ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

УДК 512.742.72

БЫСТРЫЕ АЛГОРИТМЫ ВЫЧИСЛЕНИЯ ПРЕОБРАЗОВАНИЙ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ С ПРЕДВАРИТЕЛЬНЫМИ ВЫЧИСЛЕНИЯМИ

Д.С. Хлебородов

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация
e-mail: dkhleborodov@gmail.com

Эффективные алгоритмы вычисления преобразований на основе эллиптических кривых — сравнительно новая область исследований, представляющая огромный интерес. Рассмотрено эффективное скалярное умножение точки эллиптической кривой, определенной над простым полем, с использованием предварительных вычислений. В основе предлагаемых алгоритмов лежат методы несовместного представления скаляра (Non-Adjacent Form, NAF) с окном фиксированной и переменной ширины (скользящим окном). Для оценки эффективности полученных и ранее предложенных алгоритмов введен критерий эффективности, основанный на средней вычислительной сложности. Для получения новых более эффективных алгоритмов использованы эффективные операции в простом поле, операции с точкой: сложение (ADD); удвоение (DBL и DBL_A^J); операция "удвоить и сложить" (DA); масштабирование (SCALE); свойства аффинной координатной системы; свойства координатной системы Якоби; свойства координатной системы Якоби – Чудновского. Для алгоритмов сформулированы и доказаны утверждения относительно их вычислительной сложности.

***Ключевые слова*:** быстрые алгоритмы, эллиптические кривые, сложность, предварительные вычисления, скалярное умножение точки.

FAST COMPUTATION ALGORITHMS OF TRANSFORMATIONS BASED ON ELLIPTIC CURVES WITH PRECOMPUTATIONS

D.S. Khleborodov

Bauman Moscow State Technical University, Moscow, Russian Federation
e-mail: dkhleborodov@gmail.com

The paper considers efficient algorithms for calculating transformations based on elliptic curves. This new research area is of great interest nowadays. The paper presents an effective scalar multiplication of an elliptic curve point defined over the prime field using of preliminary calculations. The proposed algorithms are based on the method of incompatible representations of the scalar (Non-Adjacent Form, NAF) with a window of both fixed and variable width (a sliding window). In order to evaluate the effectiveness of the previously obtained algorithms we proposed the architecture of the algorithms and the criterion of efficiency based on the average computational complexity. For elaborating newer and more efficient algorithms some effective operations both in the prime field and with a point, such as addition (ADD), doubling (DBL, DBL_A^J), a complex operation "double and add" (DA), a scaling duplication (SCALE), properties of the affine coordinate system, and Jacobi and Jacobi – Chudnovsky coordinate systems are used. Statements about the algorithms computational complexity are formulated and proved.

***Keywords*:** fast algorithms, elliptic curves, precomputations, computational complexity, scalar multiplication of point.

Введение. Преобразования на основе эллиптических кривых широко используются в криптосистемах с открытым ключом, в частности при вычислении цифровых подписей в отечественном (ГОСТ Р 34.10–2012) и зарубежных (FIPS PUB 186, ANSI X9.62, SECG) стандартах. Такие криптосистемы, построенные на эллиптических кривых, обеспечивают снижение вычислительных затрат, энергопотребления и памяти при обеспечении высокого уровня безопасности. Преобразования на основе эллиптических кривых также находят свое применение в алгоритмах факторизации целых чисел (метод Ленстры — ECM, Elliptic Curve Factorization Method) и в алгоритмах тестирования натуральных чисел на простоту. Такие алгоритмы представляют практический и теоретический интерес. Несмотря на значительные успехи многих исследователей в данной области, интерес к получению новых результатов по ускорению преобразований сохраняется вследствие большой актуальности приведенных выше задач.

Критерий эффективности. Рассмотрим поле $\mathbb{F}_p$, такое, что p — простое число и для $a, b \in \mathbb{F}_p$ выполняется неравенство $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$. Тогда эллиптическая кривая $\mathrm{E}(\mathbb{F}_p)$ над $\mathbb{F}_\mathrm{p}$, определенная параметрами $a, b \in \mathbb{F}_p$, состоит из набора решений или точек $P = (x, y)$, $x, y \in \mathbb{F}_p$, уравнения $E(\mathbb{F}_p) : y^2 = x^3 + ax^2 + b \pmod{p}$, вместе с особой точкой $\mathcal{O}$, называемой точкой в бесконечности.

Уравнение $y^2 = x^3 + ax^2 + b \pmod{p}$ задано в форме Вейерштрасса и называется определяющим уравнением $E(\mathbb{F}_p)$. Для точки $P = (x_P, y_P)$ x_P является *x*-координатой P, а y_P — *y*-координатой точки P. Такого рода интерпретация точки называется аффинными координатами.

Набор точек на $E(\mathbb{F}_p)$ представляет собой абелевую *группу точек эллиптической кривой* с операцией сложения $(E(\mathbb{F}_p), \boxplus)$ [1].

Главная операция преобразования на основе эллиптических кривых — *скалярное умножение точки*: дано $k \in \mathbb{Z}$ и точка $P \in E(\mathbb{F}_p)$, скалярное умножение — процесс сложения точки P с собой k раз: $dP = \underbrace{P \boxplus \ldots \boxplus P}_{k}$. Результат скалярного умножения обозначается kP.

Операцией *мультискалярного умножения* называется преобразование вида: $\boxplus_{i=1}^{n} k_i Q_i$, $k_i \in \mathbb{Z}$, $Q_i \in E(\mathbb{F}_p)$ [2].

