А. С. Платонов, А. Н. Гамова, Эффективные алгоритмы скалярного умножения в группе точек эллиптической кривой,
Чебышевский сб., 2013, том 14, выпуск 4, 180–187

https://www.mathnet.ru/cheb316

ЧЕБЫШЕВСКИЙ СБОРНИК
Том 14 Выпуск 4 (2013)

УДК003.26.51

ЭФФЕКТИВНЫЕ АЛГОРИТМЫ СКАЛЯРНОГО УМНОЖЕНИЯ В ГРУППЕ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

А. С. Платонов, А. Н. Гамова (г. Саратов)

Аннотация

В статье описываются несколько алгоритмов скалярного умножения на эллиптических кривых и производится сравнение производительности их реализаций.

Ключевые слова: эллиптические кривые.

EFFICIENT ALGORITHMS FOR SCALAR MULTIPLICATION OVER ELLIPTIC CURVES

A. S. Platonov, A. N. Gamova (c. Saratov)

Abstract

The paper describes several algorithms for scalar multiplication over elliptic curves and compares the performance of their implementations.

Key words: elliptic curves.

1. Скалярное умножение на кривых Коблица

Рассмотрим модификацию общего алгоритма скалярного умножения для частного случая эллиптических кривых над полем характеристики 2. Аномальными бинарными кривыми называются несуперсингулярные кривые E_0 и E_1, заданные уравнениями:

$$E_a : Y^2 + XY = X^3 + aX^2 + 1$$

Чаще всего их называют кривыми Коблица. Рассмотрим несколько основных свойств этих кривых. Порядок кривой $E_a(F_{2^m})$ всегда будет кратен либо 2, либо 4. При этом наибольший интерес с криптографической точки зрения представляют кривые, порядок которых имеет вид fp , где p - простое число и f равно 2 или 4.

Если предположить, что порядок кривой $E_a(F_{2^m})$ равен fp, то главной подгруппой этой кривой называется подгруппа порядка p (число f называют кофактором). При этом точка $P = (x, y)$ принадлежит главной подгруппе тогда и только тогда, когда $tr(x) = 1$ (при $a = 1$) или $tr(x) = 0$ и $tr(y) = tr(\lambda x)$, где $\lambda^2 + \lambda = x$ (при $a = 0$). Алгоритм, описанный в этой статье, работает только с точками главной подгруппы.

Одно из свойств кривых $E_a(F_{2^m})$ состоит в том, что если $(x, y) \in E_a(F_{2^m})$, то и $(x^2, y^2) \in E_a(F_{2^m})$. Также, из определения операции сложения точек эллиптической кривой следует следующее равенство:

$$(x^4, y^4) + 2(x, y) = \mu\ (x^2, y^2)$$

для любых (x, y) на $E_a(F_{2^m})$, где

$$\mu\ = (-1)^{(1-a)}.$$

Отображение $\tau : (x, y) \to (x^2, y^2)$ называется эндоморфизмом Фробениуса. Справедливо равенство:

$$(\tau^2 + 2)P = \mu\tau P$$

Рассмотрим подкольцо кольца эндоморфизмов кривой $E_a(F_{2^m})$, порожденное эндоморфизмом Фробениуса. Обозначим его $Z[\tau]$. Представление его элементов в виде многочленов от τ с целыми коэффициентами неоднозначно. Например, согласно приведенному выше равенству, эндоморфизм $\tau^2 - \mu\tau + 2$ нулевой. Рассмотрим приведенное выше уравнение в поле комплексных чисел и обозначим также через τ его корень

$$\frac{\mu + \sqrt{-7}}{2}$$

Сопоставим каждому эндоморфизму из $Z[\tau]$ комплексное число. Говорят, что кривые E_a обладают комплексным умножением на τ. Это позволяет нам заменять операции умножения на числа из кольца $Z[\tau]$ операцией τ, то есть простым возведением в квадрат обеих координат точки. Для того, чтобы использовать этот факт, рассмотрим алгоритм, преобразующий множитель n.

