УДК 512.624.9

Сравнительный анализ вычислений с использованием сочетаний различных базисов конечных полей

С.Б. Гашков, А.Б. Фролов

Гашков Сергей Борисович — доктор физико-математических наук, профессор кафедры дискретной математики Московского государственного университета им. М.В. Ломоносова, e-mail: sbgashkov@gmail.com
Фролов Александр Борисович — доктор технических наук, профессор кафедры математического моделирования НИУ «МЭИ», e-mail: frolov@.mail.ru

Представлен сравнительный анализ сложности реализации алгоритмов обращения, возведения в степень в полях характеристики два, операции спаривания Тейта и финального экспоненцирования на суперсингулярной эллиптической кривой над такими полями с учетом возможности использования различных базисов конечных полей, в которых осуществляются вычисления. Используются полиномиальный базис (п.б.) $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$, поля $GF(2^n)$, почти п.б $\{\beta, \beta^2, \ldots, \beta^n\}$, оптимальный нормальный базис (о.н.б.) $\{\beta^{2^0}, \beta^{2^1}, \ldots, \beta^{2^{n-1}}\}$ и переставленный о.н.б. $(\beta^1, \beta^2, \ldots, \beta^n) = \{\beta^{\pi(2^0)}, \beta^{\pi(2^1)}, \ldots, \beta^{\pi(2^{(n-1)})}\}$ (п.о.н.б.) базис поля $GF(2^n)$, где β есть корень минимального многочлена, генератор о.н.б. 2-го или 3-го типа, а также удвоенный п.п.б. $\{\beta, \beta^2, \ldots, \beta^{2n}\}$ и удвоенный п.о.н.б. $\{\beta_1, \beta_2, \ldots, \beta_{2n}\}$ преобразования этих базисов. Используется операция умножения в кольце $GF(2)[X]$, реализованная последовательной программой умножения по алгоритму Карацубы. С использованием этих базисов и операции рассмотрены операции умножения, обращения в возведения в степень в $GF(2^n)$. Показано, что возведение в степень $2^{2^{191}-2}$ для обращения по малой теореме Ферма можно осуществить, используя 12 умножений при ничтожных затратах на возведения в квадрат. Но обращение с использованием модификации расширенного алгоритма Евклида требует существенно меньших элементарных операций $\oplus$, & и битовых присваиваний и даже только логических операций по равнению с экспоненцированием по Ферма, что подтверждается усредненными данными по 100 исполнениям операции обращения. Операции спаривания и финального экспоненцирования выполняются в расширении 4-й степени поля $GF(2^n)$ с использованием его п.б. или о.н.б. 1-го типа при п.б., п.п.б. или п.о.н.б. исходного поля. Показано, что при использовании для умножения многочленов степени 190 в кольце последовательной программой по методу Карацубы в криптографически значимом поле $GF(2^{191})$ для спаривания наилучшее сочетание составляют п.о.н.б. поля $GF(2^n)$ и п.б. его расширения. При умножении по рекордной по числу исполняемых логических операций программе более предпочтительным является сочетание п.б. основного поля и о.н.б. 1-го типа его расширения. Однако существенное преимущество финального экспоненцирования в п.о.н.б. основного поля и о.н.б. 1-го типа его расширения влечет преимущество использования этого базиса основного поля как при спаривании, так и при финальном экспоненцировании, а для эффективной реализации следующего после операции спаривания операции финального экспоненцирования необходимо преобразование из п.б. в о.н.б. расширения поля, что реализуется просто при использовании общего для п.б. и о.н.б. минимального многочлена. Тогда финальное экспоненцирование осуществляется 17-ю умножениями в расширении поля при практически ничтожных затратах на возведения в квадрат в промежуточных вычислениях. Во втором случае спаривание и финальное экспоненцирование осуществляются при одном и том же сочетании базисов основного поля его расширения. Результаты получены на основе анализа первоисточников, алгоритмов и посредством компьютерных экспериментов.

Ключевые слова: конечное поле, расширение конечного поля, оптимальный нормальный базис, сочетание базисов, суперсингулярная эллиптическая кривая, спаривание Тейта, алгоритм спаривания с извлечением квадратных корней, алгоритм спаривания без извлечения квадратных корней, финальное экспоненцирование.

A Comparative Analysis of Calculations Performed Using Combinations of Different Bases of Finite Fields

S.B. Gashkov, A.B. Frolov

Gashkov Sergey B. — Dr.Sci. (Phys.-Math.), professor of Discrete Mathematics Dept., Moscow State University named M.V. Lomonosov, e-mail: sbgashkov@gmail.com
Frolov Aleksandr B. — Dr.Sci. (Techn.), professor of Mathematical Modeling Dept., MPEI, e-mail: frolov@.mail.ru

The article is devoted to a comparative analysis of the complexity of algorithms for carrying out inversion, exponentiation in the fields of characteristic two, Tate pairing operation, and final exponentiation on a supersingular elliptic curve over these fields, taking into account the possibility of using different bases of finite fields in which the calculations are carried out. The polynomial basis (p.b) $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$ of the field $GF(2^n)$, an almost p.b. (a.p.b.) $\{\beta, \beta^2, \ldots, \beta^n\}$, the optimal normal basis (o.n.b.) $\{\beta^{2^0}, \beta^{2^1}, \ldots, \beta^{2^{n-1}}\}$, and the permutated o.n.b. (p.o.n.b) $(\beta^1, \beta^2, \ldots, \beta^n) = \{\beta^{\pi(2^0)}, \beta^{\pi(2^1)}, \ldots, \beta^{\pi(2^{(n-1)})}$ basis of the field $GF(2^n)$ [where β is the root of the minimal polynomial], the generator of the 2nd type or 3rd type o.n.b., as well as the duplicated a.p.b. $\{\beta, \beta^2, \ldots, \beta^{2^n}\}$ and the duplicated p.o.n.b. $\{\beta_1, \beta_2, \ldots, \beta_{2n}\}$, and transformations of these bases are used. Multiplication in the ring $GF(2)[X]$ implemented using a sequential multiplication program according to Karatsuba's algorithm is applied. The operations of multiplication, exponentiation to power $2^{2^{191}-2}$ and inversion in the field $GF(2^n)$ are considered with the use of these bases and this operation. It is shown that exponentiation to power $2^{2^{191}-2}$ for inversion according to Fermat's small theorem can be implemented using 12 multiplications at insignificant expenditures for squaring. At the same time, inversion using a modification of the extended Euclidean algorithm requires a significantly fewer number of elementary operations ⊕, & and bit assignments or even only logical operations in comparison with the exponentiation by Fermat, which is confirmed by the average data on 100 executions of the inversion operation. The operations of pairing and final exponentiation are implemented in the 4th degree extension of the field $GF(2^n)$ using its 1st type o.n.b. or p.b. with p.b., a.p.b or p.o.n.b of the initial field. It is shown that, if for the multiplication of polynomials of degree 190 in the ring a sequential program according to the Karatsuba method is used, the p.o.n.b. of the field $GF(2^{191})$ and the p.b. of its expansion constitute the best combination for pairing in the cryptographically significant field $GF(2^{191})$. In carrying out multiplication using the record beating program (in the number of executable logic operations), the combination involving the p.b. of the main field and the o.n.b. of the 1st type of its expansion is more preferable. However, a significant advantage of the final exponentiation in the p.o.n.b. of the main field and o.n.b. of the 1st type of its expansion entails the advantage of using this basis of the main field both in pairing, and in the final exponentiation, and for the effective implementation of the next operation of the final exponentiation after the pairing operation, it is necessary to make conversion from the p.b. to o.n.b. extension of the field, which is implemented quite easily by using the minimal polynomial common for the p.b. and o.n.b.. Then, the final exponentiation is performed by carrying out 17 multiplications in the field extension at almost negligible cost of squaring in intermediate computations. The results are obtained by analyzing primary sources, algorithms, and via computer experiments.