Для построения эффективных алгоритмов на основе эллиптических кривых и оценки их вычислительной сложности воспользуемся иерархическим подходом. Рассмотрим математическую архитектуру алгоритмов (рис. 1) на основе эллиптических кривых, состоящую из пяти уровней. Каждый уровень представляет собой множество алгоритмов, являющихся основой для получения алгоритмов вышестоящего уровня методом композиции.

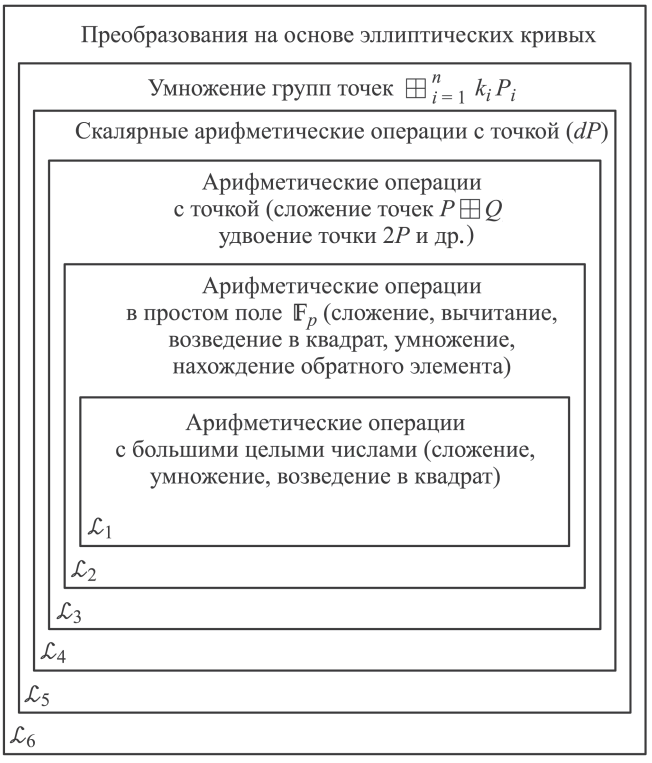

Рис. 1. Архитектура преобразований на основе эллиптических кривых

Пусть $\mathbb{E}$ — множество рассматриваемых кривых $E(\mathbb{F}_p)$, $\mathcal{L}_i$ — множество алгоритмов вычисления преобразований на основе эллиптических кривых уровня i математической иерархии. При $i = 1$ множество алгоритмов $\mathcal{L}_1$ будем называть множеством *базовых алгоритмов*. Тогда алгоритм $a \in \mathcal{L}_i$, $i > 1$, получается композицией из t алгоритмов множества $\mathcal{L}_{i-1} : \mathcal{F}_a : \mathcal{L}_{i-1}^t \longrightarrow \mathcal{L}_i$.

Архитектура алгоритмов представляет собой математическую иерархию. Множество алгоритмов всех уровней архитектуры обозначается как $\mathcal{A} = \bigcup\limits_{i=\overline{1,5}} \mathcal{L}_i$.

Вычислительной (средней вычислительной) сложностью алгоритма a является функция $L_a(n)$ ($\hat{L}_a(n)$), определяющая зависимость времени (среднего времени) работы алгоритма a от входного параметра n [3].

Введем функцию $\mathcal{S}(a)$, $a \in \mathcal{A}$, определяющую объем дополнительной памяти для хранения данных, которые связаны с предварительными вычислениями, и необходимой для работы алгоритма $a : \mathcal{S} : \mathcal{A} \longrightarrow \mathbb{Z}^+$. Тогда наиболее эффективным алгоритмом $a \in \mathcal{L}_i$, полученным композицией $\mathcal{F}_a$, является такой алгоритм, что

$$L_a(n) = L^i_{\min}(n), \quad \text{при этом} \quad L^i_{\min} = \min_{\substack{c \in \mathcal{L}_i \\ \mathbb{E}}} \{L_c(n)\}, \qquad (1)$$

где $\mathbb{E} : E(\mathbb{F}_p) : y^2 = x^3 + ax^2 + b \pmod p$ — множество эллиптических кривых, утвержденных Национальным институтом стандартов и технологий (National Institute of Standards and Technology, NIST). Далее рассмотрим подмножества кривых $\mathbb{E}$: P_{192}, P_{224}, P_{256}, P_{384} и P_{521}.

В настоящей статье будут приведены алгоритмы, когда при выполнении условия (1) выполняется соотношение $\mathcal{S}(a) \neq 0$.

Описание основных операций в поле $\mathbb{F}_p$ и обозначение их вычислительной сложности представлены в табл. 1 [4].

Таблица 1

Обозначение вычислительной сложности операций в простом поле

Операция	Описание
A	Сложение/вычитание (**a**ddition/subtraction) в поле $\mathbb{F}_p$
R	Приведение по модулю (**r**eduction) p
S	Возведение в квадрат (**s**quaring) в поле $\mathbb{F}_p$
M	Умножение в поле (**m**ultiplication) в поле $\mathbb{F}_p$
I	Нахождение мультипликативного обратного элемента (**i**nversion) в поле $\mathbb{F}_p$

Эффективные алгоритмы вычисления скалярного умножения точки. Перечислим полученные в процессе настоящего исследования алгоритмы.

Алгоритм на основе метода несовместной формы представления скаляра с окном. Предварительные вычисления повышают эффективность вычисления кратной точки. При $w \geq 2$ несовместное представление (wNAF) целого d по основанию 2 имеет вид $d = \sum_{i=0}^{m-1} k_i 2^i$, где $w \geq 2$, k_i это простое, $|d_i| < 2^{w-1}$ [3].

