УДК 519.21 **Ковтун В.Ю.**

ФОРМУЛА СЛОЖЕНИЯ ДИВИЗОРОВ С ИДЕНТИЧНЫМИ Z-КООРДИНАТАМИ В ЯКОБИАНЕ ГИПЕРЭЛЛИПТИЧЕСКОЙ КРИВОЙ ВТОРОГО РОДА НАД ПРОСТЫМИ ПОЛЯМИ

Введение

Началом применения алгебраических кривых в криптографии послужили публикации Миллера и Коблица [1, 2]. Ими было предложено воспользоваться свойством точек эллиптической кривой (ЭК), образовывать аддитивную Абелевую группу. В своей более поздней работе [3] Коблиц обосновал возможность использования более сложных кривых – гиперэллиптических (ГЭК). Для ГЭК следует рассматривать группу (якобиан) уже не самих точек кривой, а более сложных структур – дивизоров. Как отмечено в [3], ГЭК, в свою очередь, обладают целым рядом преимуществ над ЭК: являются более богатым источником Абелевых групп [3, 4] (образуют Абелевую группу, размер которой определяется произведением размера базового поля на род кривой).

Долгое время криптографические системы на ГЭК представляли сугубо академический интерес, и не имели практического применения, из-за высокой сложности программной и аппаратной реализации, низкого быстродействия, отсутствия глубоких исследований в области криптоанализа таких криптосистем и отсутствия приемлемых алгоритмов формирования общесистемных параметров [3, 4]. Активные исследования [3-18] ГЭК позволили преодолеть большинство описанных трудностей. Данная работа посвящена вопросам дальнейшего повышения производительности криптосистем на ГЭК второго рода над простыми полями.

Авторами публикаций [7, 9-18], предложен целый ряд подходов, существенно повышающих производительность криптосистем на ГЭК и делают их реальными конкурентами криптосистемам на ЭК.

Дадим базовые понятия о криптосистемах на ГЭК. Более детальную информацию можно почерпнуть из работ [3, 4].

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

$$C: v^2 + h(u)v = f(u) \quad k[u, v] \qquad (1)$$

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

В случае ГЭК рода 2, полиномы $f(u)$ и $h(u)$ будут иметь вид: $f(u) = u^5 + f_4u^4 + f_3u^3 + f_2u^2 + f_1u + f_0$, $h(u) = h_2u^2 + h_1u + h_0$, $h_i, f_j \in K$.

Дивизор D - формальная сумма точек кривой C:

$$D = \sum_{P \in C} m_P P \quad m_P \in \mathbf{Z}$$

где только конечное число коэффициентов m_P отлично от нуля.

Дивизор $D \in \mathbf{D}^0$ - *основной дивизор*, если $D = \mathrm{div}(R)$ для некоторой рациональной функции $R \in \overline{K}(C)^*$. Множество основных дивизоров кривой C над $\overline{K}$ обозначают $P_C(\overline{K}) = \{\mathrm{div}(F) : F \in \overline{K}(C)\}$, причем $P_C(\overline{K})$ подгруппа $\mathbf{D}^0$. Обычно $P(C) = P_C(\overline{K})$ называют *группу основных дивизоров кривой* C. Частная группа $J_C(\overline{K}) = \mathrm{div}_C^0(\overline{K}) / P_C(\overline{K})$ называется *якобианом кривой* C над $\overline{K}$. Частная группа $J(C) = \mathrm{div}^0(C) / P(C)$ называется *якобианом кривой* C.

В дальнейшем, будем оперировать дивизорами в форме Мамфорда [4] $D=\left(x^2+u_1x+u_0, v_1x+v_0\right)$, $\deg v<\deg u\le 2$, $u\mid f(u)-h(u)v-v^2$, причем $\forall D_i\in J(C)$, $weight(D_i)=2$, $i=\overline{1,2}$ результат $D_3=D_1+D_2$ будет $weight(D_3)=2$, что позволит избавиться от рассмотрения множества дополнительных проверок. В качестве базового поля возьмем $\mathbf{GF}(p)$, где p - нечетное простое.