При вычислении скалярного произведения nP, число n можно представить, как элемент кольца $Z[\tau]$. Такое представление n в виде суммы степеней τ называют τ-адической несовместной формой (τ-adic non-adjacent form – $TNAF$). Использование $TNAF$ натуральных чисел позволяет умножать скалярно на эти числа точки кривой E_a, не используя операцию удвоения точек, а используя только сложение и возведение в квадрат.

Algorithm 1: Вычисление TNAF числа n.

Data: натуральные числа r_0, r_1
Result: $TNAF(r_0 + r_1\tau)$

```
1  c_0 ← r_0; c_1 ← r_1;
2  S ←<>
3  while c_0 ≠ 0 or c_1 ≠ 0 do
4  |  if c_0 mod 2 = 1 then
5  |  |  u ← 2 − (c_0 − 2c_1 mod 4)
   |  |  c_0 ← c_0 − u
6  |  else
7  |  |  u ← 0
8  |  end
9  |  Добавить u в начало S
10 |  (c_0, c_1) ← (c_1 + μc_0/2, −c_0/2)
11 end
12 TNAF ← S
```

Например, $TNAF$ числа $9 = \tau^5 - \tau^3 + 1$. Значит $9P = (x^{32}, y^{32}) - (x^8, y^8) + (x, y)$ и, пренебрегая возведением в квадрат, для вычисления $9P$ потребуется выполнить 2 операции обращения и 4 операции умножения в поле F_{2^m} . При использовании стандартного метода скалярного умножения потребовалось бы 4 операции обращения и 8 операций умножения. Алгоритм 1 описывает способ вычисления $TNAF$ числа $n = r_0 + r_1\tau$.

Для дальнейшего повышения эффективности вычислений можно произвести модулярную редукцию, которая уменьшит длину $TNAF$ в два раза. Для проведения модулярной редукции требуется использование операций округления и деления в кольце $Z[\tau]$. Описание алгоритмов данных операций в этой статье не приводится, ознакомиться с ними можно в [1].

Для кривой $E_a(F_{2^m})$ порядка fp введем элемент $\delta = (\tau^m - 1)/(\tau - 1)$. Тогда для любых двух элементов ρ и γ из $Z[\tau]$, для которых справедливо $\gamma \equiv \rho \bmod \delta$, выполняется $\gamma P = \rho P$. Значит $TNAF(\gamma)$ и $TNAF(\rho)$ эквивалентны в главной подгруппе. Определим редуцированную $TNAF(RTNAF)$ элемента n:

$$RTNAF(n) = TNAF(\rho), \text{ где } \rho = n \bmod \delta.$$

Таким образом, для вычисления RTNAF нужно определить алгоритм редукции числа n по модулю δ. Введем вспомогательные переменные

$$s_i = \frac{(-1)^i}{f}(1 - \mu U_{m+3-a-i})$$

где U_k - последовательность Люка. Ниже приведен алгоритм вычисления $n \bmod \delta$.

Algorithm 2: Вычисление $n \bmod \delta$

Data: натуральное число n, параметры s_0, s_1, a, m, p.
Result: числа r_0, r_1 такие, что $r_0 + r_1\tau = n \bmod \delta$

$d_0 \leftarrow s_0 + \mu s_1$
$\lambda_0 \leftarrow s_0 n/p$
$\lambda_1 \leftarrow s_1 n/p$
$(q_0, q_1) \leftarrow$ Округление(λ_0, λ_1)
$r_0 \leftarrow n - d_0 q_0 - 2 s_1 q_1$
$r_1 \leftarrow s_1 q_0 - s_0 q_1$

Используя изложенные выше сведения, можно построить эффективный алгоритм скалярного умножения nP в главной подгруппе кривой $E_a(F_{2^m})$.

Algorithm 3: Вычисление nP с использованием $RTNAF$.