Преимуществом указанного метода является то, что число ненулевых чисел и, следовательно, необходимых сложений в ходе вычислений невелико. Предлагаемый алгоритм $a_{w\mathrm{NAF}}$ (алгоритм 1) скалярного умножения точки для данного wNAF-представления скаляра основан на этом свойстве.

Алгоритм 1. Эффективное вычисление скалярного умножения точки dP, $d \in \mathbb{Z}$, $P \in E(\mathbb{F}_p)$, $w\mathrm{NAF}(d) = (d_{n-1}, \ldots, d_0)$ на основе метода несовместной формы представления скаляра d с окном w.

Вход:

— эллиптическая кривая $E(\mathbb{F}_p)$;

— точка $P \in E(\mathbb{F}_p)$, $P = (x, y)$, $x, y \in \mathbb{F}_p$;

— представление $w\mathrm{NAF}\,(d) = (d_{n-1}, \ldots, d_0)$;

— окно $w \in \mathbb{Z}^+$.

Выход:

— результат скалярного умножения $dP = (x, y)$, $x, y \in \mathbb{F}_p$.

Эффективный алгоритм скалярного умножения точки dP, $d \in \mathbb{Z}$, $P \in E\,(\mathbb{F}_p)$, на основе метода несовместной формы представления скаляра с окном w приведен ниже:

begin
/ Предварительные вычисления */*
1. $P_1 \leftarrow P$
2. $P_2 \leftarrow \mathbf{dbl}\,(P)$
3. **foreach** $(i = 3, 5, \ldots, 2^{w-1} - 1)$ {
4. $P_i \leftarrow \mathbf{add}\,(P_{i-2}, P_2)$
5. }

/ Основные вычисления */*
6. $Q \leftarrow P_{n-1}$
7. **while** $(i \geq 0)$ {
8. **if** $(d_i > 0)$ {
9. $Q \leftarrow \mathbf{da}\,(Q, P_{d_i})$
10. } **else** {
11. **if** $(d_i < 0)$ {
12. $Q \leftarrow \mathbf{d}a\,(Q, -P_{d_i})$
13. } **else** {
14. $Q \leftarrow \mathbf{dbl}\,(Q)$
15. }
16. }
17. $i \leftarrow i - 1$
18. }
19. **return**(Q)

end

Методы несовместной формы представления скаляра с окном, использования композитных операций DBL, ADD, DA, использования свойств аффинной, Якоби и Якоби – Чудновского координатных систем. Утверждение 1. *Алгоритм* $a_{w\mathrm{NAF}}$ *(алгоритм 1) вычисления скалярного умножения точки* dP, $d \in \mathbb{Z}$, $P \in E\,(\mathbb{F}_p)$, $d = (d_{n-1}, \ldots, d_0)_2$ *имеет среднюю вычислительную сложность*:

$$\hat{L}_{a_{w\mathrm{NAF}}}(n, w) = \mathbf{I} + \left(\frac{11n}{w+1} + 2n - 10\right)\mathbf{M} + \left(8n - \frac{3n}{w+1} - 8\right)\mathbf{S},$$

w — ширина окна предварительных вычислений. При этом потребуется $2^{w-2} - 1$ *предварительных вычислений точек. Вычислительная*

сложность предварительных вычислений составит $(11 \cdot 2^{w-2} - 11)\,\mathbf{M} + (3 \cdot 2^{w-2} + 2)\,\mathbf{S}$.

◀ Плотность несовместного представления с окном w (wNAF) скаляров длиной n примерно равна $\dfrac{n}{w+1}$, где $n \in \mathbb{Z}^+$ — длина разложения скаляра $d \in \mathbb{Z}$; $w \in \mathbb{Z}^+$ — ширина окна. Это означает, что вычислительная сложность алгоритма $a_{w\mathrm{NAF}}$ равна

$$\hat{L}_{a_{w\mathrm{NAF}}}(n, w) = \left(\mathbf{DBL} + \left(2^{w-2} - 1\right)\mathbf{ADD}\right) + \left(\frac{n}{w+1} - 1\right)\mathbf{ADD} + (n-1)\,\mathbf{DBL}.$$

Кроме того, необходима память для хранения $2^{w-2} - 1$ точек. В алгоритме $a_{w\mathrm{NAF}}$ существует множество возможностей для использования свойств координатных систем.

Рассмотрим возможности этапа предварительных вычислений точек. Этот этап содержит большое количество сложений значений $2P$, результирующие значения затем используются в последующих вычислениях. Это свидетельствует о том, что результат эффективнее хранить в координатах Якоби – Чудновского. Существует два варианта получения значений $2P$: значение может быть вычислено как $2\mathcal{A} \longrightarrow \mathcal{J}^c$ или как $2\mathcal{A} \longrightarrow \mathcal{A}$. В первом случае каждое сложение в предварительных вычислениях будет иметь вид $\mathcal{J}^c \boxplus \mathcal{J}^c \longrightarrow \mathcal{J}^c$ и вычислительную сложность, равную $11\mathbf{M} + 3\mathbf{S}$. Если значение $2P$ сохранять в аффинных координатах, то каждое сложение будет иметь вид $\mathcal{J}^c \boxplus \mathcal{A} \longrightarrow \mathcal{J}^c$ и вычислительную сложность, равную $8\mathbf{M} + 3\mathbf{S}$. Компромисс заключается в том, что $2\mathcal{A} \longrightarrow \mathcal{A}$ имеет дорогостоящий шаг с нахождением обратного элемента. Даже если предположить, что $\mathbf{I} < 30\mathbf{M}$ [3, 5–10], то ширина окна w должна равняться, по меньшей мере, шести, для большей эффективности. Поэтому примем, что для алгоритма шаг $P_2 \leftarrow 2P$ имеет структуру $2\mathcal{A} \longrightarrow \mathcal{J}^c$, а шаг $P_i \leftarrow P_{i-2} \boxplus P_2$ — структуру $\mathcal{A} \boxplus \mathcal{J}^c \longrightarrow \mathcal{J}^c$ для $i = 3$, и $\mathcal{J}^c \boxplus \mathcal{J}^c \longrightarrow \mathcal{J}^c$ в остальных случаях.

