УДК 681.3.06

ЗБИТНЕВ С.И., к-т техн. наук, В.Ю. КОВТУН

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ В ЯКОБИАНЕ ГИПЕРЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Введение

Большинство современных криптографических стандартов основываются на преобразованиях в группе точек эллиптической кривой (ЭК). Постоянный рост мощностей современных вычислительных систем, популярность GRID вычислений приводит к ужесточению требований к существующим криптографическим преобразованиям. Такое развитие событий заставляет искать уязвимости, эффективные атаки, исследовать перспективные трудно-решаемых задач, которые в будущем смогли бы быть положены в основу криптографических стандартов. На сегодняшний день большое количество исследовательских работ посвящено криптографическим преобразованиям на гиперэллиптической кривой (ГЭК), как наиболее перспективным.

Целью данной работы является проведение классификации, изучения и оценки сложности известных методов криптоанализа преобразований на ГЭК, что позволит правильно расставить акценты для последующего повышения их эффективности, выявить потенциально уязвимые классы кривых, которые не следует применять в криптографических целях.

Для понимания излагаемого далее материала напомним некоторые основные определения.

1. Некоторые теоретические сведения

Теоретические положения лежащие в основе криптографии на ГЭК [27, 36, 40] отличаются от положений криптографии на ЭК, поэтому будет уместным напомнить некоторые базовые определения.

Пусть k - поле и $\overline{k}$ - его алгебраическое замыкание. *Гиперэллиптической кривой C рода g над k* $(g \ge 1)$ называется множество точек (u, v) удовлетворяющих уравнению:

$$C : v^2 + h(u)v = f(u) \text{ в } k[u, v], \tag{1}$$

где $h(u) \in k[u]$ - полином степени g, $f(u) \in k[u]$ - монический полином степени $2g+1$, причем исключаются точки $(u, v) \in \overline{k} \times \overline{k}$, которые одновременно удовлетворяют и (1) и частным производным (1) по соответствующим переменным $2u + h(u) = 0$, $h'(u)v - f'(u) = 0$.

Дивизором D называется формальная сумма точек кривой C вида:

$$D = \sum_{P \in C} m_P P, \; m_P \in \mathbf{Z}, \tag{2}$$

где только конечное число m_P - кратностей точек отлично от нуля.

Степенью дивизора D, $\deg(D)$, называется целое число равное $\sum_{P \in C} m_P$. Через $\mathbf{D}^0$ обозначим множество дивизоров степени 0.

Порядком дивизора D в точке P называется целое число m_P, $\operatorname{ord}_P(D) = m_P$.

Основным дивизором называется дивизор $D = \operatorname{div}(R)$, $D \in \mathbf{D}^0$ некоторой рациональной функции $R \in \overline{k}(C)^*$. Множество основных дивизоров кривой C над полем $k < L < \overline{k}$, обозначается $P_C(L) = \{\operatorname{div}(R) : R \in L(C)\}$, $P_C(L) \subset \mathbf{D}^0$.

Группа основных дивизоров кривой C обозначается $\mathbf{P}$ или $P(C)$, причем $P(C)=P_C(\bar{k})$. *Якобианом кривой* C называется фактор группа $\mathbf{J}=\mathbf{D}^0/\mathbf{P}$ или $J(C)=Div^0(C)/P(C)$. Если $D_1, D_2 \in \mathbf{D}^0$ принадлежат одному *классу эквивалентности дивизоров* $[D]$ в $J_C(L)$, в этом случае D_1 и D_2 являются *эквивалентными* дивизорами $D_1 \sim D_2$, причем $D_1 - D_2 \in \mathbf{P}$.

Дивизором рациональной функции $R=G/H$ является основной дивизор вида $\mathrm{div}(G(u,v))-\mathrm{div}(H(u,v))$.

Носителем дивизора $D=\sum_{P\in C} m_P P$ называется множество $\mathrm{supp}(D)=\{P\in C \mid m_P \neq 0\}$.

Почти приведенным дивизором называется дивизор вида $D=\sum m_i P_i-\left(\sum m_i\right)\infty$, где все $m_i \geq 0$, P_i - конечные точки $P_i \in \mathrm{supp}(D)$, причем $\widetilde{P}_i \notin \mathrm{supp}(D)$, за исключением $P_i=\widetilde{P}_i$, т.к. $m_i=1$.

Приведенным дивизором называется почти приведенный дивизор $D=\sum m_i P_i-\left(\sum m_i\right)\infty$ для которого верно $\sum m_i \leq g$.

Для каждого дивизора $D\in\mathbf{D}^0$ существует почти приведенный дивизор $D_1\in\mathbf{D}^0$, такой, что $D\sim D_1$. Для удобства, почти приведенные дивизоры $D=\sum m_i P_i-\left(\sum m_i\right)\infty$, $P_i=(x_i,y_i)\in C_f$ представляются в предложенной Мамфордом форме $D=\mathrm{div}(a,b)=\gcd(\mathrm{div}(a(u)),\mathrm{div}(b(u)-v))$, где $a(u)=\prod(u-x_i)^{m_i}$ и $b(u)$ - уникальный полином удовлетворяющий условиям: (i) $\deg_u(b)<\deg_u(a)$; (ii) $b(x_i)=y_i$, для всех i таких, что $m_i\neq 0$; (iii) $a(u)\,|\,\left(b(u)^2+b(u)h(u)-f(u)\right)$.

Степенью дивизора D в этом случае является $\deg_u(a)$.

Задача дискретного логарифмирования на гиперэллиптической кривой (ЗДЛГЭК) формулируется для ГЭК $C(\mathbf{F}_q)$ рода g и приведенных дивизоров $D_1, D_2 \in J(\mathbf{F}_q)$ следующим образом: необходимо найти решение уравнения $D_2=lD_1$ относительно l.

Решение ЗДЛГЭК является нетривиальной задачей и принадлежит классу $NP\cap co-AM$, где AM -класс алгоритмов, который имеют постоянное число итераций в игре Arthur-Merlin, в то время как ЗДЛЭК принадлежит четко классу $NP\cap co-NP$ [29, 48].

Общий алгоритм индексного исчисления (ИИ) считается одним из наиболее универсальных, на сегодняшний день, алгоритмом решения задачи дискретного логарифма (ЗДЛ), который обладает субэкспоненциальной сложностью [26].

Алгоритм 1. Индексного исчисления в общем виде.

Вход: $a, b \in G$, причем $a^{\lambda}=b$.

Выход: $\lambda \in \mathbf{Z}$.

1. Выбор базы разложения: $S=\{g_1,\ldots,g_k\}\subset G$, где $g_1=a$, $g_2=b$.

2. Генерация соотношений вида: $g_1^{e_i 1}\cdots g_k^{e_i k}=1$, $1\leq i\leq l$, до тех пор, пока $\mathrm{rank}(\|e_{ij}\|)=k-1$.

3. Решение системы линейных уравнений (СЛУ): $A\vec{x}\equiv 0\ (\mathrm{mod}\,|G|)$, $A=\|e_{ij}\|$.

4. Выделение решения: Зная $\vec{x}=(x_1,\ldots,x_k)$, находится $\lambda\cdot\vec{x}\equiv(\log g_1,\ldots,\log g_k)\bmod|G|$, для некоторого $\lambda\in\mathbf{Z}$. Дискретный логарифм соответствующий g_1, вычисляется как $\lambda\equiv x_1^{-1}\bmod|G|$, при условии его существования.

Дальнейший анализ показывает, что величина k является предметом оптимизации алгоритма, т.е. если k маленькое, сложность поиска соотношений является высокой, с другой стороны, если k большое, то сложность решения СЛУ является высокой.

Этап генерации соотношений обладает внутренним параллелизмом, т.е. интервал, которому принадлежит решение, может быть разделен и просеян независимо. Параметром оптимизации данного подхода является алгоритм координации просеивания между подинтервалами.

Решение СЛУ над $\mathbf{F}_p$, $\log_2 p > 160$, для решения ЗДЛ стандартными методами, как известно, требует значительного количества памяти и вычислительных ресурсов. Поэтому были в [31] были адаптированы метод Ланкзоса решения разряженных СЛУ и сопряженный градиентный метод.

Широкое применение получили следующие методы решения СЛУ [25]:

- метод Гаусса;
- структурированный метод Гаусса;
- метод Вейдмана [32];
- точный итеративный метод Lanczos'a;
- точный итеративный сопряженный градиентный метод;
- метод бидиагонализация (Голуб, Кахан) для поля общего вида;
- бидиагонализация (Голуб, Кахан) для простого поля;
- оптимизированный метод Ланкзоса для простого поля:
    - сопряженный метод;
    - ортогональный метод.

Верхняя граница сложности алгоритмов решения СЛУ составляет $O\left(k^2 + k\omega\right)$, где ω - количество отличных от нуля элементов матрицы $A = \left\| e_{ij} \right\|$.