Криптосистемы на ГЭК используют операцию - скалярного умножения дивизора на скаляр:

$$\underbrace{D+D+\cdots+D}_{k}=k\cdot D.$$

На этапе промежуточных вычислений, операции скалярного умножения, наиболее распространенным двоичным алгоритмом, выполняется сложение и удвоение дивизоров. Алгоритмы сложения и удвоения дивизоров используют дорогостоящую, с точки зрения вычислений, операцию мультипликативного инвертирования в $\mathbf{GF}(p)$. Одним из подходов, позволяющим избавиться от инвертирования, является переход к проективному представлению дивизоров [11, 12].

В проективном виде дивизор $D=\left(x^2+u_1x+u_0, v_1x+v_0\right)$ можно представить $D=\left[U_1, U_0, V_1, V_0, Z\right]$, причем $D=\left(x^2+U_1/Z\,x+U_0/Z, V_1/Z\,x+V_0/Z\right)$.

На текущий момент, наиболее производительными являются арифметические преобразования в проективном [20] и взвешенном представлении [17].

Co-Z подход

Новым толчком к повышению эффективности операции скалярного умножения точек ЭК над $\mathbf{GF}(p)$, послужила работа [22], в которой автор предлагает привести точки ЭК в проективном и модифицированном проективном представлении Якоби, к единому знаменателю и в дальнейшем оперировать точками с идентичными Z-координатами. Такой подход в литературе получил название Co-Z. Для реализации скалярного умножения на основе идеи [22], следует использовать алгоритмы, описанные в [22], оперирующие скалярным множителем в виде последовательности Эвклида или Фибоначчи [22] в представлении Зекендорфа, и не используют операцию удвоения точек, см. алгоритм А.1.

Применим Co-Z подход к алгоритму сложения дивизоров в проективном представлении. За основу следует взять алгоритмы сложения дивизоров, предложенные в [15] и усовершенствованные в [20] (алгоритм А.2).

Алгоритм А.1 Fibonacci-and-add (k, P)

Вход: $D\in J_C$, $k=(d_l,\ldots,d_2)_Z$

Выход: $[k]D\in J_C$

```
begin
    (U, V) ← (D, D)
    for i = l − 1 downto 2
        if d_i = 1 then U ← U + D (шаг сложения)
        (U, V) ← (U + V, U) (шаг Фибоначчи)
    end
    return U
end
```

Исходя из предположения, что $Z_1=Z_2=Z$, для D_1 и D_2, алгоритм А.2 может быть преобразован к алгоритму А.3. Остановимся на произведенных модификациях в А.2.

Алгоритм А.2. Сложение приведенных дивизоров	
Вход:	$\left[U_{11}, U_{10}, V_{11}, V_{10}, Z_1\right]$, $\left[U_{21}, U_{20}, V_{21}, V_{20}, Z_2\right]$