Для основного этапа вычислений потребуется $w + 1$ удвоений и сложений, поэтому координаты Якоби используются для временны́х переменных. Если предварительно вычисленные значения хранятся в координатах Якоби – Чудновского, то шаг $2P \boxplus Q$ (для значений отличных от ± 1) будет иметь структуру $2\mathcal{J} \boxplus \mathcal{J}^c \longrightarrow \mathcal{J}$. Сложность такой операции составит $15\mathbf{M} + 7\mathbf{S}$. Еще одним важным дополнением к работе алгоритма является использование группового обратного элемента для преобразования предварительно вычисленных значений к аффинным координатам. Для этого применим метод Монтгомери со сложностью вычислений $\mathbf{I} + 3(k-1)\mathbf{M}$ для k элементов поля $\mathbb{F}_p$. Преобразование предварительно вычисленных точек к аффинным координатам позволит каждому сложению использовать операцию со

структурой $2\mathcal{J} \boxplus \mathcal{A} \longrightarrow \mathcal{J}$. Таким образом, ее вычислительная сложность составит $13\mathbf{M} + 5\mathbf{S}$. Компромисс в использовании указанного подхода зависит от мощности множества предварительно вычисленных точек и количества необходимых сложений.

Если предварительно вычисленные точки конвертируются к аффинным координатам для алгоритма, то необходимо $(\mathbf{I}+3\,(2^{w-2}-2)\,\mathbf{M})$ для расчета $Z_1^{-1}, Z_3^{-1}, \ldots, Z_{2^{w-1}-1}^{-1}$, используя групповой способ нахождений обратного элемента и $(3 \cdot 2^{w-2}+2)\,\mathbf{M} + (2^{w-2}-2)\,\mathbf{S}$ для определения аффинной формы для каждого $P_i = \left(X_i Z_i^2, Y_i Z_i^{-3}\right)$. После того, как все предварительно вычисленные точки переведены в аффинные координаты, на шаге $Q \leftarrow 2Q \boxplus P_{k_i}$ выполняется операция **DA** со структурой $2\mathcal{J} \boxplus \mathcal{A} \longrightarrow \mathcal{J}$, а на шаге $Q \leftarrow 2Q \boxminus P_{k_i}$ — со структурой $2\mathcal{J} \boxminus \mathcal{A} \longrightarrow \mathcal{J}$. Шаг $Q \leftarrow 2Q$ осуществляется $2\mathcal{A} \longrightarrow \mathcal{J}$ в первый раз и $2\mathcal{J} \longrightarrow \mathcal{J}$ в остальных случаях. Последний шаг это $\mathcal{J} \longrightarrow \mathcal{A}$.

Если $d_i \neq 0$, то $d_i = \pm 1$ с вероятностью около $1/2^{w-3}$. Вычислительная сложность предварительных вычислений составляет $L_{a_{w\mathrm{NAF}}^{PreC}}(n,w) = (3\mathbf{M}+5\mathbf{S}) + (8\mathbf{M}+3\mathbf{S}) + (2^{w-2}-2)\,(11\mathbf{M}+3\mathbf{S}) = (11 \cdot 2^{w-2}-11)\,\mathbf{M} + (3 \cdot 2^{w-2}+2)$.

Приведение рассчитанных значений точек в аффинную форму имеет вычислительную сложность, равную $L_{a_{w\mathrm{NAF}}^{PreC}}(n,w) = \mathbf{I} + (6 \cdot (2^{w-2}-1)-3)\,\mathbf{M} + (2^{w-2}-1)\,\mathbf{S}$.

С предварительно вычисленными точками в форме Якоби – Чудновского остальная часть алгоритма имеет сложность

$$\hat{L}_{a_{w\mathrm{NAF}}}(n,w) = \left(\frac{1}{2^{w-2}}\right)\left(\frac{n}{w+1}-1\right)(13\mathbf{M}+5\mathbf{S}) + \left(\frac{2^{w-2}-1}{2^{w-2}}\right)\left(\frac{n}{w+1}-1\right)(15\mathbf{M}+7\mathbf{S}) + (2\mathbf{M}+4\mathbf{S}) + \left(n-\frac{n}{w+1}-1\right)(4\mathbf{M}+4\mathbf{S}) + (\mathbf{I}+3\mathbf{M}+\mathbf{S}).$$

После приведения подобных слагаемых получим

$$\hat{L}_{a_{w\mathrm{NAF}}}(n,w) = \mathbf{I} + \left(\left(\frac{1}{2^{w-2}}\right)\left(\frac{n}{w+1}-1\right)\left(13+15\left(2^{w-2}-1\right)\right) + 2\left(n-\frac{n}{w+1}-1\right)+5\right)\mathbf{M} + \left(\left(\frac{1}{2^{w-2}}\right)\left(\frac{n}{w+1}-1\right)(5+7(2^{w-2}-1)) + 8\left(n-\frac{n}{w+1}-1\right)+5\right)\mathbf{S}.$$