Подходы к решению ЗДЛГЭК

Для непосредственного применения алгоритма ИИ к ЗДЛГЭК, требуется учесть специфику группы (якобиана), в которой ищется решение. С этой целью проведем аналогию между понятиями поля и якобиана, являющими базовыми для алгоритма ИИ.

Простой дивизор называется дивизор $D = \mathrm{div}(a, b)$, состоящий лишь из одной точки, у которого полином $a = u - p$ является неприводимой полиномиальной функцией из $\overline{k}(u)$. Уравнение $v^2 + h(u)v - f(u) \equiv 0 \pmod a \Leftrightarrow v^2 + h(p)v - f(p) = 0$ в точке $u = p$ имеет два решения $b_1 \in \overline{K} \cong \overline{K}[u]/\mathrm{div}(a)$ и $b_2 = -b_1 - h(p)$. Решения соответствуют простому дивизору $P = (p, b_1)$ и его сопряжению $\overline{P} = (p, -b_1 - h(p))$, соответственно. Дивизоры, P и $\overline{P}$ называют *расщепленными*, если $P \neq \overline{P}$, иначе *разветвленными*.

Дивизор D является t*-гладким*, если его можно представить в виде $D = \sum_{i=0}^{L} e_i \mathrm{div}(a_i, b_i)$ для $e_i \in \mathbf{Z}$ и $\max\{\deg a_i\} \le t$, т.е. для него справедливо разложение $a = a_1^{e_1} a_2^{e_2} \ldots a_L^{e_L}$, $b_i = b \bmod a_i$ на монические неприводимые полиномиальные функции над k.

В основе всех алгоритмов ИИ для решения ЗДЛГЭК лежит одна из трех стратегий [27]:

1. В основе первой стратегии лежит идея, предложенная Hafner'ом и McCurley'ом (HM) для ЗДЛ в полях мнимых квадратичных чисел. Напомним, что структура $J_C(k)$ якобиана кривой C, определяется как прямая сумма циклических подгрупп. Представления дивизоров D_1 и D_2 задается посредством прямой суммы циклических подгрупп, тогда решение ЗДЛ сводится к решению СЛУ, для найденных представлений дивизоров D_1 и D_2, составленной с использованием обобщенной китайской теоремы об остатках.

База разложения $S = \{P_1, P_2, \ldots, P_n\}$ включает все расчлененные и разветвленные простые дивизоров $P_i = \mathrm{div}(a_i, b_i)$, где $\deg a_i \le t$ для некоторой границы гладкости t. Если неприводимый полином a_i расчленяемый, тогда только один из двух простых дивизоров над a_i: $\mathrm{div}(a_i, b_i)$ или $\mathrm{div}(a_i, -h - b_i)$ включаются в S. Генерация m, $m > n$ соотношений $\sum_j e_j P_j \sim 0$ осуществляется посредством поиска t-гладких основных дивизоров.

Если S образует $J_C(k)$, то отображение $\phi : \mathbf{Z}^n \to J_C(k)$, где $\phi(e_1, e_2, \ldots, e_n) \mapsto \sum_j e_j P_j$, является сюрьективным гомоморфизмом.

Каждое соотношение дает элемент $\vec{e_i} = (e_{i1}, e_{i2}, \ldots, e_{in}) \in \ker(\phi)$ и если множество из m отношений формирует полную систему $\ker(\phi)$, то, согласно теоремы Лагранжа [28], верно отношение эквивалентности $J_C(k) \cong \mathbf{Z}/d_1\mathbf{Z} \oplus \cdots \oplus \mathbf{Z}/d_n\mathbf{Z}$, где $(d_1, d_2, \ldots, d_n)$ - диагональные элементы нормальной формы Смита (SNF) матрицы соотношений $A = (\vec{e_1}\vec{e_2}\ldots\vec{e_n})$ (отношения записаны как столбцы матрицы A). Вычисление образующего элемента $X_i = \sum_{j=1}^{n} p_{ji} P_j$ для каждой циклической подгруппы $\mathbf{Z}/d_i\mathbf{Z}$ осуществляется посредством отыскания матриц преобразований с общим модулем $P = (p_{ij})$ и $Q = (q_{ij})$, таких, что $P^{-1}AQ = \mathrm{SNF}(A)$.

Далее для дивизоров $D_1 \sim \sum \alpha_i' P_i$, $D_2 \sim \sum \beta_i' P_i$ ищется представление $(\alpha_1' \ldots \alpha_n') = P^{-1}(\alpha_1 \ldots \alpha_n)^T$ и $(\beta_1' \ldots \beta_n') = P^{-1}(\beta_1 \ldots \beta_n)^T$ в якобиане $J_C(k) \cong \mathbf{Z}/d_1\mathbf{Z} \oplus \cdots \oplus \mathbf{Z}/d_n\mathbf{Z}$ посредством разложения над S. ЗДЛ состоит в решении СЛУ $\alpha_i' \equiv \beta_i' \pmod{d_i}$, $1 \le i \le n$. для нахождения $l \in \mathbf{Z}$.

2. Отличие второй стратегии от первой состоит в том, что считается известным порядок якобиана $\#J_C(k)$. После того, как будет найдено $n+1$ различных соотношений, это отличие позволяет заменить вычислительно сложное нахождение SNF, решением СЛУ по модулю $\#J_C(k)$ для нахождение нетривиальных линейных комбинаций $\sum_{i=1}^{n+1} \gamma_i \vec{e_i} = (0, 0, \ldots, 0)$, т.е. $\sum_{i=1}^{n+1} \gamma_i R_i = 0$. На их основе составляется $\sum_{i=1}^{n+1} \gamma_i (\alpha_i D_1 + \beta_i D_2) = 0$ и решением ЗДЛ будет $\log_{D_1} D_2 = -\frac{\sum \gamma_i \alpha_i}{\sum \gamma_i \beta_i} \pmod{\#J_C(k)}$.

3. Гибридная стратегия. Эта стратегия была предложена в работе [1] для поля квадратичных чисел, и может быть легко адаптирована к ЗДЛГЭК. Она является комбинацией двух предыдущих стратегий. Аналогично второй стратегии производится замена вычисления SNF на решение СЛУ над полем целых чисел по модулю $\#J_C(k)$. Ее отличие от предыдущих состоит в том, что генерируется n соотношений вида $\sum_j e_{ij} P_j = 0$, и два дополнительных соотношения $-D_1 + \sum_j e_{(n+1)j} P_j = 0$, $D_2 + \sum_j e_{(n+2)j} P_j = 0$. На этих соотношений основе формируется матрица соотношений:

$$B = \begin{bmatrix} A & \overrightarrow{e_{n+1}} & \overrightarrow{e_{n+2}} \\ \vec{0} & 1 & 0 \\ \vec{0} & 0 & 1 \end{bmatrix},$$

где $A = (\vec{e_1}\vec{e_2}\ldots\vec{e_n})$ - матрица, состоящая из первых n соотношений записанных в столбцах. Таким образом, равенство $D_1 = lD_2$ справедливо, если и только если существует решение $\vec{x} \in \mathbf{Z}^{n+2}$ матричного уравнения $B\vec{x} = (0, \ldots, 0, 1, l)$. Решение ЗДЛ l (не обязательно

наименьшее) получается из уравнения $l = \vec{b} \cdot \vec{y}$, где $B = \begin{bmatrix} B' \\ \vec{b} \end{bmatrix}$ и $\vec{y} \in \mathbf{Z}^{n+2}$ - решение уравнения $B'\vec{y} = (0, \ldots, 0, 1)$. Эффективность этой стратегии определяется существованием метода генерации соотношений вида $\sum_j e_j P_j = 0$, который быстрее чем метод для генерации $\sum_j e_j P_j = \alpha D_1 + \beta D_2$.

Укажем условия существования, для описанных выше стратегий [27], субэкспоненциального алгоритма со сложностью $O(L_n[c]) = \exp\left((c + o(1))\sqrt{\log n \log\log n}\right)$:

1. дивизоры из S должны образовывать $J_C(k)$ и их количество составляет $n \in O\left(L_{q^g}[\rho]\right)$ для некоторой константы $\rho > 0$;

2. количество случайно выбранных дивизоров будет равно $O\left(L_{q^g}\left[\frac{a}{\rho}\right]\right)$ для некоторой константы $a \in (0, 1)$, перед тем, как будет найден t-гладкий дивизор (допущение, что размер базы разложения составит $O\left(L_{q^g}[\rho]\right)$).

Оптимизация алгоритма ИИ заключается в поиске значения ρ, такого, что решение СЛУ и генерация соотношений будет обладать сравнительно одинаковой сложностью. Согласно первому условию $n \in O\left(L_{q^g}[\rho]\right)$, сложность решения СЛУ составит $O\left(L_{q^g}[l\rho]\right)$, для некоторой константы $l > 0$, причем непосредственно решение СЛУ размером $n \times n$ составит $O\left(n^l\right)$. Сложность генерации соотношений зависит и от второго предположения о гладкости и, в частности:

- от сложности генерации дивизора и тестирования одного дивизора на гладкость;
- от количества гладких дивизоров необходимых для решения ЗДЛ.