Key words: finite field, finite field extension, optimal normal basis, combination of bases, supersingular elliptic curve, Tate pairing, pairing algorithm with square root extraction, pairing algorithm without square root extraction, final exponentiation.

Введение

Идея сочетания базисов для ускорения вычислений в конечных полях впервые была представлена в [1] на основе полученных оценок сложности преобразований базисов в полях, имеющих оптимальный нормальный базис 2-го или 3-го типа [2]. В [3—5] описан ряд модификаций умножения в о.н.б. 2-го или 3-го типа. В частности, в [5] описан алгоритм умножения в так называемом оптимальном полиномиальном базисе 2-го типа (в терминологии работы [1] — почти полиномиальном базисе) с использованием операции умножения в кольце $GF(2)[X]$ и конвертированием результата в п.п.б. с использованием переставленного о.н.б., т. е. без операции приведения по модулю неприводимого многочлена. В представленной работе[1] конкретизирована идея использования сочетаний базисов применительно к актуальной задаче ускорения операции спаривания на суперсингулярной эллиптической кривой [6—8] над полем характеристики два, имеющим о.н.б. Напомним, что о.н.б. 1-го типа поля $GF(2^n)$ возникает, если $p = n+1$ является простым числом, а 2 является примитивным корнем по модулю р о.н.б. 2-го или 3-го типа поля $GF(2^n)$ возникает, когда $p = 2n+1$ является простым числом, а характеристика поля 2 является примитивным корнем по модулю p. Если $p \equiv 4 (\mathrm{mod}\ 4)$, а 2 является квадратичным вычетом, то имеем о.н.б. 3-го типа, иначе – о.н.б. 2-го типа.

В настоящей работе использованы поля $GF(2^{191})$ и $GF(2^{11})$, оба имеют о.н.б. 3-го типа. Используемое для реализации алгоритма спаривания расширение 4-й степени этих полей имеет о.н.б 1-го типа. Наличие о.н.б. позволяет ускорить операцию возведения в степень и другие операции, осуществляемые на ее основе, в частности, операцию извлечения квадратного корня. Полиномиальные базисы перечисленных полей имеют генераторы, являющиеся корнями неприводимых трехчленов, что упрощает реализацию операции умножения в этих базисах. Таким образом, имеются основания исследовать возможность совместного использования

[1] Работа выполнена при финансовой поддержке РФФИ, проект 14-01-00671а.

о.н.б. и п.б. В статье приведены данные о количестве элементарных операций при выполнении тех или иных преобразований, полученные посредством экспериментов с использованием компьютерных программ[2].

О базисах конечного поля *GF*(2^n) при существовании оптимального нормального базиса второго или третьего типа

Пусть $p = 2^n + 1$ — простое число, p делит $2^n - 1$. Возьмем $\alpha \in GF(2^n)$, такое, что $\alpha \neq 1$, $\alpha^p = 1$. Пусть 2 является примитивным корнем по модулю p из 1, т. е. множество всех степеней числа 2 по модулю p образует группу Z_p^* или же это множество является множеством всех квадратичных вычетов по модулю p.

Рассмотрим последовательность $\beta_i = \alpha^i + \alpha^{-i} = \alpha^i + \frac{1}{\alpha^i} \in GF(2^{2n})$ для всех целых i (в случае $2n \equiv 1 (\mathrm{mod} p)$, $\beta_i \in GF(2^n)$). Очевидно, что $\beta_i = \beta_{-i}$.

Множество $\{1, \beta_1, \beta_1^2, \ldots, \beta_1^{n-1}\}$ составляет полиномиальный базис (п.б.) поля $GF\{2^n\}$, порождаемый корнем β_1 соответствующего неприводимого полинома $P(X)$. Множество $\{\beta_1, \beta_1^2, \ldots, \beta_1^{n-1}, \beta_1^n\}$ называется почти полиномиальным базисом (п.п.б.). Преобразование элемента $(a_0, a_1, \ldots, a_{n-1})$ из п.б. в п.п.б. осуществляется прибавлением умноженного на коэффициент a_0 вектора коэффициентов многочлена $P(X)$, при этом получается вектор $(b_1, b_2, \ldots, b_{n-1}, a_0)$ (нулевой элемент исчезает, а n-й добавляется). Преобразование элемента $(b_1, b_2, \ldots, b_{n-1}, b_n)$ из п.п.б. в п.б. осуществляется прибавлением умноженного на коэффициент b_n вектора коэффициентов многочлена $P(X)$, при этом получается вектор $(b_n, a_1, \ldots, a_{n-1})$ (нулевой n-й элемент исчезает, а нулевой добавляется). При этих преобразованиях исполняется $n-1$ операция $\oplus$.

Будем использовать также удвоенный п.п.б. $\{\beta_1, \beta_1^2, \ldots, \beta_1^{n-1}, \beta_1^n, \beta_1^{n+1}, \beta_1^{2n-1}\}$.

Множество $\{\xi_1, \ldots, \xi_n\} = \{\beta_1^{2^0}, \beta_1^{2^1}, \ldots, \beta_1^{2^{n-1}}\}$ составляет оптимальный нормальный базис (о.н.б.) 2-го или 3-го типа.

Рассмотрим базис $\{\beta_1, \ldots, \beta_n\} \subseteq GF(q^n)$ Он получается перестановкой π,

$$\pi(j) = \begin{cases} q^j \bmod p, \text{если } q^j \bmod p \le n; \\ p - q^j \bmod p, \text{если } q^j \bmod p > n \end{cases}$$

из о.н.б. $\{\xi_1, \ldots, \xi_n\}$, где $\xi_i = \beta_i^{2^{i-1}}$, $i = 1, \ldots n$ и называется *переставленным* о.н.б. (п.о.н.б.): для $j = 0, \ldots, n-1 \beta_j = \xi_{\pi(j)}$. Обратное преобразование соответствует обратной перестановке π^{-1}: $\xi_j = \beta_{\pi^{-1}(j)}$, $j = 1, \ldots, n$.

Можно проверить, что

$$\sum_{i=1}^{n} \beta_i = \sum_{i=1}^{n} (\alpha^i + \alpha^{-i}) = \sum_{i=1}^{2n} \alpha^i = 1,$$

поскольку

$$\alpha^{-i} = \alpha^{p-i} = \alpha^{2n+1-i}, \quad \sum_{i=0}^{2n} \alpha^i = \sum_{i=0}^{p-1} \alpha^i = \frac{\alpha^p - 1}{\alpha - 1} = 0.$$

Будем использовать также *удвоенный* п.о.н.б. $\{\beta_1, \ldots, \beta_n, \beta_{n+1}, \ldots, \beta_{2n}\}$, $\beta_i = \xi_{\pi(i)}$, $\xi_i = \beta_1^{2^{i-1}}$, $i = 1, \ldots, 2n$.