С предварительно вычисленными точками в аффинной форме, остальная часть алгоритма имеет сложность

$$\hat{L}_{a_w\mathrm{NAF}}(n,w)=\left(\frac{n}{w+1}-1\right)(13\mathbf{M}+5\mathbf{S})+(2\mathbf{M}+4\mathbf{S})+$$
$$+\left(n-\frac{n}{w+1}-1\right)(2\mathbf{M}+8\mathbf{S})+(\mathbf{I}+3\mathbf{M}+\mathbf{S}).$$
$$+\hat{L}_{a_w\mathrm{NAF}}(n,w)=\mathbf{I}+\left(\frac{11n}{w+1}+2n-10\right)\mathbf{M}+\left(8n-\frac{3n}{w+1}-8\right)\mathbf{S}. \blacktriangleright$$

В алгоритме 1 требуется такое же число промежуточных операций в простом поле $\mathbb{F}_p$, что и при операциях нахождения обратного элемента по умножению в простом поле. Для некоторых реализаций это представляет собой проблему. Возможным решением может быть сохранение промежуточных результатов в координатах Z^2, Z^3.

Для предложенного алгоритма $a_w\mathrm{NAF}$ выполняется соотношение $\mathcal{S}(a_w\mathrm{NAF})=2^{w-2}-1$.

Сравнительная оценка вычислительной сложности алгоритма, полученного Д. Хенкерсоном, и предлагаемого алгоритма $a_w\mathrm{NAF}$ вычисления скалярного умножения точки на основе метода несовместной формы представления скаляра с шириной окна w при различных значениях ширины w на подмножестве эллиптических кривых $P_{192}\subset\mathbb{E}$ представлена в табл. 2 (рис. 2).

При увеличении ширины окна w вычислительные затраты основной части алгоритма сокращаются за счет их смещения на этап, связанный с предварительными вычислениями.

Таблица 2

Средняя вычислительная сложность предлагаемого алгоритма $a_w\mathrm{NAF}$ и алгоритма Хенкерсона, $P_{192}\subset\mathbb{E}$

w	Алгоритм Хенкерсона $\hat{L}_{a_w\mathrm{NAF}}(n)$	Предлагаемый алгоритм $\hat{L}_{a_w\mathrm{NAF}}(n)$	$\mathcal{S}(a_w\mathrm{NAF})$
3	$2\mathbf{I}+1155{,}0\mathbf{M}+915{,}0\mathbf{S}$	$2\mathbf{I}+902{,}0\mathbf{M}+1384{,}0\mathbf{S}$	1
4	$2\mathbf{I}+1112{,}2\mathbf{M}+894{,}2\mathbf{S}$	$2\mathbf{I}+796{,}0\mathbf{M}+1412{,}0\mathbf{S}$	3
5	$2\mathbf{I}+1129{,}0\mathbf{M}+891{,}0\mathbf{S}$	$2\mathbf{I}+726{,}0\mathbf{M}+1432{,}0\mathbf{S}$	7
6	$2\mathbf{I}+1228{,}4\mathbf{M}+909{,}3\mathbf{S}$	$2\mathbf{I}+675{,}7\mathbf{M}+1445{,}7\mathbf{S}$	15
7	$2\mathbf{I}+1473{,}0\mathbf{M}+963{,}0\mathbf{S}$	$2\mathbf{I}+638{,}0\mathbf{M}+1456{,}0\mathbf{S}$	21

Алгоритм на основе метода несовместной формы представления скаляра со скользящим окном. Метод несовместной формы представления скаляра со скользящим окном основан на wNAF-представлении скаляра. Набор чисел представления по основанию 2 имеет вид $\{-2^{w-1}+1,-2^{w-1}+3,\ldots,-1,1,\ldots,2^{w-1}-3,2^{w-1}-1\}$. Таким образом, будут вычислены точки $3P$, $5P$,…,$(2^{w-1}-1)P$.

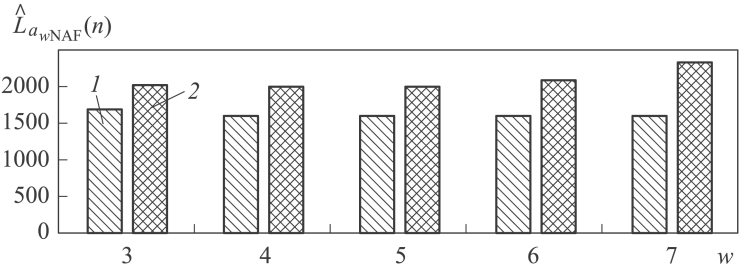

Рис. 2. Сравнение средней вычислительной сложности предлагаемого алгоритма $a_{w\mathrm{NAF}}$ (*1*) и алгоритма Хенкерсона (*2*) скалярного умножения точки на подмножестве эллиптических кривых $P_{192} \subset \mathbb{E}$, $w = 3, \ldots, 7$

Рассматриваемый метод имеет преимущество над NAF-методом, поскольку в NAF-представлении целого не присутствует двух последовательных ненулевых значений. Поэтому для любых значений ширины w из NAF-представления числа d, число представителей будет принадлежать к интервалу $(-2^{w+1}/3; 2^{w+1}/3)$ [3].

Выбор значения w определяет число точек, которые необходимо предварительно рассчитать. Когда выбирается небольшая ширина окна w, то число точек, которые необходимо предварительно рассчитать, приблизительно на треть больше, чем в wNAF-методе с параметром w $(2^{w+1}/3 \;\; 2^{w+1}/4)$. Преимущество такого метода заключается в том, что средняя плотность ненулевых чисел меньше, чем в wNAF-методе. Следовательно, количество сложений, необходимых для вычисления скалярного умножения, уменьшается. Предлагаемый алгоритм $a_{s\backslash w\mathrm{NAF}}$ (алгоритм 2) основан на методе несовместной формы представления скаляра со скользящим окном.