Data: точка эллиптической кривой P, натуральное число n, параметры s_0, s_1, a, m, p.
Result: $Q = nP$

$(r_0, r_1) \leftarrow n \bmod \delta$
$Q \leftarrow \mathbf{O}$, $P_0 \leftarrow P$
while $r_0 \neq 0$ *or* $r_1 \neq 0$ **do**
  if $r_0 \ mod \ 2 = 1$ **then**
    $u \leftarrow 2 - (r_0 - 2r_1 \ mod \ 4)$
    $r_0 \leftarrow r_0 - u$
    if $u = 1$ **then**
      $Q \leftarrow Q + P_0$
    end
    if $u = -1$ **then**
      $Q \leftarrow Q - P_0$
    end
  end
  $P_0 \leftarrow \tau P_0$
  $(r_0, r_1) \leftarrow (r_1 + \mu r_0/2, -r_0/2)$
end

2. Проективные координаты

Проведем сравнение производительности алгоритма 3 с методом, использующим смешанные проективные координаты [2]. Ниже изложено краткое описание данного метода.

При сложении двух точек эллиптической кривой над полем характеристики, большей 3 (т.е. для кривой, имеющих вид $y^2 = x^3 + ax + b$) требуется произ-

вести два умножения, одно возведение в квадрат и одно обращение. Переход к другой системе координат позволяет полностью исключить операцию обращения за счет увеличения числа других операций. Поэтому если для данного поля операция обращения занимает значительно больше времени, чем операция умножения, то использование другой системы координат может значительно ускорить вычисления.

Рассмотрим эллиптическую кривую над полем K, и положим c и d положительными числами. Определим отношение эквивалентности на ненулевых тройках из K^3 следующим образом:

$$(X_1, Y_1, Z_1) \sim (X_2, Y_2, Z_2), X_1 = \lambda^c X_2, Y_1 = \lambda^d Y_2, Z_1 = \lambda Z_2, \lambda \in K, \lambda \neq 0.$$

Класс эквивалентности, содержащий тройку $(X, Y, Z) \in K^3 \setminus \{(0, 0, 0)\}$, обозначим следующим образом:

$$(X : Y : Z) = (\lambda^c X, \lambda^d Y, \lambda Z) : \lambda \in K.$$

Класс эквивалентности $(X : Y : Z)$ называется проективной точкой, а (X, Y, Z) - ее представителем. Множество всех проективных точек обозначается $P(K)$. Следует отметить, что если $(X', Y', Z') \in (X : Y : Z)$, то $(X' : Y' : Z') = (X : Y : Z)$. Таким образом, каждый элемент класса эквивалентности может служить его представителем. В частности, если положить $Z \neq 0$, то $(X/Z^c, Y/Z^d, 1)$ является представителем проективной точки $(X : Y : Z)$, причем единственным представителем с координатой $Z = 1$. Таким образом, мы получаем однозначное соответствие между набором проективных точек

$$P(K) = (X : Y : Z) : X, Y, Z \in K, Z \neq 0$$

и набором аффинных точек

$$A(K) = (x, y) : x, y \in K.$$

Набор проективных точек

$$P(K)^0 = (X : Y : Z) : X, Y, Z \in K, Z = 0$$

называется прямой в бесконечности, так как точки из этого набора не соответствуют никаким аффинным точкам.

Проективная форма уравнения эллиптической кривой E над полем K получается путем подстановки $x = X/Z^c, y = Y/Z^d$ и избавления от знаменателей (то есть путем домножения на Z в некоторой степени). Если некоторый представитель класса $(X, Y, Z) \in K^3 \setminus \{(0, 0, 0)\}$ удовлетворяет полученному проективному уравнению, то ему удовлетворяет любой представитель класса $(X : Y : Z)$. Таким образом можно сказать, что проективная точка $(X : Y : Z)$ лежит на

кривой E. Проективные точки из $P(K)^0$, лежащие на кривой E, будут являться бесконечно удаленными точками. Использование подобных замен позволяет избавиться от операции обращения при сложении двух точек за счет увеличения числа других операций.