Приведем алгоритмы преобразования из п.п.б. в п.о.н.б. и из п.о.н.б. в п.п.б. Обозначим $T(\mathbf{a}, r, t)$ следующее преобразование отрезка $[\mathbf{a}[r + 2t], \ldots, \mathbf{a}[r + 4t]]$ вектора $\mathbf{a}$:

b:=**a**
для $i = (1, t - 1)$:
  если $2rt + t + i \le |\mathbf{a}| - 1$:
    $\mathbf{b}[2rt + t - i, s] = (\mathbf{a}[2rt + t - i] \oplus \mathbf{a}[2rt + t + i])$.

Обозначим $T(\mathbf{a}, k)$ последовательное преобразование отрезками длины $t = 2^k$ вектора $\mathbf{a}$ в целом:

$t = 2^k$
$d = \lfloor |\mathbf{a}|/t \rfloor$
  для $r = (0, d - 1)$: $T(\mathbf{a}, r, t)$.

Преобразование $\mathbf{b} = T(\mathbf{a})$ п.п.б. в п.о.н.б:

  b:=**a**
$l = \lfloor |\mathbf{a}| \rfloor - 1$ int(len(lst) – 1)
  $d = \lfloor \log_2 l \rfloor$
  для $k = (1, d+1)$:
    $\mathbf{b} = T(\mathbf{b}, k)$

Преобразование $\mathbf{b} = T^{-1}(\mathbf{a})$ п.о.н.б в п.п.б:

  a:=**b**
$l = \lfloor |\mathbf{b}| \rfloor$
  $d = \lfloor \log_2 l \rfloor$
  для $k = (1, d+1)$:
    $\mathbf{b} = T(\mathbf{b}, d - k)$.

Наконец, потребуется преобразование $D(\mathbf{a})$ представления $\mathbf{a}$ из удвоенного п.о.н.б. в п.о.н.б.:

  для $i = (1, n)$:
    $\mathbf{b}[i] = \mathbf{a}[i] \oplus \mathbf{a}[2n-i+1]$.

При его исполнении выполнятся n операций $\oplus$.

Об операции умножения в *GF*(2^n)

Следуя [5], к элементам поля, представленным в п.п.б., будем применять операцию умножения как к элементам кольца $GF(2)[X]$. Результат при этом получается в удвоенном п.п.б. В экспериментах эту операцию реализуем последовательной программой умножения по алгоритму Карацубы, учитывая, что «свободные» элементы векторов-сомножителей нулевые. Обозначим ее $\times_K$. Также следуя [5], будем обозначать $Bottom(\mathbf{a})$ младшую половину представления элемента $\mathbf{a}$ поля в

[2] Свидетельство о гос. рег. программы для ЭВМ № 2016661242. Программа реализации трехстороннего протокола согласования ключей в компьютерной сети с использованием полиномиального и нормального базисов конечного поля характеристики два / С. Б. Гашков, А. Б. Фролов, А. М. Винников, С. А. Лукин, О. К. Суханова. 2016.

Свидетельство о гос. рег. программы для ЭВМ 2016661241 Программа реализации трехстороннего протокола согласования ключей в компьютерной сети с использованием полиномиального базиса конечного поля характеристики два / А. Б. Фролов, А. М. Винников. 2016.

удвоенном п.п.б. и *Top*(**a**) — представление этого элемента с нулевыми элементами вместо *Bottom*(**a**). Очевидно, что при преобразовании $D(Top(\mathbf{a}))$ вместо операций $\oplus$ исполняются присваивания.

Теперь можно описать две операции умножения в п.п.б.:

$\mathbf{d}_P = MultiplicationapbP(\mathbf{a}, \mathbf{b})$ с результатом в п.п.б. и

$\mathbf{d}N = MultiplicationapbN(\mathbf{a}, \mathbf{b})$ с результатом в п.о.н.б.:

$\mathbf{c} = \mathbf{a}\times_K\mathbf{b}$;

$\mathbf{d}_P = Bottom(\mathbf{c})+T^{-1}(D(T(Top(\mathbf{c}))))$;

$\mathbf{d}_P = T(Bottom(\mathbf{c}))+ D(T(Top(\mathbf{c}))),$

где $\mathbf{c} = \mathbf{a}\times_K\mathbf{b}$, «+» — n-битовая операция $\oplus$.

Операции умножения с исходными операндами в п.о.н.б.:

$MultiplicationaonbP(\mathbf{a}, \mathbf{b})$ с результатом в п.п.б. и

$\mathbf{d}_N = MultiplicationaonbN(\mathbf{a}, \mathbf{b})$ с результатом в п.о.н.б. отличаются тем, что вместо $\mathbf{c} = \mathbf{a}\times_K\mathbf{b}$, используется $\mathbf{c} = T(\mathbf{a})\times_K T(\mathbf{b})$.

Умножение в полиномиальном базисе с использованием полинома $P(X)$, имеющего вес 15, неэффективно из-за сложности приведения по модулю. Поэтому для сравнения сложности операций в п.б. и п.п.б. или п.о.н.б. операции в п.б. реализуются с использованием трехчленов и их корней, порождающих базис. Обозначим $R(\mathbf{a})$ операцию приведения по модулю трехчлена, где **a** — многочлен степени $2n–2$. Тогда операция умножения в п.б. функционально описывается как $\mathbf{d}_{PB} = MultiplicationPB(\mathbf{a}, \mathbf{b}) = R(\mathbf{a}\times^K\mathbf{b})$, где $\times^K$ — операция в кольце $GF(2)[X]$, также выполняемая по последовательной программе умножения по алгоритму Карацубы (отличается от $\times_K$ тем, что «свободные» элементы векторов-сомножителей могут быть и ненулевыми).

Возведение в квадрат в полиномиальном базисе выполняется по алгоритму, непосредственно учитывающему приведение при вычислении элементов вектора-результата. Обозначим эту операцию *SquaringPB*(**a**).

Об операции возведения в степень, равную степени характеристики поля в $GF(2^n)$.

Обозначим $MultysquaringaonbP(\mathbf{a}, j)$ операцию возведения в степень 2^j элемента **a**, представленного в п.о.н.б. с результатом в п.п.б.:

для $i = (1, n)$:

$\mathbf{b}[i] = \mathbf{a}[\pi[\pi^{-1}–j]\bmod n]$; $\mathbf{c} = T(\mathbf{b})$.

Операция $MultysquaringapbP(\mathbf{a}, j)$ отличается тем, что применяется к элементу, представленному в п.п.б. и вместо **a** используется $T(\mathbf{a})$.

Обозначим $MultysquaringPB(\mathbf{a}, j)$ операцию возведения в степень $2j$ элемента **a**, представленного в п.б. с результатом в п.б.:

$\mathbf{c} = \mathbf{a}$;

для $i=(1, j)$: $\mathbf{c}= SquaringPB(\mathbf{c})$.

Об операциях возведения в фиксированную степень и обращения в $GF(2^n)$

Если степень заранее известна, то по методу Брауэра можно построить аддитивную цепочку, позволяющую минимизировать количество исполняемых умножений за счет увеличения параметров j в операциях $MultysquaringaonbP(\mathbf{a}, j)$ или $MultysquaringPB(\mathbf{a}, j)$.