Алгоритм решение ЗДЛ, построенный по описанным выше стратегиям, является вероятностным субэкспоненциальным алгоритмом типа Лас-Вегас (если решение получено, то оно гарантировано корректное). Тест ЗДЛ на разрешимость осуществляется проверкой образует ли множество соотношений ядро $\ker(\phi)$. Для этого проверяется равен ли определитель h' матрицы соотношений порядку якобиана $\#J_C(k)$. Если $\#J_C(k)$ - неизвестен, то проверяется неравенство $H^* < h' < 2H^*$, где $H^* < \#J_C(k) < 2H^*$. Определитель h' может быть эффективно вычислен посредством точной и эвристической оценки границы количества классов эквивалентности в поле гиперэллиптических функций [2].

Практические реализации алгоритмов ЗДЛГЭК

Для перехода к реальным оценкам сложности решения ЗДЛГЭК потребуется более детально изложить известные по литературе алгоритмы решения, согласно приведенным выше стратегиям.

Алгоритм ADH (Adleman, DeMarrais, Huang)

Алгоритм ADH впервые был предложен как алгоритм ИИ для решения ЗДЛГЭК. Он был описан для случая, когда q - нечетное простое. Соответствует стратегии 1. Генерация соотношений осуществляется посредством поиска разложения D_1 и D_2 в S на случайные почти приведенные основные дивизоры, образованные рациональными функциями вида $Av + B$, где $A, B \in k[u]$. Каждый такой гладкий дивизор получается из соотношения $(A_i v + B_i) = \sum_j e_{ij} P_j \sim 0$. Необходимо сгенерировать такое количество соотношений, при котором ранг матрицы соотношений будет равен n, это в свою очередь приведет к успешному решению ЗДЛГЭК согласно стратегии 1.

Разложение основного дивизора $(Av+B)$ на простые дивизоры $(Av+B)=\sum_j e_j P_j$ выполняется посредством разложения нормы функции $N(Av+B)$, т.е. $N(Av+B)=B^2+ABh-A^2=\prod_{j=1}^{n} a_j^{e'_j}$, где $P_j=\mathrm{div}(a_j,b_j)$ - простые дивизоры из S. Известно, что $e_j=e'_j$, если $Ab_j+B\equiv 0(\mathrm{mod}\, a_j)$, иначе $e_j=-e'_j$.

Если $\log q\le(2g+1)^{0.98}$, то алгоритм ADH как изложено в [6], имеет сложность $O\left(L_{q^{2g+1}}[c]\right)$ для константы $0<c<2.313$. При получении этого результата в [6] сделаны следующие предположения:

- $J_C(k)$ образован из всех простых дивизоров степени не более $\log_q L_{q^{2g+1}}[c]$, для некоторой константы $c>0$.

- $D=\mathrm{div}(a,b)$ считается случайным дивизором. Вероятность, что D является t-гладким равна вероятности, что случайный полином степени $\deg a$ над k раскладывается на множители степени не более t.

В [6] сделан вывод, что количество попыток тестирования соотношения на пригодность составит $O\left(L_{q^{2g+1}}\left[\frac{9}{16c}\right]\right)$ при условии, что $\log q\le(2g+1)^{0.98}$.

Известно, что сложность решения СЛУ и генерации соотношений составляет $O\left(L_{q^{2g+1}}[lc]\right)$ и $O\left(L_{q^{2g+1}}\left[\frac{9}{16c}\right]\right)$, соответственно, где оптимальное значение было достигнуто при $c=\frac{3}{4\sqrt{l}}$. Количество итераций при которых выполняется решение ЗДЛГЭК (генерация соотношений и решение СЛУ) получится равным $O\left(L_{q^{2g+1}}[c]\right)$, откуда общая сложность алгоритма составил $O\left(L_{q^{2g+1}}[c]\right)$ при $c=\frac{3(l+1)}{4\sqrt{l}}$ [6]. Полезно будет отметить, выводы из результатов анализа [7] этого алгоритма свидетельствуют о том, что для решения СЛУ необходимо $l<7.376$ и $c=2.313$.

Алгоритм MST (Muller, Stein, Thiel)

Алгоритм MST [8] является развитием идей ADH, который применим к полям нечетной характеристики. В этом алгоритме не учитывается влияние отношения $g/\log q$ на сложность алгоритма [15]. Алгоритм соответствует стратегии 1. Для поля нечетной характеристики [9] и поля четной характеристики поля [10] было получено, что ЗДЛЭК полиноминально сводится к так называемой инфраструктуре ЗДЛ в действительном квадратичном поле функций. В свою очередь, ЗДЛГЭК эквивалентна инфраструктуре ЗДЛ над константным расширением поля [11].

Сделаем некоторые пояснения связанные с рассматриваемым подходом. Рассматриваем конечное поле нечетной характеристики $k=\mathbf{F}_q$ и кривую $C: v^2=f$, где $f\in k[u]$ - монический, $\deg_u(f(u))=2g+2$, тогда $L=k(C)=k(u)\left(\sqrt{f}\right)$ - действительное квадратичное поле функций. *Инфраструктурой ЗДЛ* (ИЗДЛ) называется задача нахождения $\deg(\alpha)$ приведенного основного идеала $I_a=aK[u]+(b+v)K[u]$ и $I_a=(\alpha)$ для заданных полиномов $a,b\in k[u]$, т.е. необходимо определить расстояние $\delta(I_a,O)$. Заметим, что существует уникальный, вплоть до константы, образующий элемент для I_a, такой, что $0\le\deg(\alpha)\le R$.

На первом этапе формируется база разложения $S=\{I_{p_1},I_{p_2},\ldots,I_{p_n}\}$ из всех расчлененных и разветвленных простых идеалов P_i с $\deg P_i\le t$ для некоторой границы гладкости t. Отыскиваются m, $m>n$ t-гладких основных идеалов, которые образуют соотношения вида $I_b=\prod_j I_{p_j}^{e_j}=(\beta)$. Каждое такое соотношение дает элемент

$\vec{v}_j = \left(e_{j1}, e_{j2}, \ldots, e_{jn}, \deg\left(\beta_j\right)\right) \in \mathbf{Z}^{n+1}$, Γ_t - множество всех элементов $\vec{v}_j \in \mathbf{Z}^{n+1}$ и Γ_t' - множество $\vec{e}_j = \left(e_{j1}, e_{j2}, \ldots, e_{jn}\right) \in \mathbf{Z}^n$. Известно, если множество S образует группу классов идеалов, то Γ_t является $(n+1)$ размерной решеткой с дискриминантом $\hbar = \hbar' R$ и Γ_t' является n размерной решеткой с дискриминантом $\hbar'$. Детально остановимся на процессе генерации соотношений:

- случайным образом выбирается $\left(e_1, e_2, \ldots, e_n\right) \in \left\{0, \ldots, q^{2g+2}\right\}^n$ и вычисляется $I_c = \prod_j I_{p_j}^{e_j}$;

- находится приведенный идеал $I_b \sim I_c$ и $\deg(\beta)$, где $\beta \in L$ такой, что $I_b = (\beta) I_c$;

- если I_b может быть разложено в базе разложения S, как $I_b = \prod_j I_{p_j}^{v_j}$, тогда $I_b I_c^{-1} = \prod_j I_{p_j}^{(v_j - e_j)} = (\beta)$ дает искомое соотношение.

- если была поучена полная образующая система $\left\{\vec{v}_1, \ldots, \vec{v}_m\right\}$ для решетки Γ_t, то удаляя последнюю координату в каждом векторе $\vec{v}_j$, получается полная образующая система $\left\{\vec{v}_1', \ldots, \vec{v}_m'\right\}$ для решетки Γ_t'. Зададим матрицу $A = \left(\vec{v}_1 \vec{v}_2 \ldots \vec{v}_m\right)$, где векторы $\vec{v}_j$ записаны в виде столбцов матрицы, аналогично $A' = \left(\vec{v}_1' \vec{v}_2' \ldots \vec{v}_m'\right)$.

- если I_a разлагается в S как $I_a = \prod_j I_{p_j}^{a_j}$, то $\vec{a} = \left(a_1, a_2, \ldots, a_n\right) \in \Gamma_t'$, т.к. I_a - принципал. Следовательно, существует решение $\vec{x} = \left(x_1, x_2, \ldots, x_m\right) \in \mathbf{Z}^m$ для СЛУ $A'\vec{x} = \vec{a}$. Таким образом, образующий элемент для I_a является $\alpha = \prod_j \beta_j^{x_i}$ и $\deg(\alpha) = \sum_j x_j \deg\left(\beta_j\right)$.

- если I_a не разлагается в S, необходимо выполнить нахождение приведенного, основного идеала $I_b \sim I_a$, который разлается в базе. Для вычисления $\deg(\alpha)(\mathrm{mod}\, R)$, необходимо определить R посредством вычисления $\det \Gamma_t$ и аналогично $\det \Gamma_t'$, а затем вычислить $R = \det \Gamma_t / \det \Gamma_t'$.