Наиболее часто используемые системы координат для кривых над полями больших характеристик ($y^2 = x^3 + ax + b$) – стандартная проективная система координат (P), система координат Якоби (J), система координат Чудновского-Якоби (Jc), модифицированная система координат Якоби (Jm).

Некоторые системы координат позволяют быстро производить удвоение точки, другие - сложение двух точек. Однако при реализации алгоритмов на эллиптических кривых вовсе не обязательно использовать только одну систему координат. Если нам требуется складывать различные точки, можно использовать одну систему координат, а если нужно производить удвоения точки - другую. Алгоритм 4 использует эту идею.

Algorithm 4: Вычисление nP с использованием смешанных координат.

Data: точка $P = (X1 : Y1 : Z1)$ в системе координат Якоби, лежащая на кривой $y^2 = x^3 + ax + b$ и целое число $n > 0$
Result: $nP = (Xn : Yn : Zn)$ в системе координат Якоби

```
1  P* ← (1, 1, 0); T ← (X1 : Y1 : Z1 : aZ1^4)
2  while n > 1 do
3  |  if n mod 2 = 1 then
4  |  |  u ← 2 − n mod 4
5  |  |  n ← n − u
6  |  |  if u = 1 then
7  |  |  |  P* ← P* + T
8  |  |  end
9  |  |  if u = −1 then
10 |  |  |  P* ← P* − T
11 |  |  end
12 |  end
13 |  n ← n/2
14 |  T ← 2T
15 end
```

3. Сравнение производительности алгоритмов скалярного умножения

В таблице 1 показаны результаты сравнения трех алгоритмов умножения точки на скаляр: стандартный алгоритм для кривой $E_a(F_{2^m})$ с использованием

NAF формы, алгоритм 4 с использованием смешанных координат для кривой $E_a(F_P)$ и алгоритм 3, использующий $RTNAF$ для кривой Коблица. Для кривых характеристики 2 использовались кривые Коблица, рекомендованные стандартом $NIST$. Для алгоритма 4 использовались случайно сгенерированные кривые аналогичной характеристики.

Характеристика поля, биты	163	233	283	409	571
t(Standart NAF), мс	6.96	13.89	20.21	45.66	101.57
t(Algorithm 4), мс	2.08	3.39	3.92	8.32	18.46
t(Algorithm 3), мс	2.06	3.71	6.02	13.14	28.43

Таблица 1: Среднее время проведения операции скалярного умножения

Использование смешанных координат дает больший прирост производительности, особенно при увеличении характеристики P. Однако алгоритм $RTNAF$ может также использовать проективные координаты для сложения двух точек. Для этого нужно модифицировать метод вычисления τP - эта операция должна возводить в квадрат все координаты точки. Идея алгоритма, использующего одновременно $RTNAF$ и проективные координаты, состоит в том, что для сложения двух точек мы используем проективные координаты, а для удвоения точки - операцию τ. Однако при реализации этого алгоритма стоит учитывать, что использование проективных координат для сложения двух точек даст прирост производительности лишь в том случае, если в поле F_{2^m} операция обращения выполняется хотя бы в 13 раз медленнее (в проективной системе Лопеса–Дахаба), чем операция умножения.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Solinas J. A. Efficient Arithmetic on Koblitz Curves // Journal Designs, Codes and Cryptography. 2000. Vol. 19, № 2-3. P. 195–249.

2. Платонов А. С., Гамова А. Н. Проективные координаты в криптографии эллиптических кривых // Компьютерные науки и информационные технологии: материалы Международной научной конференции. Саратов, 2012. С. 256–258.

REFERENCES

1. Solinas J. A. Efficient Arithmetic on Koblitz Curves // Journal Designs, Codes and Cryptography. 2000. Vol. 19, № 2-3. pp. 195–249.

2. Platonov A. S., Gamova A. N. Projective coordinates for elliptic curve cryptography // Computer Science and Information Technology: Proceedings of the International Scientific Conference. Saratov, 2012. pp. 256–258 (in Russian).

Саратовский государственный университет им. Н. Г. Чернышевского.
Поступило 14.09.2013