Пример

Реализуем обращение в поле $GF(2^{191})$ по малой теореме Ферма (обращение по Ферма), т. е. возведением элемента a в степень $2^{191}–2$ в п.п.б. При этом используем вспомогательную аддитивную цепочку 1, 2, 4, 8, 9, 18, 19, 38, 76, 152, 190. Будем использовать следующие обозначения операций:

$$MultysquaringonbP(\mathbf{z}, j) = z^{(j)}_{aonbP} = \mathbf{z}^{2j},$$
$$MultysquaringaonbP(\mathbf{z}, j) = z^{(j)}_{aonbP} = \mathbf{z}^{2j},$$

$$MultiplicationapbN(\mathbf{y}, \mathbf{z}) = y\times_{apbN}\mathbf{z},$$
$$MultiplicationapbP(\mathbf{y}, \mathbf{z}) = \mathbf{z}_1 = y\times_{apbP}\mathbf{z}.$$

Программа обращения элемента $\mathbf{a}\in GF(2^{191})$ в п.п.б. представляется последовательностью операций:

$$\mathbf{x} = \mathbf{a};\ \mathbf{y} = \mathbf{z}_{apbP}{}^{(1)} = \mathbf{x}^2;\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{x} = \mathbf{x}^{2^2-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(2)} = \mathbf{x}^{2^4-2^2};\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{z} = \mathbf{x}^{2^4-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(4)} = \mathbf{x}^{2^8-2^4};\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{z} = \mathbf{x}^{2^8-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(1)} = \mathbf{x}^{2^9-2^1};\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{x} = \mathbf{x}^{2^9-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(9)} = \mathbf{x}^{2^{18}-2^9};\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{z} = \mathbf{x}^{2^{18}-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(1)} = \mathbf{x}^{2^{19}-2};\ \mathbf{z} = \mathbf{y}\times_{apbN}\mathbf{x} = \mathbf{x}^{2^{19}-1};$$
$$\mathbf{y} = \mathbf{z}_{aonbP}{}^{(19)} = \mathbf{x}^{2^{38}-2^{19}};\ \mathbf{z}_1 = \mathbf{y}\times_{apbP}\mathbf{z} = \mathbf{x}^{2^{38}-1};$$
$$\mathbf{y} = \mathbf{z}_{1apbP}{}^{(38)} = \mathbf{x}^{2^{76}-2^{38}};\ \mathbf{v} = \mathbf{y}\times\mathbf{z}1_{apbP} = \mathbf{x}^{2^{76}-1};$$
$$\mathbf{y} = \mathbf{v}_{apbP}{}^{(76)} = \mathbf{x}^{2^{152}-2^{76}};\ \mathbf{z} = y\times_{apbP}\mathbf{v} = \mathbf{x}^{2^{152}-1};$$
$$\mathbf{y} = \mathbf{z}_1\times_{apbN}\mathbf{x} = \mathbf{x}^{38};\ \mathbf{v} = \mathbf{y}\times_{apbP}\mathbf{z} = \mathbf{x}^{2^{190}-2^{38}};$$
$$\mathbf{v} = \mathbf{z}_1\times_{\text{apbN}}\mathbf{v} = \mathbf{x}^{2^{190}-1};\ \mathbf{y} = \mathbf{v}_{aonbP}{}^{(1)} = \mathbf{x}^{2^{191}-2} = \mathbf{c}.$$

Обращение в поле $GF(2^{191})$ по Ферма в п.б. осуществляется по аналогичному алгоритму, отличающемуся использованием операций $MultiplicationPB(\mathbf{a}, \mathbf{b})$ и $MultysquaringPB(\mathbf{a}, \mathbf{j})$ вместо соответствующих операций рассмотренного алгоритма.

Альтернативно обращение в п.б. можно выполнить по известной модификации алгоритма Евклида (обращение по Евклиду) [9][3]. Особенностью является то, что не применяются деления с остатком.

Обращение по Евклиду в п.п.б. осуществляется преобразованием исходного элемента из п.п.б. в п.б. и преобразованием результата обращения из п.б. в п.п.б.

Количество исполняемых операций при выполнении некоторых рассмотренных операций в п.п.б в поля $GF(2^{191})$ представлено в табл. 1. В этой таблице и далее

[3] См. также: Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А. Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы.М.: КомКнига. 2012. С. 190.

указаны количества операций ⊕, & и *e* (однобитовые присваивания): суммарные количества операций ⊕ и & в строках {⊕, &} и их общее количество в строках {⊕, &, e}. В скобках указаны относительные значения по сравнению со значениями для операции умножения. Очевидно преимущество умножения по Евклиду (при использовании для умножения упомянутой последовательной программы) как по суммарному числу элементарных операций, так и по суммарному числу только логических операций. В квадратных скобках указано расчетное количество тех же операций в предположении умножения многочленов степени 190 наиболее быстрым из заявленных алгоритмов [10]. При этом видно, что существенное преимущество обращения по Евклиду по числу логических операций сохранилось бы.

Рассмотрим особенности имплементации двух известных алгоритмов [8] с использованием сочетаний различных базисов.

Протокол с операцией извлечения квадратного корня использует преимущества п.о.н.б. поля $GF(2^{191})$. Протокол без извлечения квадратных корней допускает реализацию в п.б. поля $GF(2^n)$ при отсутствии в нем о.н.б. 2-го типа. Здесь оценим затраты элементарных операций при его исполнении в поле $GF(2^{191})$ в сравнении с протоколом с извлечением квадратного корня.

Наряду с операциями, рассмотренными выше, будем использовать операции $MultysquaringapbP\_4n(\mathbf{a}, j) = \mathbf{a}_{apbP\_4nP}^{(j)} = \mathbf{a}_{apbP\_4nP}^{(j)} = \mathbf{a}_{apbP\_4n}^{2j}$ возведение в степень 2^j в расширении 4-й степени поля $GF(2^n)$ при п.п.б. поля $GF(2^n)$ и п.б. его расширения; $Multiplicationapb P_{4n}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \times_{apbP\_4nP} \mathbf{b}$ — умножение при п.п.б. поля $GF(2^n)$и п.б. его расширения 4-й степени; $+_{4n}$ — сложение в расширении 4-й степени поля $GF(2^n)$ в любом базисе последнего. Аналогичные обозначения $\mathbf{a}_{aonbN\_4nP}^{(j)}$, $\mathbf{a}_{aonbN\_4nN}^{(j)}$, $\mathbf{a} \times_{aonbN\_4nP} \mathbf{b}$ $\mathbf{a} \times_{aonbN\_4nN} \mathbf{b}$ используются для операций в расширении 4-й степени поля $GF^{(2n)}$ при других сочетаниях базисов.

Таблица 1

Операция	Умножение $\times_к$	Умножение в поле	Обращение по Ферма	Обращение по Евклиду среднее по 100 примерам
⊕	44892 [14380]	46252 [15740]	561693 (12,14) [195549(12,42)]	87041 (1,88) [87041 (5,52)]
&	5826 [6724]	5826 [6724]	69912 (12,0) [80688 (12,0)]	0(0,0) [0(0,0)]
{⊕, &}	50718 [21104]	52078 [22464]	631605 (12, 12) [276237(12, 42)]	87041 (1, 67) [87041 (3, 87)]
e	22808	23557	281920 (12, 0)	319009 (4, 21)
{⊕, &, e}	73256	75635	913525 (12, 1)	406050 (5, 37)

Об операциях в расширении 4-й степени поля $GF(2^n)$

В поле $GF((2^n)^4)$ — расширении 4-й степени поля $GF(2^n)$ имеются как п.б., так и о.н.б. 1-го типа. Операции сложения и умножения в нем в любом из этих базисов могут строиться с использованием операций в поле $GF(2^n)$ в любом из рассмотренных выше базисов. Операция умножения реализуется по алгоритму Карацубы с приведением по модулю неприводимого многочлена $1+Y+Y^2+Y^3+Y^4$ как в о.н.б. 1-го типа, так и в п.б.