Алгоритм 2. Эффективное вычисление скалярного умножения точки dP, $d \in \mathbb{Z}$, $P \in E(\mathbb{F}_p)$ на основе метода несовместной формы представления скаляра d с окном w.

Вход:

— эллиптическая кривая $E(\mathbb{F}_p)$;
— точка $P \in E(\mathbb{F}_p)$, $P = (x, y)$, $x, y \in \mathbb{F}_p$;
— окно w;
— представление $w\mathrm{NAF}(d) = (d_{n-1}, \ldots, d_0)$.

Выход:

— результат скалярного умножения $dP = (x, y)$, $x, y \in \mathbb{F}_p$.

Эффективный алгоритм скалярного умножения точки dP, $d \in \mathbb{Z}$, $P \in E(\mathbb{F}_p)$, на основе метода несовместной формы представления скаляра с окном приведен ниже:

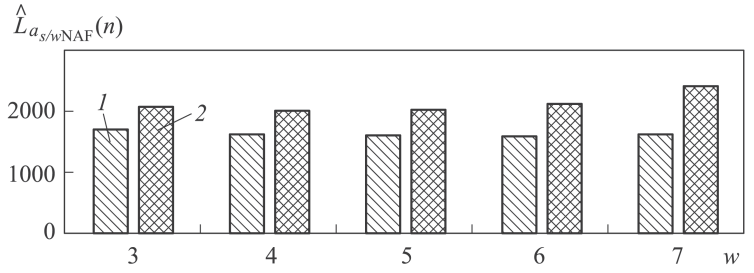

Рис. 3. Сравнение средней вычислительной сложности предлагаемого алгоритма $a_{s\backslash w\mathbf{NAF}}$ (*1*) и алгоритма Хенкерсона (*2*) скалярного умножения точки на подмножестве эллиптических кривых $P_{\mathbf{192}} \subset \mathbb{E}$

begin
/ Предварительные вычисления */*
1. $P_1 \leftarrow P$
2. $P_2 \leftarrow \mathbf{dbl}\,(P)$
3. **foreach** $(i = 3, 5, \ldots, 2\,(2^w - (-1)^w)/3 - 1)$ {
4. $P_i \leftarrow \mathbf{add}\,(P_{i-2}, P_2)$
5. }

/ Основные вычисления */*
6. $Q \leftarrow \mathcal{O}$
7. $i \leftarrow d - 1$
8. **while** $(i \geq 0)$ {
9. $j \leftarrow 0$
10. $z \leftarrow i$
11. **while** $(j \leq w)$ {
12. **if** $((d_i, \ldots, d_{i-j+1})\ \mathbf{mod}\ 2 \equiv 0)$ {
13. $z \leftarrow j$
14. }
15. $j \leftarrow j + 1$
16. }
17. $Q \leftarrow 2^{z-1}Q$
18. **if** $(m > 0)$ {
19. $Q \leftarrow \mathbf{d}a\,(Q, P_m)$
20. } **else** {
21. **if** $(m < 0)$ {
22. $Q \leftarrow \mathbf{d}a\,(Q, -P_{-m})$
23. } **else** {
24. $Q \leftarrow \mathbf{d}bl\,(Q)$
25. }
26. }
27. $i \leftarrow i - t$
28. }
29. **return**(Q)

end

Методы несовместной форм представления скаляра со скользящим окном, использования композитных операций DBL, ADD, DA, использования свойств аффинных, Якоби и Якоби – Чудновского координатных систем. Утверждение 2. *Алгоритм* $a_{s\backslash w\mathrm{NAF}}$ *(алгоритм 2) вычисления скалярного умножения точки* dP, $d \in \mathbb{Z}$, $P \in E(\mathbb{F}_p)$, $d = (d_{n-1}, \ldots, d_0)_2$, *имеет среднюю вычислительную сложность*:

$$\hat{L}_{a_{s\backslash w\mathrm{NAF}}}(n,w) = \mathbf{I} + \left(\frac{11n}{w+v(w)} + 2n - 10\right)\mathbf{M} + \left(8n - \frac{3n}{w+v(w)} - 8\right)\mathbf{S},$$

где w *— ширина окна предварительных вычислений;* $v(w) = \frac{4}{3} - \frac{(-1)^w}{32^{w-2}}$. *При этом потребуется* $\frac{4}{3}(2^{w-2}-1)$ *предварительных вычислений точек. Вычислительная сложность предварительных вычислений составит*

$$L_{a^{PreC}_{s\backslash w\mathrm{NAF}}}(n,w) = \left(11 \cdot \frac{2^w - (-1)^w}{3} - 11\right)\mathbf{M} + \left(3 \cdot \frac{2^w - (-1)^w}{3} + 2\right)\mathbf{S}.$$

◄ Согласно утверждению Хенкерсона [4], средняя длина последовательности нулей между окнами шириной w равна $v(w) = \frac{4}{3} - \frac{(-1)^w}{3 \cdot 2^{w-2}}$, а вычислительная сложность алгоритма предварительных вычислений —

$$L_{a^{PreC}_{s\backslash w\mathrm{NAF}}}(n,w) = \left(\mathbf{DBL} + \left(\frac{2^w - (-1)^w}{3} - 1\right)\mathbf{ADD}\right) + \\ + \left(\left(\frac{n}{w+v(w)} - 1\right)\mathbf{ADD} + (n-1)\,\mathbf{DBL}\right),$$