Отличие алгоритм MST от алгоритма ADH состоит в том, что алгоритм MST не используются эвристические предположения. В него заложено, что существует образующее множество для группы классов идеалов, чья размерность является полиномом в q^g, обычно множество простых идеалов степени $\left|2\log_q(4g-2)\right|$ является образующим множеством. В дополнение, доказана точная оценка вероятности t-гладкости случайного приведенного дивизора.

Строгое описание алгоритма MST рассматривается в [8], там же показано, что если $2g+2 \geq \log q$, то при $n \in O\left(L_{q^{2g+2}}[\rho]\right)$ количество попыток, прежде чем соотношение будет найдено, составит $O\left(L_{q^{2g+2}}\left[\frac{1}{4\rho}\right]\right)$. Если решения СЛУ потребует $O\left(L_{q^{2g+2}}[5\rho]\right)$, то генерация соотношений потребует $O\left(L_{q^{2g+2}}\left[2\rho + \frac{1}{4\rho}\right]\right)$. Например, при количестве соотношений $m \in O\left(L_{q^{2g+2}}[\rho]\right)$, выполняется $O\left(L_{q^{2g+2}}[\rho]\right)$ произведений идеалов со случайной степенью, оптимальное значение $\rho = \frac{5}{2\sqrt{3}}$ и общая сложность составит $O\left(L_{q^{2g+2}}[1.44]\right)$. Исходя из полученной сложности, алгоритм MST обладает меньшей сложностью, чем ADH.

Алгоритм FP (Flassenberg, Paulus)

В работе [21] впервые предложен подход НМ [13] по аналогии с полем квадратичных функций и полем квадратичных чисел для ЗДЛ над полями нечетной характеристики, а также применение алгоритмов Полларда, Похлинга - Хелмана, ADH для решения ЗДЛ в

мнимых квадратичных полях (или ГЭК). Алгоритм соответствует стратегии 1. Процесс генерации соотношений состоит в следующем:

1. Случайным образом выбирается $(e_1, e_2, \ldots, e_n) \in \mathbf{Z}^n$ и вычисляется $D = \sum_j e_j P_j$;

2. Выполняется приведение дивизор D к D';

3. Если D' является t-гладким с $D' = \sum_j v_j P_j$, то соотношением будет $D' - D = \sum_j (v_j - e_j) P_j \sim 0$.

Отсечение не перспективных путей решения является основным методом уменьшения сложности большинства алгоритмов решения *NP*-полных задач. Наиболее ценным в предложенном алгоритме является отсеивание, основанное на свойстве: для всех $f(X)$ с коэффициентами из кольца R, в том случае если $p, x_0 \in R$ и $p \mid f(x_0)$, то $p \mid f(x_0 + ip)$, для всех $i \in R$. Заметим, что в случае $R = \mathbf{Z}$, все $x \in [-M, M]$, для которых функция $f(x)$ является гладкой, могут быть найдены со сложностью соизмеримой со сложностью тестирования функции $f(x)$ используя «решето Эратосфена».

Суть отсечения для ЗДЛГЭК состоит в том, что поиск произвольного t-гладкого дивизора $D' \sim D$ порождает искомое соотношение. Рассмотрим нормальную форму $f(X, Y) \in (k[u])[X, Y]$ дивизора $D = \mathrm{div}(a, b)$. Для $\forall x, y \in k[u]$ существует дивизор $D' = \mathrm{div}(f(x, y), b') \sim D$. Следовательно, можно отбросить полином $f(X) = f(X, 1)$ или даже $f(X, Y)$, как предложено в [14] для поиска x, таких, что $f(x)$ является t-гладким. Каждый такой полином x порождает необходимое соотношение.

В [14] показано, что сложность алгоритма FP [14] является значительно ниже сложности ADH [3].

Авторами [14] реализованы вычисления в $J_C(k)$ для кривых $v^2 = u^{2g+1} + 2u + 1$ над полями $\mathbf{F}_{11}, \mathbf{F}_{101}, \mathbf{F}_{1009}, \mathbf{F}_{10007}, \mathbf{F}_{100003}$ и рода $g \in [1, 12]$. Ими исследовались несколько вариантов варианты алгоритмов: без отсечения, с пробным делением (версия НМ), с отсечением, алгоритм больших и малых шагов. Наибольшим быстродействием среди рассмотренных алгоритмов обладала реализация с отсечением. С возрастанием рода, алгоритм больших и малых шагов показал наибольшую производительность.

Алгоритм Smart

Работа [12] посвящена реализации варианта алгоритма ADH с использованием FP решета (на основе подхода Полларда), которое аналогично алгоритму факторизации решета для поля чисел.

Приведем основные этапы алгоритма Smart для кривой вида $C: v^2 = f(u)$ над полем $\mathbf{F}_q$, q - нечетное простое:

1. Строится база разложения S. В базу S заносятся все простые дивизоры из носителей дивизоров D_1 и D_2. Затем, в отличие от ADH, заносятся все ветвящиеся дивизоры из поля квадратичных функций K. Следующими добавляется множество расщепленных простых дивизоров, степень которых не превышает g. Посредством параметра B производится равномерная регулировка количества элементов в S. Т.е. в базе разложения находится B простых дивизоров степени 1, $B/2$ дивизоров степени 2, $B/3$ дивизоров степени 3 и т.д.

2. Образование матрицы соотношений M. M содержит $n+2$ строки, причем первые 2 с соотношениями для D_1 и D_2. Матрица заполняется из следующих соображений:

а) $m_{1,1} = -1$, в остальные заносятся простые дивизоры из базы разложения, образующие D_1;

б) $m_{2,2} = -1$, в остальные заносятся простые дивизоры из базы разложения, образующие D_2;

в) в третью строку заносятся простые дивизоры из базы разложения, которые образуют $\mathrm{div}(f)$, это дает соотношения между ветвящимися дивизорами.

3. Отсечение. Необходимо найти полиномы $a, b \in \mathbf{F}_q[u]$, такие, что дивизор функции $f = av + b$ имеет носитель, в который входят лишь элементы из базы разложения. Каждой такой полиномиальной функции $f \in \mathbf{F}_q[u]$ ставится в соответствие уникальный идентификатор (образ) $e \in \mathbf{N}$ для индексирования массива отсечения. Массив отсечения – двумерный массив, индексирующий полиномиальные функции согласно их образов. Каждый элемент инициализируется значением $\deg\left(N_{K/\mathbf{F}_q[u]}(av+b)\right) = \deg\left(b^2 - a^2 f\right)$ согласно предположения [12], где a и b полиномиальные функции, чьи идентификаторы задают строку и столбец индекса массива.

4. Осечение. Производится понижение сложности алгоритма решета поля чисел посредством замены стандартного решета на решето Полларда [39].

5. Решение СЛУ $Mx = 0 \pmod n$ в поле F_n. Размеры матрицы предлагается уменьшать за счет перехода к нормальной форме Эрмита или предварительном применении структурированного метода Гаусса. В результате получается верхняя треугольная матрица с первым столбцом отличным от нуля.

6. Выделение решения. Выполняется параллельное тестирование нескольких основных дивизоров вида $(Av + B)$ на гладкость. Проведенные эксперименты показали, что алгоритм ADH с отсечениями является не эффективным на практике.

Результаты применения предложенного алгоритма для решения ЗДЛГЭК для кривых над полями $\mathbf{F}_{11}$ с родами 1-9, $\mathbf{F}_{101}$ с родами 1-3 и $\mathbf{F}_{1009}$ с родом 1 описаны в [12].

Алгоритм Pollard

Метод ρ-Полларда [3, 4] может быть эффективно реализован, для ГЭК имеющих нетривиальный эффективно вычислимый автоморфизм. Для автоморфизма σ порядка m в $J_C(k)$, G_i обозначает некоторое представление дивизора R. Если R разлагается как $R = P_1 + P_2 + \cdots + P_r$ на простые дивизоры P_i степени меньшей либо равной t, то R можно записать в виде $R = \theta^{l_1} G_1 + \cdots + \theta^{l_r} G_r$, $0 \le l_i \le m$. Этот прием позволяет уменьшить размер базы разложения в m раз, и тем самым уменьшить сложность решения СЛУ в m^2 раз [5].

Алгоритм больших и малых шагов

Подход описанный в [30] позволяет понизить сложность всех алгоритмов решения ЗДЛГЭК, благодаря информации об остатке по модулю m характеристического полинома эндоморфизма Фробениуса для якобиана ГЭК, т.е. группа классов почти приведенных дивизоров разбивается на m классов смежности. Так, сложность классического алгоритма составляет $O\left(q^{(2g-1)/4} / \sqrt{m}\right)$, в то время, как алгоритм больших и малых шагов обладает сложностью $O\left(q^{g(g+1)/8} / m^{g/2}\right)$, где $m = \Omega\left(q^{\varepsilon+(g-2)/4}\right)$, $\forall \varepsilon > 0$.

