicit isogenies. Draft, 1992.

[11] *N. D. Elkies*. Elliptic and modular curves over finite fields and related computational issues. In D. A. Buell and J. T. Teitelbaum, editors, Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A. O. L. Atkin, volume 7 of AMS/IP Studies in Advanced Mathematics, pages 21–76. American Mathematical Society, International Press, 1998.

[12] *G. Hanrot, M. Quercia, and P. Zimmermann*. The middle product algorithm, I. Speeding up the division and square root of power series. Applicable Algebra in Engineering, Communication and Computing, 14(6):415–438, 2004.

[13] *A. Joux and R. Lercier*. Counting points on elliptic curves in medium characteristic. Cryptology ePrint Archive, Report 2006/176, 2006. http://eprint.iacr.org/.

[14] *E. Kaltofen and V. Shoup*. Subquadratic-time factoring of polynomials over finite fields. Mathematics of Computation, 67(223):1179–1197, 1998.

[15] *K. S. Kedlaya*. Counting points on hyperelliptic curves using Monsky- Washnitzer cohomology. Journal of the Ramanujan Mathematical Society, 16(4):323–338, 2001.

[16] *V. M¨uller*. Ein Algorithmus zur Bestimmung der Punktanzahl elliptischer Kurven ¨uber endlichen K¨orpern der Charakteristik gr¨oßer drei. PhD thesis, Technischen Fakult¨at der Universit¨at des Saarlandes, 1995.

[17] *A. Sch¨onhage and V. Strassen. Schnelle* Multiplikation großer Zahlen. Computing, 7:281–292, 1971.

[18] *R. Schoof*. Counting points on elliptic curves over finite fields. Journal de Th´eorie des Nombres de Bordeaux, 7(1):219–254, 1995.

[19] *V. Shoup*. A new polynomial factorization algorithm and its implementation.Journal of Symbolic Computation, 20(4):363–397, 1995.

[20] *V. Shoup*. The Number Theory Library. 1996–2005. http://www.shoup.net/ntl.

[21] *H. M. Stark*. Class-numbers of complex quadratic fields. In W. Kuyk, editor, ффfModular functions of one variable I, volume 320 of Lecture Notes in Mathematics, pages 155–174. Springer Verlag, 1973. Proceedings International Summer School University of Antwerp, RUCA, July 17-Agust 3, 1972.

Пономар Володимир Андрійович, аспірант факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Область наукових інтересів: криптографічні перетворення, безпечне програмування, методи багатофакторної автентифікації та їх застосування з метою захисту інформації, захист криптографічних засобів інформації

Бережний Олександр Григорович, студент факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Область наукових інтересів: криптографічні перетворення, швидкі алгоритми для обчислення ізогеній на еліптичних кривих.

УДК 681.3.06

Быстрые алгоритмы для вычисления изогений эллиптических кривых / В.А. Пономарь, А.Г. Бережной // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, №. 3 – С. 203 – 214.

В работе рассматриваются и анализируются алгоритмы вычислений изогений эллиптических кривых над конечным полем. Анализируется алгоритм вычисления изогений выбраной степени, который базируется на быстрых алгоритмах разложения β –функции в ряд Вейерштрасса.

Ключевые слова: эллиптические кривые, изогении эллиптических кривых, быстрые алгоритмы вычисления изогений.

Ил. 12. Библиогр.: 21 назв.

UDC 681.3.06

Fast algorithms for calculating isogeny of elliptic curves / V.A. Ponomar, O. G. Berezhnyi // Applied Radio Electronics: Sci. Journ.. – 2016. – Vol. 15, №. 3. – P. 203 – 214.

The paper deals with algorithms for calculating isogeny of elliptic curves over finite fields. The fast algorithm for calculating isogeny with a chosen degree is analyzed, which is based on the fast algorithms of β - function decomposition into Weierstrass row.

Keywords: elliptic curves, isogeny of elliptic curves, fast algorithms for calculating isogeny.

Fig. 12. Ref.: 21 items.