или

$$L_{a^{PreC}_{s\backslash w\mathrm{NAF}}}(n,w) = (3\mathbf{M}+5\mathbf{S}) + (8\mathbf{M}+3\mathbf{S}) + \left(\frac{2^w - (-1)^w}{3} - 2\right)(11\mathbf{M}+3\mathbf{S}).$$

С учетом конвертации значений в аффинную форму вычислительная сложность составит

$$L_{a^{PreC}_{s\backslash w\mathrm{NAF}}}(n,w) = \mathbf{I} + \left(6\left(\frac{2^w - (-1)^w}{3} - 1\right) - 3\right)\mathbf{M} + \left(\frac{2^w - (-1)^w}{3} - 1\right)\mathbf{S}.$$

При сохранении предварительно вычисленных точек в координатах Якоби – Чудновского, последующий этап алгоритма имеет среднюю вычислительную сложность

$$\hat{L}_{a_{s\backslash w\mathrm{NAF}}}(n,w) = \left(\frac{6}{2^w - (-1)^w}\right)\left(\frac{n}{w+v(w)} - 1\right)(13\mathbf{M}+5\mathbf{S}) +$$

$$+\left(1-\frac{6}{2^w-(-1)^w}\right)\left(\frac{n}{w+v\,(w)}-1\right)(15\mathbf{M}+7\mathbf{S})+(2\mathbf{M}+4\mathbf{S})+$$
$$+\left(n-\frac{n}{w+v\,(w)}-1\right)(2\mathbf{M}+8\mathbf{S})+(\mathbf{I}+3\mathbf{M}+\mathbf{S}),$$

а при преобразовании в аффинную форму среднюю вычислительную сложность —

$$\hat{L}_{a_{s\setminus w\mathrm{NAF}}}(n,w)=\left(\frac{n}{w+v\,(w)}-1\right)(13\mathbf{M}+5\mathbf{S})+(2\mathbf{M}+4\mathbf{S})+$$
$$+\left(n-\frac{n}{w+v\,(w)}-1\right)(2\mathbf{M}+8\mathbf{S})+(\mathbf{I}+3\mathbf{M}+\mathbf{S})=$$
$$=\mathbf{I}+\left(\frac{11n}{w+v\,(w)}+2n-10\right)+\left(8n-\frac{3n}{w+v\,(w)}-8\right)\mathbf{S}. \blacktriangleright$$

Сравнительная оценка средней вычислительной сложности алгоритма Хенкерсона и предлагаемого алгоритма $a_{s\setminus w\mathrm{NAF}}$ вычисления скалярного умножения точки на основе метода несовместной формы представления скаляра со скользящим окном w на подмножествах эллиптических кривых $P_{192}\subset\mathbb{E}$ представлена в табл. 3 (рис. 3).

Таблица 3

Средняя вычислительная сложность предлагаемого алгоритма $a_{s\setminus w\mathrm{NAF}}$ и алгоритма Хенкерсона, $P_{192}\subset\mathbb{E}$

w	Алгоритм Хенкерсона $\hat{L}_{a_{s\setminus w\mathrm{NAF}}}(n)$	Предлагаемый алгоритм $\hat{L}_{a_{s\setminus w\mathrm{NAF}}}(n)$	$\mathcal{S}\left(a_{s\setminus w\mathrm{NAF}}\right)$
3	$2\mathbf{I}+1129{,}3\mathbf{M}+903{,}0\mathbf{S}$	$2\mathbf{I}+843{,}3\mathbf{M}+1400{,}0\mathbf{S}$	2
4	$2\mathbf{I}+1114{,}6\mathbf{M}+892{,}7\mathbf{S}$	$2\mathbf{I}+776{,}3\mathbf{M}+1418{,}3\mathbf{S}$	4
5	$2\mathbf{I}+1164{,}9\mathbf{M}+897{,}4\mathbf{S}$	$2\mathbf{I}+705{,}3\mathbf{M}+1437{,}6\mathbf{S}$	10
6	$2\mathbf{I}+1304{,}1\mathbf{M}+925{,}8\mathbf{S}$	$2\mathbf{I}+662{,}8\mathbf{M}+1449{,}2\mathbf{S}$	20
7	$2\mathbf{I}+1652{,}1\mathbf{M}+1004{,}0\mathbf{S}$	$2\mathbf{I}+627{,}1\mathbf{M}+1459{,}0\mathbf{S}$	42

Заключение. Проанализирован метод несовместной формы представления скаляра с окном w (wNAF) и на его основе получен новый более быстрый, чем wNAF-алгоритм Хенкерсона скалярного умножения точки эллиптической кривой $a_{w\mathrm{NAF}}$. Доказано утверждение относительно вычислительной сложности предложенного алгоритма $a_{w\mathrm{NAF}}$. Для алгоритма $a_{w\mathrm{NAF}}$ выполнено соотношение $\mathcal{S}\left(a_{w\mathrm{NAF}}\right)=2^{w-2}-1$.

Новая, более низкая оценка получена за счет использования следующих операций: $2\mathcal{A}\longrightarrow\mathcal{J}^c$; $\mathcal{A}\boxplus\mathcal{J}^c\longrightarrow\mathcal{J}^c$; $\mathcal{J}^c\boxplus\mathcal{J}^c\longrightarrow\mathcal{J}^c$; $2\mathcal{J}\boxplus\mathcal{J}^c\longrightarrow\mathcal{J}$; $2\mathcal{J}\boxplus\mathcal{A}\longrightarrow\mathcal{J}$; $2\mathcal{J}\boxminus\mathcal{A}\longrightarrow\mathcal{J}$; $2\mathcal{A}\longrightarrow\mathcal{J}$; $2\mathcal{J}\longrightarrow\mathcal{J}$; $\mathcal{J}\longrightarrow\mathcal{A}$ ($\mathcal{A}$, $\mathcal{J}^c$ и $\mathcal{J}$ — представление точки в аффинных, Якоби – Чудновского и Якоби координатных системах); свойств

координат Якоби; метода замены умножений в поле $\mathbb{F}_p$ возведением в квадрат; метода Монтгомери и быстрых композитных операций **DA** и **T** (“утроения”). Операция $c \longrightarrow \tilde{c}$ обозначает преобразование и сохранение точки из координат c в $\tilde{c}$, если c не совпадают с $\tilde{c}$ и только сохранение, если совпадают.