Алгоритм Bauer и Enge для произвольных конечных полей

Алгоритм ADH был одновременно обобщен для ГЭК над произвольными конечными полями в работах [6] и [15] независимо. Алгоритм [6] схож с ADH и имеет соизмеримую сложность при одинаковых эвристических допущениях. Аналогично FP, алгоритм [15, 16] аналогичен методу HM [13], где соотношения образуются без отсеивания. В отличие от алгоритма [6], алгоритм [15] основан на строго доказанных утверждениях. База разложения

образует $J_C(k)$ согласно обоснованию [8]. Доказательство положения о гладкости производится в [17]. Сложность образования базы разложения составляет $O\left(\left(g^2\log^2 q\right)L_{q^g}\left[\rho+\frac{1}{\sqrt{\upsilon}}\right]\right)\subseteq O\left(L_{q^g}\left[\rho+\frac{1}{\sqrt{\upsilon}}\right]\right)$ битовых операций. Причем известно, что простые дивизоры из $J_C(k)$, имеют степень $\left|2\log_q(4g-2)\right|$. В [15] впервые показана зависимость сложности от соотношения между g и $\log q$, что если выполняется условие $g\geq\upsilon\log q$ (условие субэкспоненциальной сложности алгоритма) для положительной константы υ, то количество соотношений составит $n\in O\left(L_{q^g}\left[\rho+\frac{1}{\sqrt{\upsilon}}\right]\right)$ и количество попыток поиска соотношения ограничено $\frac{5}{4}\sqrt[40]{e}(2g+1)L_{q^g}\left(\frac{1}{4\rho}\right)\in O\left(L_{q^g}\left[\frac{1}{2\rho}+o(1)\right]\right)$. Сложность решения СЛУ составляет $O\left(L_{q^g}\left[l\rho\right]\right)$ или $O\left(L_{q^g}\left[2\left(\rho+\frac{1}{\sqrt{\upsilon}}\right)+o(1)\right]\right)$, отсюда сложность генерации соотношений составит $O\left(n^2L_{q^g}\left[\frac{1}{4\rho}+o(1)\right]\right)\subseteq O\left(L_{q^g}\left[2\rho+\frac{2}{\sqrt{9}}+\frac{1}{2\rho}+o(1)\right]\right)$, так как требуется n соотношений. Общая сложность алгоритма составляет $O\left(L_{q^g}\left[\max\left\{5\left(\rho+\frac{1}{\sqrt{\upsilon}}\right),2\left(\rho+\frac{1}{\sqrt{\upsilon}}\right)+\frac{1}{4\rho}\right\}+o(1)\right]\right)$ [16]. Оптимизацию параметра ρ производят посредством задания функции $f:\rho\mapsto\max\left\{5\left(\rho+\frac{1}{\sqrt{\upsilon}}\right),2\left(\rho+\frac{1}{\sqrt{\upsilon}}\right)+\frac{1}{4\rho}\right\}$, которая имеет единственный глобальный минимум. Полагая, что $l=4$ для этапа решения СЛУ получается оптимальное значение $\rho=\frac{1}{2}\left(\sqrt{1+\frac{1}{\upsilon}}-\frac{1}{\sqrt{\upsilon}}\right)$ и общая сложность составит $O\left(L_{q^g}\left[c\right]\right)$ при $c=2\left(\sqrt{1+\frac{1}{\upsilon}}+\frac{1}{\sqrt{\upsilon}}\right)$. Для алгоритма [15] были получены упрощенные теоретические оценки сложности алгоритма решения ЗДЛГЭК в $O\left(\exp\left(\rho\sqrt{(g\log q)\log(g\log q)}\right)\right)$ битовых операциях, где $\rho=\frac{5}{2\sqrt{3}}\left(\sqrt{1+\frac{3}{\upsilon}}+\sqrt{\frac{3}{\upsilon}}\right)+o(1)$ для константы $\upsilon>0$, $o(1)\to 0$ при $g\log q\to\infty$. Программная реализация алгоритмов не производилась, поэтому отсутствуют результаты экспериментов.

Алгоритм Gaudry для небольшого рода ГЭК

Среди приведенных выше алгоритмов, отсутствуют алгоритмы, спроектированные с известным порядком якобиана $\#J_C(k)$. Такой алгоритм был впервые предложен в работе [5]. Он соответствует стратегии 2, и спроектирован с учетом знания порядка $\#J_C(k)$. После образования базы разложения S, куда включаются только простые дивизоры степени 1, выполняются случайные аддитивные шаги на множестве приведенных дивизоров эквивалентных $\alpha D_1+\beta D_2$ (подход Теске [18]). Каждый встречающийся 1-гладкий дивизор порождает соотношение. Шаги выполняются, пока не будет сгенерировано необходимое количество соотношений.

Эффективность процесса получения соотношений, обосновывается фактом, что каждый новый кандидат образуется посредством случайного аддитивного шага, требующего единственное сложение в $J_C(k)$. Для тестирования дивизора $\mathrm{div}(a,b)$ на 1-гладкость, проверяется равенство $\gcd\left(u^q-u,a\right)=\deg a$ (доказывается, что a не имеет кратных множителей), т.к. $\gcd\left(u^q-u,a\right)$ является произведением различных полиномов степени 1, являющихся делителями a. На практике, вероятность наличия у a кратных множителей или проигнорируют или обнаруживают, вычисляя степень $\gcd\left(\left(u^q-u\right)^g,a\right)$.

Приведем алгоритм в общем виде:

1. Построение базы разложения. Для каждого u_i над $\mathbf{F}_q$ степени 1, ищется v_i, такой, что $\mathrm{div}(u_i,v_i)\in J_C(k)$. В случае его существования, $\mathrm{div}(u_i,v_i)$ сохраняется в базе S лишь один из двух $\mathrm{div}(u_i,v_i)$ или $-\mathrm{div}(u_i,v_i)$.

2. Инициализация случайных шагов. Для $j=\overline{1,r}$ выбираются случайные $\alpha_j, \beta_j \in [1,n]$ и вычисляются шаги $T_j = \alpha_j D_1 + \beta_j D_2$. Начальная точка $R_0 = \alpha_0 D_1 + \beta_0 D_2$.

3. Вычисление $j = \mathrm{H}(R_0)$, $R_0 = R_0 + T_j$, $\alpha_0 = \alpha_0 + \alpha_j \bmod n$ и $\beta_0 = \beta_0 + \beta_j \bmod n$, до тех пор, пока $R_0 = \mathrm{div}(u_0(z), v_0(z))$ является гладким.

4. Разложение R_0 в базе S посредством факторизации $u_0(z)$ над $\mathbf{F}_q$ и определение позиций множителей в S. Результат разложения $R_k = \sum m_{ik} g_i$ заносится как строка в матрицу $M = \|m_{ik}\|$. Переопределение $\alpha_k = \alpha_0$ и $\beta_k = \beta_0$.

5. Если $k < \#S + 1$, то $k = k + 1$ и переход к п.3.

6. Решение СЛУ в поле $\mathbf{F}_n$. Поиск не нулевого вектора $\|\gamma_k\|$ из ядра матрицы M^T.

7. Если $\sum \beta_k \gamma_k \neq 0$, то $\lambda = \left(\sum \alpha_k \gamma_k\right) / \left(\sum \beta_k \gamma_k\right) \bmod n$ иначе переход к п.2.

8. Вернуть λ.

Обозначим через I_J - сложность операций в $J_C(\mathbf{F}_q)$, I_q - сложность операций в $\mathbf{F}_q$, $I_{g\cdot q}$ - сложность операций в кольце полиномиальных функций $k[C]$, I_n - сложность операций в $\mathbf{F}_n$, $n \approx q^g$.

Согласно введенным обозначениям, сложность алгоритма по пунктам составит:

1. $O(qI_q)$.
2. $O(I_J \log n)$.
3. $O(g!(I_J + I_n + I_{g\cdot q}))$.
4. $O(qI_{g\cdot q})$.

Количество итерации цикла 3-5 составит $\#S = O(q)$, сложность всего цикла $O(qg!(I_J + I_n + I_{g\cdot q})) + O(q^2 I_{g\cdot q})$.

6. Размер матрицы составляет $O(q)$, количество отличных от нуля элементов матрицы M составит $O(gq)$. Сложность решения СЛУ алгоритмом Ланкзоса составит $O(gq^2 I_n)$.

7. $O(qI_n)$.

Общая сложность алгоритма составляет: $O(g!qI_J) + O((g!q + gq^2)(I_n + I_{g\cdot q})) + O(qI_q)$.

Зафиксировав g и варьируя величиной q, в [5] показано, что алгоритм имеет сложность $O(g^3 q^2 \log^2 q + g^2 g! q \log^2 q)$ битовых операций. При аналогичных условиях алгоритмы общего вида решения ЗДЛ имеют сложность $O(q^2)$. Результаты анализа [5] показывают, что для $g > 4$, ЗДЛГЭК может быть решена эффективнее, чем позволяют алгоритмы общего вида. Исследования [5] уточняют в отношении какого рода допускается более эффективный алгоритм решения ЗДЛ, чем общий алгоритм.