Существенно различаются операции возведения в степень, равную степени характеристики поля. В о.н.б. 1-го типа независимо от базиса поля $GF(2^n)$ возведение в степень 2^j осуществляется циклическим сдвигом 4-элементного вектора на j mod 4 позиций, а в п.б. требуется j-кратное возведение в квадрат. В обоих случаях предварительно элементы исходного вектора возводятся в степень $2j$ в соответствии с базисом, в котором они представлены.

Об операции спаривания на суперсингулярной эллиптической кривой

Операция спаривания на суперсингулярной эллиптической кривой $Y^2=X^3+X+b$, $b\in(0,1)$, над полем $GF(2^n)$ является основной операцией криптографических протоколов, основанных на спаривании [7, 8].

Протокол с извлечением квадратного корня [8]

а) П.п.б. поля $GF(2^{191})$, о.н.б. поля $GF((2^{191})^4)$.

Алгоритм $Pairingapbonb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{apb}, \mathbf{b})$ спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{apb}$ при $\mathbf{b} = 1_{apb}$ (единица в поля $GF(2^n)$ в п.п.б.):

$\mathbf{C} = [1_{apb}, 1_{apb}, 1_{apb}, 1_{apb}]$;
$\mathbf{s} = \mathbf{t}_{apb\ apbP\_4nN}^{(1)}$;
для $i = (1, n)$
$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{apbP}^{(1)}$;
$\boldsymbol{\beta} = \boldsymbol{\alpha}_{apbP}^{(1)}$;
$\mathbf{z} = \boldsymbol{\alpha} + \mathbf{x}$;
$\mathbf{v} = \boldsymbol{\alpha} \times_{apbP} \mathbf{x}$;
$\mathbf{w} = \mathbf{z}+\mathbf{v}+\boldsymbol{\beta}+\mathbf{y}+1_{apb}$;
$\mathbf{u}=[\mathbf{z} \times_{apbP} \mathbf{t}_{apb}[0]);$ $\mathbf{z} \times_{apbP} \mathbf{t}_{apb}[1]);$ $\mathbf{z} \times_{apbP} \mathbf{t}_{apb}[2]);$ $\mathbf{z} \times_{apbP} \mathbf{t}_{apb}[3]))$;
$\mathbf{v} = \mathbf{z}+1_{apb}$;
$\mathbf{r} = [\mathbf{v} \times_{apbP} \mathbf{t}_{apb} \mathbf{s}\ [0];\ \mathbf{v} \times_{apbP} \mathbf{s}\ [1];\ \mathbf{v} \times_{apbP} \mathbf{s} = [2];\ \mathbf{v} \times_{apbP} \mathbf{s}[3]]$;
$\mathbf{v} = [\mathbf{w}, \mathbf{w}, \mathbf{w}, \mathbf{w}] +_{4n} \mathbf{u} +_{4n} \mathbf{r}$;

$\mathbf{C} = \mathbf{C}\times_{apbP\_4nN}\mathbf{v}$;
$\mathbf{x} = \mathbf{x}_{apbP}^{(n-1)}$
$\mathbf{y} = \mathbf{y}_{apbP}^{(n-1)}$.
Вернуть **C**.

б) П.о.н.б. поля $GF(2^{191})$, о.н.б. поля $GF((2^{191})^4)$.

Алгоритм *Pairingaonbonb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{aonb}$, **b**) спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{aonb}$ при $\mathbf{b} = 1_{aonb}$:
$\mathbf{C} = [1_{aonb}, 1_{aonb}, 1_{aonb}, 1_{aonb}]$;
$\mathbf{s} = \mathbf{t}_{aonb\ aonbN\_4n}^{(1)}$;
для $i = (1, n)$
$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{aonb\ aonbN}^{(1)}$;
$\boldsymbol{\beta} = \boldsymbol{\beta}_{aonb\ aonbN}^{(1)}$;
$\mathbf{z} = \boldsymbol{\alpha} + \mathbf{x}$
$\mathbf{v} = \boldsymbol{\alpha}\times_{aonbNx}\mathbf{x}$;
$\mathbf{w} = \mathbf{z}+\mathbf{v}+\boldsymbol{\beta}+\mathbf{y}+1_{aonb}$;
$\mathbf{u}=[\mathbf{z}\times_{aonbN\_4n}\mathbf{t}_{aonb}[0]); \mathbf{z}\times_{aonbN\_4n}\mathbf{t}_{aonb}[0]);$
$\mathbf{z}\times_{aonbN\_4n}\mathbf{t}_{aonb}[0]); \mathbf{z}\times_{aonbN\_4n}\mathbf{t}_{aonb}[0]));$
$\mathbf{v} = \mathbf{z}+1_{apb}$;
$\mathbf{r} = [\mathbf{v}\times_{aonbN\_4n}\mathbf{s}\ [0];\ \mathbf{v}\times_{aonbN\_4n}\mathbf{s}\ [1];\ \mathbf{v}\times_{aonbN\_4n}\mathbf{s}\ [2];$
$\mathbf{v}\times_{aonbN\_4n}\mathbf{s}\ [3]]$;
$\mathbf{v} = [\mathbf{w}, \mathbf{w}, \mathbf{w}, \mathbf{w}]+_{4n}\mathbf{u}+_{4n}\mathbf{r}$;
$\mathbf{C} = \mathbf{C}\times_{aonbN\_4n}\mathbf{v}$;
$\mathbf{x} = \mathbf{x}_{aonbN}^{(n-1)}$;
$\mathbf{y} = \mathbf{y}_{aonbN}^{(n-1)}$.
Вернуть **C**.

в) П.п.б. поля $GF(2^{191})$, п.б. поля $GF((2^{191})^4)$.

Алгоритм *Pairingapbpb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{apb}$, **b**) спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{apb}$ при $\mathbf{b} = 1_{aonb}$:
$\mathbf{C} = [1_{apb}, 0_{apb}, 0_{apb}, 0_{apb}]$;
$\mathbf{s} = \mathbf{t}_{apb\ \ apbP\_4nNPb}^{(1)}$;
для $i = (1, n)$
$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{apbP}^{(1)}$;
$\boldsymbol{\beta} = \boldsymbol{\beta}_{apbP}^{(1)}$;
$\mathbf{z} = \boldsymbol{\alpha} + \mathbf{x}$;
$\mathbf{v} = \boldsymbol{\alpha}\times_{apbP}\mathbf{x}$;
$\mathbf{w} = \mathbf{z}+\mathbf{v}+\boldsymbol{\beta}+\mathbf{y}+1_{apb}$;
$\mathbf{u}=[\mathbf{z}\times_{apbP}\mathbf{t}_{apb}[0]); \quad \mathbf{z}\times_{apbP}\mathbf{t}_{apb}[1]); \quad \mathbf{z}\times_{apbP}\mathbf{t}_{apb}[2]);$
$\mathbf{z}\times_{apbP}\mathbf{t}_{apb}[3]));$
$\mathbf{v} = \mathbf{z}+1_{apb}$;
$\mathbf{r} = [\mathbf{v}\times_{apbP}\mathbf{t}_{apb}\mathbf{s}\ [0];\ \mathbf{v}\times_{apbP}\mathbf{s}\ [1];\ \mathbf{v}\times_{apbP}\mathbf{s} = [2];$
$\mathbf{v}\times_{apbP}\mathbf{s}[3]]$;
$\mathbf{v} = [\mathbf{w}, 0_{apbP}, 0_{apbP}, 0_{apbP}]+_{4n}\mathbf{u}+_{4n}\mathbf{r}$;
$\mathbf{C} = \mathbf{C}\times_{apbP\_44NPb}\mathbf{v}$;
$\mathbf{x} = \mathbf{x}_{apbP}^{(n-1)}$;
$\mathbf{y} = \mathbf{y}_{apbP}^{(n-1)}$.
Вернуть **C**.