Проанализирован метод несовместной формы представления скаляра со скользящим окном w ($s\backslash w\mathrm{NAF}$) и на его основе получен новый более быстрый, чем известный $s\backslash w\mathrm{NAF}$-алгоритм Хенкерсона скалярного умножения точки эллиптической кривой $a_{s\backslash w\mathrm{NAF}}$. Доказано утверждение относительно вычислительной сложности предложенного алгоритма $a_{s\backslash w\mathrm{NAF}}$. Для алгоритма $a_{s\backslash w\mathrm{NAF}}$ выполняется соотношение $\mathcal{S}\left(a_{s\backslash w\mathrm{NAF}}\right) = \frac{4}{3}\left(2^{w-2} - 1\right)$.

ЛИТЕРАТУРА

1. *Hankerson D.*, *Menezes A.*, *Vanstone S.* Guide to elliptic curve cryptography. Springer-Verlag, 2004.
2. *Matthieu R.* Fast and regular algorithms for scalar multiplication over elliptic curves // IACR Cryptology. 2011. No. 338.
3. *Rokhlin V.*, *Tygert M.* Fast algorithms for spherical harmonic expansions // SIAM J. Sci. Computing. 2008. No. 27 (6). P. 1903–1928.
4. *Hisil H.*, *Carter G.*. Ed. Dawson E. New formulae for efficient elliptic curve arithmetic. 2007.
5. *Bernstein Daniel J.*, *Lange T.* Explicit-formulas database. 2014. URL: http://hyperelliptic.org/EFD (дата обращения: 03.01.2015).
6. *Хлебородов Д.С.* Эффективные алгоритмы скалярного умножения точки эллиптической кривой без предварительных вычислений // Глобальный научный потенциал. 2014. № 5 (38). 35 с.
7. *Björn F.* Double-and-add with relative Jacobian coordinates // Cryptology. 2014. ePrint Archive.
8. *Bernstein Daniel J.*, *Lange T.* Faster addition and doubling on elliptic curves. 2007. P. 29–50.
9. *Dygin D.M.*, *Grebnev S.V.* Efficient implementation of the GOST R 34.10 digital signature scheme using modern approaches to elliptic curve scalar multiplication, 2013.
10. *Edwards H.M.* A normal form for elliptic curves // Bulletin of the American Mathematical Society. 2007. No. 44. P. 393–422. URL: http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html (дата обращения: 18.03.2015).

REFERENCES

[1] Hankerson D., Menezes A., Vanstone S. Guide to elliptic curve cryptography. Springer-Verlag, 2004.

[2] Matthieu R. Fast and regular algorithms for scalar multiplication over elliptic curves. *IACR Cryptology*, 2011, no. 338.

[3] Rokhlin V., Tygert M. Fast algorithms for spherical harmonic expansions. *SIAM J. Sci. Computing*, 2008, no. 27 (6), pp. 1903–1928.

[4] Hisil H., Carter G., Ed.: Dawson E. New formulae for efficient elliptic curve arithmetic, 2007.

[5] Bernstein Daniel J., Lange T. Explicit-formulas database, 2014. URL: http://hyperelliptic.org/EFD (accessed: 03.01.2015).
[6] Khleborodov D.S. Efficient Algorithms for Scalar Multiplication of an Elliptic Curve Point without Precomputation. *Global'nyy nauchnyy potentsial* [Global Research Potential], 2014, no. 5 (38). 35 p.
[7] *Björn F*. Double-and-add with relative Jacobian coordinates. *Cryptology*, 2014. ePrint Archive.
[8] Bernstein Daniel J., Lange T. Faster addition and doubling on elliptic curves. 2007, pp. 29–50.
[9] Dygin D.M., Grebnev S.V. Efficient implementation of the GOST R 34.10 digital signature scheme using modern approaches to elliptic curve scalar multiplication, 2013.
[10] Edwards H.M. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 2007, no. 44, pp. 393–422. URL: http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html (accessed: 18.03.2015).

Статья поступила в редакцию 08.03.2015

Хлебородов Денис Сергеевич — аспирант кафедры "Информационная безопасность" МГТУ им. Н.Э. Баумана. Автор трех научных работ в области быстрых алгоритмов вычисления преобразований на основе эллиптических кривых.
МГТУ им. Н.Э. Баумана, Российская Федерация, 105005 Москва, 2-я Бауманская ул., д. 5.

Khleborodov D.S. — postgraduate of the Information Security Department of the Bauman Moscow State Technical University. Author of three publications in the field of fast computation algorithms of transformations based on elliptic curves.
Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

Просьба ссылаться на эту статью следующим образом:

Хлебородов Д.С. Быстрые алгоритмы вычисления преобразований на основе эллиптических кривых с предварительными вычислениями // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2015. № 3. С. 65–78.

Please cite this article in English as:

Khleborodov D.S. Fast computation algorithms of transformations based on elliptic curves with precomputations. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2015, no. 3, pp. 65–78.