В работе [5] рассматривается два примера с порядком $\#J_C(k) = 10^{40}$, не обладающие эффективно вычислимым групповым автоморфизмом в $J_C(k)$, ГЭК рода 6 над полем $\mathbf{F}_{5026243}$ и $\mathbf{F}_{2^{23}}$. Практическое применение алгоритма ρ-Полларда для заданных $J_C(k)$ оказалось не целесообразным по причине достаточно большого $\#J_C(k)$.

Алгоритм DGM (Duursma, Gaudry, Morain)

В алгоритме DGM [41], для снижения сложности получения решения ЗДРГЭК в $\sqrt{m}$, предложено использовать автоморфизм порядка t, который имеется у некоторых кривых,

раз. Кривые, для которых известен эффективно вычислимый автоморфизм и к которым был применен подход [41], приведены в таблице 2.

Таблица 2.

Автор	Кривая	Поле	Автоморфизм	Порядок автоморфизма m
Коблиц	$v^2+v=u^5+u^3$ (*)	$\mathbf{F}_{2^n}$	Фробениус+ $\begin{cases} u \mapsto u+1 \\ v \mapsto v+u^2 \end{cases}$	$4n$
	$v^2+v=u^5+u^3+u$ (*)	$\mathbf{F}_{2^n}$	Фробениус	$2n$
	$v^2+v=u^{2g+1}+u$	$\mathbf{F}_{2^n}$	Фробениус	$2n$
	$v^2+v=u^{2g+1}$	$\mathbf{F}_{2^n}$	Фробениус	$2n$
Бахлер, Коблиц, Чао и др.	$v^2+v=u^{2g+1}$ (+), $p \equiv 1(2g+1)$	$\mathbf{F}_p$	Комплексное умножение в поле $Q(\zeta_{2g+1})$	$2(2g+1)$
Сакаи, Сакураи	$v^2+v=u^{13}+u^{11}+u^9+u^5+1$ (+)	$\mathbf{F}_{2^{29}}$	Фробениус + $\begin{cases} u \mapsto u+1 \\ v \mapsto v+u^6+u^5+ \\ +u^4+u^3+u^2 \end{cases}$	$4 \cdot 29$
Дуурсма, Сакураи	$v^2=u^p-u+1$ (+)	$\mathbf{F}_{p^n}$	Фробениус + $(u,v) \mapsto (u+1, v)$	$2np$

где (*) – кривые, которые были взломаны методом Fray-Ruck [42] эффективность которого гораздо выше метода ρ-Полларда даже при наличии автоморфизма; (+) – кривые большого рода, которые были взломаны алгоритмом ИИ [5]; ς - корень пятой степени из -1.

Практическая реализация DGM приведена в работе [41] для двух кривых:

- $v^2=u^5-1$ над полем $\mathbf{F}_{31^3}$, где порядок дивизора образующего группу $n=778201$;
- $v^2+v=u^5+u^3+u$ над полем $\mathbf{F}_{2^{11}}$, где порядок дивизора образующего группу $n=599479$.

Возможно уменьшение сложности за счет уменьшения размера базы разложения [19], при соотношении «пригодных» дивизоров $1/l$, сложность генерации соотношений увеличится на множитель l^g, но потребуется в l раз меньше соотношений и сложность решения СЛУ $O\left(L_{q^{2g+1}}[lc]\right)$ уменьшится в l^2 раз. При оптимальном значении $l=O\left((q/g!)^{1/(g+1)}\right)$, общая сложность составит $O\left(q^{\frac{2g}{g+1}+\varepsilon}\right)$ для $q \to \infty$.

Алгоритм GHS (Gaudry, Hess, Smart)

В [19, 43, 51] рассматривается модифицированный алгоритм [5] для решения ЗДЛГЭК для ГЭК рода 4 над полем $\mathbf{F}_{2^{21}}$. Суть модификации состоит в использовании усеченной базы разложения, которая включает лишь простые дивизоры степени 1 обладающие нормой $v+\alpha$, причем $\alpha \equiv 0 \left(\mathrm{mod}\, x^3\right)$. К тому же, для уменьшения количества соотношений и размера матрицы соотношений, большинство гладких дивизоров $\mathrm{div}(a,b)$ отбрасываются, посредством быстрого теста $\alpha \equiv 0 \left(\mathrm{mod}\, x^3\right)$. Общая сложность алгоритма оказалась ниже на 25% чем у ρ-Полларда: $O\left(N \cdot L_{e^g}\left[\frac{1}{2}, 3.54\sqrt{\log q}\right]\right)$ групповых операций, при $g \to \infty$, $N=\deg_y C(x,y)$. Анализ сложности алгоритма показывает, что для ГЭК рода 4 можно предложить алгоритм, который будет эффективнее общего.

Алгоритм EG (Enge, Gaudry)

Авторы [20] предложили свой вариант подхода [5] для ГЭК большого рода, в которых известен порядок группы и сохраняется предположение о гладкости. Он соответствует стратегии 2. Как и в алгоритме [16], база разложения содержит все простые дивизоры степени меньшей t и тестирование на гладкость расширилось проверкой равенства $a = \mathrm{lcm}\left(\left\{\gcd\left(u^{q^i} - u, a\right) \mid i \in [1, t]\right\}\right)$ (проверка на наличие кратных множителей a может быть проведена как и описывалось в [5]).

В [38] используются случайные шаги $\alpha D_1 + \beta D_2$ [39] в $J_C(k)$ и для получения t-гладких дивизоров (все полученные гладкие дивизоры сохраняются), которые образуют соотношения $\alpha_i D_1 + \beta_i D_2 \sim R_i = \sum_j e_{ij} P_j$. После получения $k+1$ соотношений, где k количество простых дивизоров в базе разложения, решается СЛУ $\sum_{i=1}^{k+1} \gamma_i \left(e_{i1}, e_{i2}, \ldots, e_{ik}\right) = (0, 0, \ldots, 0)$ или $\sum_{i=1}^{k+1} \gamma_i R_i = 0$, одним из известных методов. На выходе имеем $\sum \gamma_i \left(\alpha_i D_1 + \beta_i D_2\right) = 0$, откуда $\log_{D_1} D_2 = -\left(\sum \gamma_i \alpha_i\right) / \left(\sum \gamma_i \beta_i\right) \bmod r$.

Используя эффект гладкости [20], в [16] доказывается утверждение, что сложность составит $O\left(L_{q^g}[c]\right)$, $c = \sqrt{2}\left(\sqrt{1 + \frac{1}{2\vartheta}} + \frac{1}{\sqrt{2\vartheta}}\right)$ при $g \geq \vartheta \log q$ для некоторой константы $\vartheta > 0$. Доказательство следует из факта, что зная порядок $J_C(k)$, возможно использование рандомизированного метода Ланкзоса для решения СЛУ вместо вычисления SNF матрицы целых чисел. Следовательно, сложность решения СЛУ $O\left(L_{q^{2g+1}}[lc]\right)$ будет минимизирована при $l = 2$, которое предпочтительнее чем 4, и может быть использовано для получения заявленной в [41, 44, 52] сложности.

Алгоритм JMS (Jacobson, Menezes, Stein)

В [21] реализована улучшенная версия алгоритма EG [20] для решения ЗДЛГЭК рода 31 над полями $\mathbf{F}_{2^2}, \mathbf{F}_{2^3}, \mathbf{F}_{2^4}, \mathbf{F}_{2^5}$. Особенностями предложенного алгоритма являются: оптимизация стратегия проверки на гладкость и выбор оптимальной границы гладкости t. Общая сложность алгоритма составляет $O\left(\exp\left((1 + o(1))\sqrt{(2g+1)\log q \log((2g+1)\log q)}\right)\right)$.

Алгоритм Theriault'a

Из [22] известна улучшенная версия алгоритма [5] для небольшого рода. Авторы [22] предлагают несколько модификаций. Первая модификация состоит в использовании лишь части простых дивизоров степени 1 в базе разложения, это в свою очередь снижает вероятность поиска гладких дивизоров, и сложность решения СЛУ снижается (имеет меньшую размерность). Решив задачу балансировки сложности алгоритмов генерации соотношений и решения СЛУ [22], было получено оптимальное количество простых дивизоров степени 1. В этом случае общая сложность алгоритма составит $O\left(g^5 q^{2 - \frac{2}{g+1} + \varepsilon}\right)$.

Вторая модификация заключается в хранении следа полностью разлагаемых в базе разложения дивизоров, за исключением единственного «большого простого» дивизора (метод «больших простых»).

В случае нахождения двух соотношений, которые или содержат одинаковые простые множители или содержат большие простые множители, являющиеся отрицанием друг друга, то такие соотношения комбинируются для формирования соотношения. Во втором случае общая сложность составит $O\left(g^5 q^{2 - \frac{4}{2g+1} + \varepsilon}\right)$.