г) П.о.н.б. поля $GF(2^{191})$, п.б. поля $GF((2^{191})^4)$.

Алгоритм *Pairingaonbpb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{aonb}$, **b**) спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{aonb}$ при $\mathbf{b} = 1_{aonb}$ отличается от алгоритма *Pairingaonbonb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{aonb}$, **b**) так же, как алгоритм *Pairingapbpb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{apb}$, **b**) отличается от алгоритма *Pairingapbonb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{aonb}$, **b**).

Протокол без извлечения квадратного корня [8]

а) П. б. поля $GF(2^{191})$, о.н.б. поля $GF((2^{191})^4)$.

Алгоритм *Pairingpbonb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{pb}$, **b**) спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{pb}$при $\mathbf{b}=1_{pb}$:
$\mathbf{C} = [1_{pb}, 1_{pb}, 1_{pb}, 1_{pb}]$;
$\mathbf{s} = \mathbf{t}_{pb\ \ pb\_4nN}^{(1)}$;
$\mathbf{u} = \mathbf{x}_{pbpb}^{(1)}$
$\mathbf{v} = \mathbf{u}$;
$\mathbf{y} = \mathbf{y}_{pbpb}^{(1)}$;
для $i = (1, n)$
$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{pbpb}^{(4)}$;
$\boldsymbol{\beta} = \boldsymbol{\beta}_{pbpb}^{(4)}$;
$\mathbf{w} = \mathbf{a}\times_{bp}(\mathbf{v}+1_{pb}, n)+\mathbf{u}+\mathbf{y}+\boldsymbol{\beta}+((\underline{n}-1)/2)_{pb}$;
$\mathbf{v} = \mathbf{a} + \mathbf{v}$;
$\mathbf{r} = \mathbf{u} + 1_{pb}$;
$\mathbf{a} = [\mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[0] + \mathbf{r}\times_{pb}\mathbf{s}[0];\ \mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[1] + \mathbf{r}\times_{pb}\mathbf{s}[1];$
$\mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[2] + \mathbf{r}\times_{pb}\mathbf{s}[2];\ \mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[3] + \mathbf{r}\times_{pb}\mathbf{s}[3]]$;
$\mathbf{C} = \mathbf{C}_{bp4\_n}^{(1)}\times_{pbP\_4nN}\mathbf{a}$;
$\mathbf{u} = \mathbf{u} + \mathbf{v} + 1_{pb}$;
$\mathbf{v} = \mathbf{v} + 1_{pb}$.

б) П. б. поля $GF(2^{191})$, п.б. поля GF .

Алгоритм *Pairingpbpb*(**α**, **β**, **x**, **y**, $\mathbf{t}_{pb}$, **b**) спаривания точек $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ с параметром спаривания $\mathbf{t}_{pb}$ при $\mathbf{b} = 1_{pb}$ отличается от рассмотренного алгоритма в первых двух строках $\mathbf{C} = [1_{pb}, 0_{pb}, 0_{pb}, 0_{pb}]$; $\mathbf{s} = \mathbf{t}_{pbpb\_4nP}^{(1)}$ и в строках
$\mathbf{a} = [\mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[0] + \mathbf{r}\times_{bp}\mathbf{s}[0];\ \mathbf{w} + \mathbf{v}\mathbf{x}_{bp}\mathbf{t}_{pb}[1] + \mathbf{r}\times_{bp}\mathbf{s}[1];$
$\mathbf{v}\times_{bp}\mathbf{t}_{pb}[2] + \mathbf{r}\times_{bp}\mathbf{s}[2];\ \mathbf{v}\times_{bp}\mathbf{t}_{pb}[3] + \mathbf{r}\times_{bp}\mathbf{s}[3];$
$\mathbf{C} = \mathbf{C}_{pb4\_n}^{(1)}\times_{pb\_4nP}\mathbf{a}$.

В табл. 2, 3 приведены данные о числе операций, исполняемых в ходе рассмотренных алгоритмов спаривания на суперсингулярной эллиптической кривой $Y^2=X^3+X+1$ над полем $GF(2^{191})$. Очевидно (при умножении многочленов 190 степени последовательной программой по методу Карацубы) преимущество протокола извлечением квадратного корня при использовании п.о.н.б поля $GF(2^{191})$ и п.п его расширения 4-й степени как по общему числу исполняемых элементарных операций, так и по числу исполняемых логических операций. В таблицах в скобках приведены относительные значения соответствующих количеств для этой реализации протокола. В квадратных скобках приведены расчетные данные для исполнения этих же алгоритмов при умножении наиболее быстрым из заявленных алгоритмов (см. ранее). При этом с небольшим преимуществом предпочтительным оказывается алгоритм без извлечения квадратных корней, исполняемый в п.б. основного поля и о.н.б. 1-го типа его расширения.

Об операции финального экспоненцирования на суперсингулярной эллиптической кривой

В трехстороннем протоколе согласования ключей операция спаривания сопровождается операцией финального экспоненцирования — возведением результата спаривания в степень, равную частному от деления

Таблица 2

Базис поля $GF(2^n)$, $n = 191$	**Операция**	**Число операций по алгоритму с извлечением корней, если базис поля $GF(2^n)^4$**	
		Оптимальный нормальный 1-го типа	**Полиномиальный**
Почти полиномиальный 3-го типа	&	18969456 (2, 1) [21893344 (2, 13)]	17856690 (2, 0) [20609060 (2, 0)]
	⊕	152823976 (2, 0) [53476904 (1, 98)]	143914388 (2, 0) [50395108 (1, 86)]
	{&, ⊕}	171793432 (2, 0) [75370248 (2, 02)]	161771078 (2, 0) [71004168 (1, 9)]
	Присваивания *e*	79246536 (2, 0)	74630038 (1, 9)
	{&, ⊕, *e*}	251039968 (2, 1)	236401116 (1, 9)
	Умножения в $GF(2)[X]$	3256 (2, 0)	3065(5, 0)
	Сложения в $GF^{(2n)}$	8247 (1, 0)	7852 (1, 0)
	64*e*	4632 (1, 0)	4608 (1, 0)
Переставленный нормальный 2-го типа	&	13370670 (1, 5) [15431580 (1, 5)]	8907954 (1, 0) [27068878 (1, 0)]
	⊕	109976034 (1, 5) [39950994 (1, 48)]	73721726 (1, 0) [10280996 (1, 0)]
	{&,⊕}	123346704 (1, 5) [55382574 (1, 48)]	82629680 (1, 0) [37349874 (1, 0)]
	e	57305419 (1, 5)	39179357 (1, 0)
	{&, ⊕, *e*}	180652123 (1, 5)	121809037 (1, 0)
	Умножения в $GF(2)[X]$	2295 (1, 5)	1529 (1, 0)
	Сложения в $GF^{(2n)}$	8440 (1, 0)	8236 (1, 0)
	64*e*	4632 (1, 0)	4608 (1, 0)