Оба алгоритма являются асимптотически быстрее алгоритма [5], при условии, что $q > (g-1)!$ для первого случая и $q > (g-1)!/g$ для второго и являются асимптотически

быстрее общего алгоритма для $g \geq 3$. Следовательно ГЭК рода 3 не являются стойкими, как считалось ранее.

Спуск Вейля

В работе [23] показана возможность применение спуска Вейля [37] для атаки на ЗДЛГЭК небольшого рода над $F_{2^{mn}}$ с составной степенью расширения. Впервые в [37] сводят ЗДЛЭК над полем F_{q^n} к ЗДЛГЭК в якобиане ГЭК большего рода над некоторым подполем F_q поля F_{q^n}. Приведем условия, применения атаки:

1. Числа l и n положительные целые.
2. Причем $q = 2^l$, $k = F_q$ и $K = F_{q^n}$.
3. Не суперсингулярная ЭК E над K задана уравнением $E: y^2 + xy = x^3 + ax^2 + b$, $a \in K$, $b \in K^*$.
4. Порядок ЭК $\#E(K) = hr$, где h небольшое целое 2 или 4, r-простое. Следовательно $r \approx q^n$. Пусть $b_i = b^{q_i}$ и $m(b) = \dim_{\mathbf{F}_2}\left(\mathrm{Span}_{\mathbf{F}_2}\left\{\left(1, b_0^{1/2}\right), \left(1, b_1^{1/2}\right), \ldots, \left(1, b_{n-1}^{1/2}\right)\right\}\right)$. Положим, что или n является нечетным или $m(b) = 0$ или $Tr_{K/\mathbf{F}_2}(a) = 0$.
5. Построение ограничения Вейля $W_{E/k}$ скалярных величин $E(K)$, которое является n-размерным абелевым многообразием над k.
6. В результате пересечения $W_{E/k}$ с $n-1$ гиперплоскостью, получается ГЭК C, рода $g = 2^{m-1}$ или $g = 2^{m-1} - 1$, где $m = m(b)$.

Обычно $m \approx n$, т.е. $g \approx 2^{n-1}$. Известно, что $\#J_C(k) \approx q^g \approx q^{2^{n-1}}$. Далее решение ЗДЛГЭК возможно одним из описанных выше методов ИИ.

В [19, 43, 44] показано применение спуска Вейля для сведения ЗДЛЭК в подгруппе точек $E(K)$ порядка r к подгруппе $J_C(k)$ порядка r ГЭК $C(k)$ рода g. Атака спуском Вейля практически не применима к кривым заданным над простым полем и двоичным расширенным, когда степень расширения больше 40 и является простым числом [23].

Применение спуска Вейля к большему классу ЭК, заданных над полем характеристики отличной от 2, благодаря эффективному поиску кривых изогенных для заданной (взломанной) кривой, рассмотрено в [45-47].

Сравнение

Результирующие оценки сложности различных алгоритмов ЗДЛГЭК обобщена в табл.2.

Таблица 2

№	Название алгоритма	Условие применения	Сложность
1	Adleman, DeMarrais, Huang	$C(F_q)$, q - нечетное простое, g - большое	$O\left(L_{q^{2g+1}}[c]\right)$, $c = 2.313$
2	Muller, Stain, Theil	$C(F_q)$, q - нечетное простое, g - большое	$O\left(L_{q^{2g+2}}[1.44]\right)$
3	Flassenberg, Paulus	$C(F_q)$, q - нечетное простое, g - большое	?
4	Smart	$C(F_q)$, q - нечетное простое, g - большое	?
5	ρ-Pollard	$J_C(F_q)$ содержит m классов	$O\left(q^{(2g-1)/4}/\sqrt{m}\right)$

		смежности	
6	Больших и малых шагов	$C(F_q)$	$O\left(q^{g(g+1)/8}/m^{g/2}\right)$, $m=\Omega\left(q^{\varepsilon+(g-2)/4}\right)$, $\forall\varepsilon>0$
7	Bauer & Enge	$C(F_q)$	$O\left(\exp\left(\rho\sqrt{(g\log q)\log(g\log q)}\right)\right)$, $\rho=\frac{5}{2\sqrt{3}}\left(\sqrt{1+\frac{3}{\upsilon}}+\sqrt{\frac{3}{\upsilon}}\right)+o(1)$, $\upsilon>0$, $o(1)\to 0$ при $g\log q\to\infty$
8	Gaudry	$C(F_q)$, известен $\#J_C(F_q)$, g - небольшое	$O\left(g^3q^2\log^2 q+g^2g!q\log^2 q\right)$
9	Duursma, Gaudry, Morain	$C(F_q)$, известен порядок автоморфизма Фробениуса	$O\left(q^{\frac{2g}{g+1}+\varepsilon}\right)$ при $q\to\infty$
10	Gaudry, Hess, Smart	ГЭК рода 4 над $\mathbf{F}_{2^{21}}$	$O\left(N\cdot L_{e^g}\left[\frac{1}{2}, 3.54\sqrt{\log q}\right]\right)$, при $g\to\infty$, $N=\deg_y C(x,y)$
11	Enge, Gaudry	g - большое	$O\left(L_{q^g}[c]\right)$, $c=\sqrt{2}\left(\sqrt{1+\frac{1}{2g}}+\frac{1}{\sqrt{2g}}\right)$
12	Jacobson, Menezes, Stein	$g=31$, $\mathbf{F}_{2^m}$	$O\left(\exp\left((1+o(1))\sqrt{(2g+1)\log q\log((2g+1)\log q)}\right)\right)$
13	Theriault	$C(F_q)$, g - небольшое	$O\left(g^5q^{2-2/(g+1)+\varepsilon}\right)$ при $q>(g-1)!$ и $O\left(g^5q^{2-4/(2g+1)+\varepsilon}\right)$ при $q>(g-1)!/g$

Из данных приведенных в табл. 2 следует, что алгоритмы 1-4, 11 и12 эффективно решают ЗДЛГЭК большого рода, следовательно, в криптографических целях имеет смысл применять лишь кривые небольшого рода. В то же время, алгоритмы 8, 10, 13 являются эффективными для ГЭК рода более 4. Причем алгоритм 13 уменьшает границу до 3, в этом случае он имеет сложность меньшую, чем сложности универсальных алгоритмов.

Как известно [18], что для противостояния универсальным алгоритмам криптоанализа (полного перебора, Полларда, Похлинга-Хелмана, больших и малых шагов) предлагается использовать ГЭК $\#J_C(F_q)=q^g\geq 2^{160}$. Исследования [3, 25] показывают, что кривые с эффективно вычислимым эндоморфизмом Фробениуса являются уязвимыми. Наличие эндоморфизма позволят повысить эффективность универсальных алгоритмов, а также позволяет применить специализированные алгоритмы, например алгоритм 9 в табл. 2.

Выводы

В работе рассмотрены наиболее эффективные алгоритмы решения ЗДЛГЭК, из известных на сегодняшний день. Полученные результаты подтверждают факт, что дальнейшее развитие ассиметричной криптографии лежит в увеличении сложности кривой (рода), т.е. перехода к ГЭК. Повышая род, повышается и сложность решения ЗДЛ, причем возможно уменьшение размера базового поля без ущерба стойкости. Другими словами, ГЭК позволяют обеспечить аналогичную сложность ЗДЛ при соизмеримых параметрах с ЭК, в тоже время дальнейшее повышение сложности кривой (род >3) – не целесообразно, т.к. возможно эффективное применение алгоритма 13 (табл. 2). Уменьшенный размер базового поля открывает «новые горизонты» для ГЭК, которыми являются мобильные устройства.

Анализ алгоритмов позволил сформулировать подходы к реализации скрытых резервов эффективности посредством:

- уменьшения размера базы разложения;
- прореживания (отсеивание) соотношений на этапе их генерации (FP метод, Smart метод);
- решения, как разряженных, так и плотных СЛУ над полем;
- распараллеливания (например генерации соотношений, решения СЛУ);
- использования повторного логарифмирования [53, 54];

- гармонизации подходов к ЗДЛГЭК, используемых для решения ЗДЛ над полем и задачи факторизации.

Остается открытым вопрос о существовании алгоритма криптоанализа преобразований на ГЭК со сложностью соизмеримой с $O\left(\exp\left((c+o(1))m^{1/3}(\log m)^{2/3}\right)\right)$ для алгоритма криптоанализа Адлеманна-Копперсмита в поле [49, 50]. Дальнейшего исследования требует классификация кривых уязвимых к атаке спуском Вейля.