Таблица 3

Базис поля $GF(2^n)$, $n = 191$	**Операция**	**Число операций по алгоритму с извлечением корней, если базис поля $GF(2^n)^4$**	
		Оптимальный нормальный 1-го типа	**Полиномиальный**
Полиномиальный	&	11164532 (1, 3) [22998101 (1, 25)]	13402100 (1, 5) [98090816 (1, 5)]
	⊕	81459093 (1, 1) [12885100 (0, 8)]	96025416 (1, 3) [25847816 (0, 95)]
	{&, ⊕}	92623625 (2, 3) [35883201 (0, 96)]	109427516 (1, 3) [123938632 (1, 1)]
	Присваивания *e*	59576778 (1, 5)	69848196 (1, 8)
	{&, ⊕, *e*}	152200403 (2, 4)	17927571 (1, 5)
	Умножения в $GF(2)[X]$	1916 (2, 5)	2300 (1, 5)
	Сложения в $GF^{(2n)}$	8633 (1, 0)	9192 (1, 1)
	Блоки по 64*e*	2316 (0, 3)	1534 (0, 3)

порядка мультипликативной группы поля $GF((2^n)^4)$ на порядок эллиптической кривой. Для рассмотренной выше суперсингулярной эллиптической кривой это число равно

d=309163001841380667575628151282363358919704166954968792967160240895984012937857959440293752760129934932222666949490778779849873591807930187844368086139493033775397492815298 55.

Учитывая, что при бинарном разложении этого числа единицы занимают позиции 0—95, 97—190, 192—381, 478 и 573, получим разложение экспоненты

$$d = (((2^{95} + 1)2^{286} + (2^{190} - 1))2^{95} + (2^{94} - 1))2^{97} + (2^{96} - 1) = \\ = ((\mathbf{a}^0 2^{286} + \mathbf{a}^1)2^{95} + \mathbf{a}^2)2^{97} + \mathbf{a}^3.$$

Получаем алгоритм финального экспоненцирования

$\mathbf{x}^d = (((\mathbf{y}_0)^{2^{286}}\mathbf{y}_1)^{2^{95}}\mathbf{y}_2)^{2^{27}}\mathbf{y}_3$, где $\mathbf{y}_0 = \mathbf{x}^{a_1} = \mathbf{x}^{2^{95}}\mathbf{x}$, а остальные элементы $\mathbf{y}_1 = \mathbf{x}^{a_1}, \mathbf{y}_2 = \mathbf{x}^{a_2}, \mathbf{y}_3 = \mathbf{x}_3^a$ получаются по аддитивной цепочке 1, 2, 4, 8, 10, 14, 20, 40, 80, 94, 96, 160, 180, 190 длины 13.

Строим по ней цепочку для $\mathbf{a}_i = 2^{b_i} - 1$, $i = 1, 2, 3$(в ней шаги удвоения сгруппированы вместе):

$$1,2,2^2 -1, (2^2 - 1)2^2, (2^4 - 1)2^4, 2^8 - 1, (2^8 - 1)2^2, \\ (2^{10} - 1)2^4, 2^{14} - 1, (2^{10} - 1)2^{10}, 2^{20} - 1, (2^{20} - 1)2^{20}, \\ (2^{40} - 1)2^{40}, 2^{80} - 1, (2^{80} - 1)2^{14}, 2^{94} - 1, (2^{94} - 1)2^2, \\ 2^{96} - 1, (2^{80} - 1)2^{80}, 2^{160} - 1, (2^{160} - 1)2^{20}, 2^{180} - 1, \\ (2^{180} - 1)2^{10}, 2^{190} - 1.$$

В цепочке, если не считать удвоений, 13 сложений. По ней строится цепочка умножений для вычисления $\mathbf{y}_i = \mathbf{x}^{a_i}$ с 13-ю умножениями (остальное — возведения в степени вида 2^n). Всего же будет 17 умножений (с уче-

том сделанных). Программа финального экспоненцирования при использовании п.п.б. поля $GF(2^{191})$ и о.н.б. 1-го типа его расширения 4-й степени соответствует следующей цепочке операций:

$$\mathbf{x} = \mathbf{a};\ \mathbf{v} = \mathbf{x}_{apbP\_4N}^{(1)} = \mathbf{x}^{2};\ \mathbf{z}_1 = \mathbf{v}\times_{apbP\_4N}\mathbf{x} = \mathbf{x}^{2^2-1};$$
$$\mathbf{v} = \mathbf{z}_{apbP\_4N}^{(2)} = \mathbf{x}^{2^4-2^2};\ \mathbf{z} = \mathbf{v}\times_{apbP\_4N}\mathbf{z}_1 = \mathbf{x}^{2^4-1};$$
$$\mathbf{v} = \mathbf{z}_{apbP\_4N}^{(4)} = \mathbf{x}^{2^8-2^4};\ \mathbf{z} = \mathbf{y}\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^8-1};$$
$$\mathbf{v}_2 = \mathbf{x}_{apbP\_4N}^{(2)} = \mathbf{x}^{2^2};\ \mathbf{v} = \mathbf{z}\times_{apbP\_4N}\mathbf{v} = \mathbf{x}^{2^{10}-2^2};$$
$$\mathbf{z}_2 = \mathbf{v}\times_{apbP\_4N}\mathbf{z}_1 = \mathbf{x}^{2^{10}-1};\ \mathbf{v} = \mathbf{x}_{apbP\_4N}^{(4)} = \mathbf{x}^{2^4};$$
$$\mathbf{z} = \mathbf{v}\times_{apbP\_4N}\mathbf{z}_2 = \mathbf{x}^{2^{114}-2^4};\ \mathbf{v}_1 = \mathbf{v}\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{14}-1};$$
$$\mathbf{v}_{10} = \mathbf{x}_{apbP\_4N}^{(10)} = \mathbf{x}^{2^{10}};\ \mathbf{z}_3 = \mathbf{z}_2\times_{apbP\_4N}\mathbf{z}_2 = \mathbf{x}^{2^{20}-2^{10}};$$
$$\mathbf{z}_4 = \mathbf{z}_2\times_{apbP\_4N}\mathbf{z}_2 = \mathbf{x}^{2^{20}-1};\ \mathbf{v}_3 = \mathbf{x}_{apbP\_4N}^{(20)}\mathbf{v}_3 = \mathbf{x}^{2^{20}};$$
$$\mathbf{z} = \mathbf{v}_3\times_{apbP\_4N}\mathbf{z}_3 = \mathbf{x}^{2^{40}-2^{20}}I;\ \mathbf{z}_5 = \mathbf{v}\times_{apbP\_4N}\mathbf{z}_4 = \mathbf{x}^{2^{40}-1};$$
$$\mathbf{x}_{apbP\_4N}^{(40)} = \mathbf{x}^{2^{40}};\ \mathbf{z} = \mathbf{v}\times_{apbP\_4N}\mathbf{z}_5 = \mathbf{x}^{2^{80}-2^{40}};$$
$$\mathbf{z}_6 = \mathbf{z}_5\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{80}-1};\ \mathbf{z} = \mathbf{v}_2\times_{apbP\_4N}\mathbf{y}_2 = \mathbf{x}^{2^{96}-2^2};$$
$$\mathbf{y}_3 = \mathbf{z}_1\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{96}-1};\ \mathbf{v} = \mathbf{x}_{apbP\_4N}^{(80)} = \mathbf{x}^{2^{80}};$$
$$\mathbf{z} = \mathbf{z}_6\times\mathbf{v} = \mathbf{x}^{2^{160}-2^{80}};\ \mathbf{z} = \mathbf{v}_6\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{160}-1};$$
$$\mathbf{z} = \mathbf{v}_3\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{180}-2^{20}};\ \mathbf{z} = \mathbf{v}_3\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{180}-2^{20}};$$
$$\mathbf{z} = \mathbf{v}_3\times_{apbP\_4N}\mathbf{z}_4 = \mathbf{x}^{2^{180}-1};\ \mathbf{z} = \mathbf{v}_{10}\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{190}-2^{10}};$$
$$\mathbf{y}_1 = \mathbf{z}_2\times_{apbP\_4N}\mathbf{z} = \mathbf{x}^{2^{190}-1};\ \mathbf{z} = \mathbf{x}_{apbP\_4N}^{(95)} = \mathbf{x}^{2^{95}};$$
$$\mathbf{y}_0 = \mathbf{z}\times_{apbP\_4N}\mathbf{x} = \mathbf{x}^{2^{95}+1};\ \mathbf{z} = \mathbf{y}_{0\ apbP\_4N}^{(286)} = \left(\mathbf{x}^{2^{95}+1}\right)^{286};$$
$$\mathbf{z} = \mathbf{z}\times_{apbP\_4N}\mathbf{y}_1\left(\mathbf{x}^{2^{95}+1}\right)\mathbf{x}^{2^{190}-1}.$$