Список литературы. 1. *U. Vollmer*. Asymptotically fast discrete logarithms in quadratic number fields. Algorithmic number theory-ANTS-IV, LNCS 1838, 200, pp.581-594. 2. *A. Stein, E. Teske*. Explicit bounds and heuristics on class numbers in hyperelliptic functions fields. Mathematics of computation, 71 (2002), pp.837-861. 3. *R. Gallant, R. Lambert, S. Vanstone*. Improved the parallelized Pollard lambda search on anomalous binary curves. Mathematics of computation, 69 (2000), pp.1699-1705. 4. *M. Wiener, R. Zuccherato*. Faster attacks on elliptic curve cryptosystems. Selected areas in cryptography, LNCS 1556, 1999, pp.190-200. 5. *P. Gaudry*. An algorithm for solving the discrete log problem on hyperelliptic curves. Advances un cryptology-EUROCTYPT'2000, LNCS 1807, 2000, pp.19-34. 6. *M. Bauer*. A subexponential algorithm for solving the discrete logarithm problem in the Jacobian of high genius hyperelliptic curves over arbitrary finite fields. Preprint, 1999. 7. *L. Adleman, J. DeMarrais, M. Huang*. A subexponential discrete logarithms over the rational subgroup of the jacobians of large genius hyperelliptic curves over finite fields. Algorithmic number theory-ANTS-II, LNCS 877, 1994, pp.28-40. 8. *V. Muller, A. Stain, C. Thiel*. Computing discrete logarithms in real quadratic congruence function fields of large genius. Mathematics of computation, 68 (1999), pp.807-822. 9. *A. Stein*. Equivalences between elliptic curves and real quadratic congruence function fields. Journal de Theorie des numbers de Bordeaux, 9 (1997), pp.75-95. 10. *R. Zuccherato*. The equivalence between elliptic curve and quadratic function field discrete logarithms in characteristic 2. Algorithmic number theory-ANTS-III, LNCS 1423, 1998, pp.621-638. 11. *S. Paulus, H.-G. Ruck*. Real and imaginary quadratic representations of hyperelliptic function fields. Mathematics of computation, 68 (1999), pp.1233-1241. 12. *N. Smart*. Experiments using an analogy of the number field sieve algorithm to solve the discrete logarithm problem in the Jacobians of hyperelliptic curves. HP Laboratories Bristol, Technical report HPL-97-130, 1997. 13. *J. Hafner, K. McCurley*. A rigorous subexponential algorithm for computation of class groups. Journal of the American Mathematical Society, 2 (1989), pp.837-850. 14. *R. Flassenberg, S. Paulus*. Sieving in function fields. Experimental mathematics, 8 (1999), pp.229-249. Available at: ftp://ftp.informatik.tu-darmstadt.de/pub/TI/TR/TI-97-13.rafla.ps.gz. 15. *A. Enge*. Computing discrete logarithms in high-genius hyperelliptic jacobians in provably subexponential time. Mathematics of computation, 71 (2001), pp.729-742. 16. *A. Enge*. A general framework for subexponential discrete logarithm algorithms in groups of unknown order. Finite geometries, Developments in mathematics vil. 3, Kluwer Academic publishers, Dordrecht 2001, pp.133-146. 17. *A. Enge, A. Stein*. Smooth ideals in hyperelliptic function fields. Mathematics of computation, 71 (2001), pp.1219-1230. 18. *E. Teske*. Speeding up Pollard's rho method for computing discrete logarithms. Algorithmic number theory, LNCS 1423, 1998, pp.541-554. 19. *P. Gaudry, F. Hess, N. Smart*. Constructive and destructive facets of Weil descent on elliptic curves. Journal of cryptology, 15 (2002), pp.19-46. 20. *A. Enge, P. Gaudry*. A general framework for subexponential discrete logarithm algorithms. Acta arithmetica, 102 (2002), pp.83-103. 21. *M. Jacobson, A. Menezes, A. Stein*. Solving elliptic curve discrete logarithm problems using Weil descent. Journal of the Ramanujan mathematical society, 16 (2001), pp.231-260. 22. *N. Theriault*. Index calculus attack for hyperelliptic curves of small genius. Preprint, 2003. 23. *S. Galbraith*. Weil descent of jacobians. Discrete applied mathematics. 128 (2003), pp.165-180. 24. *V. Shoup*. Lower bounds for discrete logarithms and related problems. Advances in cryptology – EUROCRYPT'97, LNCS 1233, pp.256-266. 25. *R. Lambert*. Computation aspects of discrete logarithms. A thesis for the degree of DP in electrical engineering. Waterloo, Ontario, Canada, 1996. 26. *J. Buchmann, D. Weber*. Discrete logarithms: recent progress. 1998. 27. *M. Jacobson Jr, A. Menezes, A. Stein*. Hyperelliptic curves and cryptography. 28. *Тевяшев А.Д.* Алгебра и геометрия. Алгебраические структуры. Категории. – Харків: ХТУРЕ, 1998. - 388с. 29. *H. Shizuya, T. Itoh, K. Sakurai*. On the complexity of hyperelliptic discrete logarithm problem. 30. *K. Matsuo, J. Chao, S. Tsuijii*. Baby step gaint step algorithms in point counting of hyperelliptic curves. Special section on discrete mathematics and its applications of IEICE. Vol. E86-A, No. 5, May 2003. pp.1127-1134. Japan. 31. *A.M. Odlyzko*. Discrete logarithms in finite fields and their cryptographic significance. Advances in cryptology – EUROCRYPT'84 (T. Beth, N. Cot, I. Ingemarsson, eds.), LNCS 209, Springer-Verlag, 1984, pp.224-314. 32. *D.H. Wiedemann*. Solving sparse linear equations over finite fields. IEEE Trans. Info. Theory IT-32 (1986), No. 1, pp.54-62. 33. *C. Pomerance, J.W. Smith*. Reduction of huge, sparse matrices over finite fields via created catastrophes. Experimental mathematics 1 (1992), pp.89-94. 34. *B.A. LaMacchia, A.M. Odlyzko*. Solving large sparse linear systems over finite fields. Advances in cryptology-CRYPTO'90. (S. Vanstone ed.). LNCS 537, Springer-Verlag, 1990, pp.109-133. 35. *R. Gallant, R. Lambert, S. Vanstone*. Improving the parallelized Pollard lambda search on binary anomalous curves. Research Report CORR 98-15, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1998. 36. *L. Wagner*. Algebro-geometric attack methods in elliptic curve cryptography. Honours thesis. November 18, 2002. 37. *G. Frey*. Applications of arithmetical geometry to cryptographic constructions. Proceedings of the Fifth International Conference on Finite Fields and Applications. Springer-Verlag, 2001, pp.128-161. 38. *M. Jacobson, A. Menezes, A. Stein*. Solving elliptic curve discrete logarithm problems using Weil descent. May 16,

2001.39. *J.M. Pollard*. The lattice sieve. In A.K. Lenstra, H.W. Lenstra, ed. The development of the number field sieve. Springer-Verlag, LNM 1554, 1993. at pp.43-49. 40. *A.J. Menezes, Y. Wu, R.J. Zuccherrato*. An elementary introduction to hyperelliptic curves. Technical report CORR96-19, Department of combinatorics and optimization, University of Waterloo, Waterloo, Ontario, 1996. In: Koblitz, N.: Algebraic aspects of cryptography, Springer-Verlag, Berlin Heidelberg New York. 1998. 41. *I. Duursma, P. Gaudry, F. Morain*. Speeding up the discrete log computation on curves with automorphisms. 42. *G. Frey, H.-G. Ruck*. A remark concerning m -divisibility and discrete logarithm in the divisor class group of curves. Math. Comp., 62 (206), pp.865-874, April 1994. 43. *A. Menezes, M. Qu*. Analysis of the Weil descent attack of Gaudry, Hess and Smart. Research Report CORR 00-48, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2000. 10 pages. Revised April 22, 2001. 44. *M. Maurer, A. Menezes, E. Teske*. Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. Research Report CORR 01-59, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2001. 42 pages. 45. *S.D. Galbraith, F. Hess, N.P. Smart*. Extending the GHS Weil descent attack. Available: http://erpint.iacr.org 46. *E. Teske*. An elliptic curve trapdoor system. Research Report CORR 03-07, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2003. 47. *A. Menezes, E. Teske, A. Weng*. Weak fields for ECC. Research Report CORR 03-15, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2003. 48. *T. Okamoto, K. Sakurai.* Efficient algorithms for the construction of hyperelliptic cryptosystems. 49. *E. Thome*. Computation of Discrete Logarithms in $F_{2^{607}}$. Laboratorie d'Informatique (LIX) Ecole polytechnique, Essen, France, 2001. 50. *E. Thome*, Computing discrete logs in large characteristic 2 finite fields. Laboratorie d'Informatique (LIX) Ecole polytechnique, Essen, France, 25 September 2002. 51. *S.D. Galbraith, N. P. Smart*. A cryptographic application of Weil descent. Cryptography and coding, 7th IMA Conference, Springer-Verlag, LNCS 1746, pp. 191-200, 1999. The full version of the paper is HP Labs Technical report, HPL-1999-70. 52. *A. Miyaji, M. Nakabayashi, S. Takano*. New explicit conditions of elliptic curve traces for FR-reduction. 53. *D. Johnson, A. Menezes, S.Vanstone*. The Elliptic curve digital signature algorithm. Certicom research. Canada. 54. *J. M. Pollard*. Kangaroos, Monopoly and discrete logarithms. Journal of Cryptology, 13: 437-447, 2000.