Спаривание и финальное экспоненцирование

В трехстороннем протоколе согласования ключей эти две операции выполняются совместно. В табл. 4 представлены данные о числе операций при различных выборах базисов. При использовании о.н.б. 1-го типа вместо п.б. при финальном экспоненцировании значения 90809312 [86278370] заменятся на 90793760 [86262818].

Как видим, во всех случаях, в том числе и при использовании быстрого умножения в кольце, наилучшим является использование п.о.н.б. основного поля и п.б. его расширения, т. е. незначительное «отставание» спаривания в п.о.н. б. основного поля и п.б. расширения от варианта п.б. основного поля и о.н.б. расширения компенсируется существенным преимуществом финального экспоненцирования при первом сочетании базисов по сравнению со вторым.

Заключение

В настоящей работе на основе компьютерных экспериментов и анализа алгоритмов показано преимущество использования сочетаний различных базисов конечных полей в алгоритмах спаривания и финального экспоненцирования на суперсингулярной эллиптической кривой при криптографически значимой степени расширения бинарного поля. Показано, что при наличии о.н.б. 2-го или 3-го типа алгоритм спаривания извлечением квадратного корня предпочтительнее алгоритма без этой операци. При использовании последовательной программы умножения по методу Карацубы наиболее эффективно он реализуется в п.о.н.б. поля $GF(2^{191})$ и п.б. его расширения. При использовании рекордной по числу исполняемых операций программы умножения лучшим с незначительным преимуществом оказывается сочетание п.б. основного поля и о.н.б. его расширения. Однако существенное преимущество финального экспоненцирования в п.о.н.б. основного поля и о.н.б. 1-го типа его расширения влечет преимущество использования этого базиса основного поля как при спаривании, так и при финальном экспоненцировании, а для эффективной реализации следующего после операции спаривания операции финального экспоненцирования необходимо преобразование из п.б. в о.н.б. расширения поля, что реализуется просто при использовании общего для п.б. и о.н.б. минимального многочлена.

Литература

1. **Болотов А.А., Гашков С.Б.** О быстром умножении в нормальных базисах конечных полей //Дискретная математика. 2001. Т. 13. Вып. 3. С. 3—31.

2. **Mullin R.C., Onyszchuk I.M., Vanstone S.A., Wilson R.M.** Optimal Normal Bases in GF(pn) // Discrete Appl. Math. 1988/89. Vol. 22. P, 149—161.

3. **Shokrollahi J.** Efficient Implementation Of Elliptic Curve Cryptography on FPGA. PhD Thesis. Universitet Bonn, 2007.

4. **Gathen von zur J., Shokrollahi A., Shokrollahi J.** Efficient Multiplication Using Type 2 Optimal Normal Bases // In WAIFI ’07, LNCS. 2007. P. 55—68.

Таблица 4

Базис поля $GF^{(2n)}$, n=191	**Число операций при выборе базиса поля $GF^{(2n)4}$:**	
	О.н.б. 1-го типа	**П.б.**
По алгоритму с извлечением корней и финальному экспоненцированию		
П.п.б.	179841574 [175310632]	165516772 [160985830]
П.о.н.б	131510784 [126979842]	90809312 [86278370]
По алгоритму без извлечения корней и финальному экспоненцированию		
П.б.	103806253 [99275311]	120975456 [116444514]

5. **Bernstein D.J., Lange T.** Type-II Optimal Polynomial Bases. Arithmetic of Finite Fields. Proceedings. LNCS. 6087. 2010. P. 41—61.

6. **Duursma I., Lee H.** Tate Pairing Implementation For Hyperelliptic Curves $y^2 = xp - x + d$ // Lecture Notes in Computer Science. 2003. Vol. 2894. P. 111—123.

7. **Kwon S.** Efficient Tate Pairing Computation For Supersingular Elliptic Curves Jver Binary Fields. Cryptology ePrint archive. Report 2004/303. 2004.

8 **Joux A.** A One Round Protocol For Tripartite Di-Hellman // Lecture Notes in Computer Science. ANTS. 2000. Vol. 1838. P. 385—394.

9. **Sroeppel R, Orman H., Malley S'O., Spatschek O.** Fast Key Exchange With Elliptic Curve Systems // Lecture Notes In Computer Science. CRYPTO95. № 965. P. 43—56.

10. **Bernstein D.J.** Minimum Number Of Bit Operations For Multiplication. URL: http://binary.cr.yp.to/m.html (access date: 2009).

References

1. **Bolotov A.A., Gashkov S.B.** O Bystrom Umnozhenii v Normalnych Bazisach Konechnych Poley. Diskretnaya matenetika;13(3):3–31.

2. **Mullin R.C., Onyszchuk I.M., Vanstone S.A., Wilson R.M.** Optimal Normal Bases in GF(pn). Discrete Appl. Math. 1988/89;22:149—161.

3. **Shokrollahi J.** Efficient Implementation Of Elliptic Curve Cryptography On Fpga. PhD Thesis. Universitet Bonn, 2007.

4. **Gathen von zur J., Shokrollahi A., Shokrollahi J.** Efficient Multiplication Using Type 2 Optimal Normal Bases. In WAIFI '07, LNCS. 2007:55—68.

5. **Bernstein D.J., Lange T.** Type-II Optimal Polynomial Bases. Arithmetic of Finite Fields. Proceedings. LNCS. 6087. 2010:41—61.

6. **Duursma I., Lee H.** Tate Pairing Implementation For Hyperelliptic Curves $Y^2 = Xp - X + D$. Lecture Notes In Computer Science. 2003.;2894:111—123.

7. **Kwon S.** Efficient Tate Pairing Computation for Supersingular Elliptic Curves Jver Binary Fields. Cryptology ePrint archive. Report 2004/303. 2004.

8 **Joux A.** A One Round Protocol for Tripartite Di-Hellman: Lecture Notes in Computer Science. ANTS. 2000;1838:385—394.

9. **Sroeppel R., Orman H., Malley S'.O., Spatschek O.** Fast Key Exchange With Elliptic Curve Systems. Lecture Notes in Computer Science. CRYPTO95. 965.:43—56.

10. **Bernstein D.J.** Minimum Number of Bit Operations for Multiplication. URL: http://binary.cr.yp.to/m.html (Access Date: 2009).

Статья поступила в редакцию 18.11.2016