Выход:	$[U_1', U_0', V_1', V_2', Z'] = [U_{11}, U_{10}, V_{11}, V_{10}, Z_1] + [U_{21}, U_{20}, V_{21}, V_{20}, Z_2]$, $weight(D_1) = weight(D_2) = 2$	
#	**Выражение**	**Сложность**
1	Предвычисления: $Z = Z_1 \cdot Z_2$, $\tilde{U}_{21} = Z_1 \cdot U_{21}$, $\tilde{U}_{20} = Z_1 \cdot U_{20}$, $\tilde{V}_{21} = Z_1 \cdot V_{21}$, $\tilde{V}_{20} = Z_1 \cdot V_{20}$	5M
2	Вычисление результатнты r для u_1 и u_2: $y_1 = U_{11} \cdot Z_2 - \tilde{U}_{21}$, $y_2 = \tilde{U}_{20} - U_{10} \cdot Z_2$, $y_3 = U_{11} \cdot y_1 + y_2 \cdot Z_1$, $r = y_2 \cdot y_3 + y_1^2 \cdot U_{10}$	1S, 6M
3	Вычисление почти-инверсии $inv = r/u_2 \bmod u_1$, $inv = inv_1 x + inv_0$: $inv_1 = y_1$, $inv_0 = y_3$	
4	Вычисление $s = (v_1 - v_2)\, inv \bmod u_1$, $s = s_1 x + s_0$: $w_0 = V_{10} \cdot Z_2 - \tilde{V}_{20}$, $w_1 = V_{11} \cdot Z_2 - \tilde{V}_{21}$, $w_2 = inv_0 \cdot w_0$, $w_3 = inv_1 \cdot w_1$, $s_0 = w_2 - U_{10} \cdot w_3$, $s_1 = (inv_0 + Z_1 \cdot inv_1) \cdot (w_0 + w_1) - w_2 - w_3 \cdot (Z_1 + U_{11})$ Если $s_1 = 0$ тогда следует рассмотреть особый случай	8M
5	Предвычисления: $R = r \cdot Z$, $s_2 = s_0 \cdot Z$, $s_3 = s_1 \cdot Z$, $\tilde{R} = R \cdot s_3$, $w_0 = s_1 \cdot s_0$, $w_1 = s_1 \cdot s_3$, $w_2 = s_0 \cdot s_3$, $w_3 = w_1 \cdot \tilde{U}_{21}$, $w_4 = R \cdot s_1$	9M
6	Вычисление $l = su_2$, $l = x^3 + l_2 x^2 + l_1 x + l_0$: $l_0 = w_0 \cdot \tilde{U}_{20}$, $l_2 = w_3 + w_2$, $l_1 = (w_1 + w_0) \cdot (\tilde{U}_{21} + \tilde{U}_{20}) - l_0 - w_3$	2M
7	Вычисление $u' = (s(l + h + 2v_1) - k) u_1^{-1}$, $k = (f - v_1 h - v_1^2)/u_1$, $u' = x^2 + u_1' x + u_0'$: $\tilde{U}_1' = 2w_2 - s_3 \cdot s_1 y_1 + h_2 \tilde{R} - R^2$, $\tilde{U}_0' = s_2^2 + s_1 \cdot y_1 \cdot (s_1 \cdot U_{11} - 2s_2) + y_2 \cdot w_1 + 2w_4 \cdot \tilde{V}_{21} + h_1 \tilde{R} + + R \cdot [h_2 (s_2 - s_1 U_{11}) + r \cdot (y_1 + 2\tilde{U}_{21} - f_4 Z)]$	2S, 8M
8	Коррекция: $U_0' = \tilde{U}_0' \cdot \tilde{R}$, $U_1' = \tilde{U}_1' \cdot \tilde{R}$, $Z' = s_3^2 \cdot \tilde{R}$	1S, 3M
9	Вычисление $v' \equiv -(h + s_1 l + v_2) \bmod u'$, $v' = v_1' x + v_0'$: $V_1' = \tilde{U}_1' \cdot (l_2 - \tilde{U}_1' + h_2 \tilde{R}) + s_3^2 \cdot (\tilde{U}_0' - h_0 \tilde{R} - w_4 \tilde{V}_{21} - l_1)$, $V_0' = \tilde{U}_0' \cdot (l_2 - \tilde{U}_1' + h_2 \tilde{R}) - s_3^2 \cdot (l_0 + h_2 \tilde{R} + w_4 \cdot \tilde{V}_{20})$	5M
		4S, 46M

Шаг А.2.1, может быть опущен, т.к. все координаты обладают идентичным знаменателем, что позволяет уйти от 5 умножений в $\mathbf{GF}(p)$.

Шаг А.2.2, при вычислении y_1 и y_2, нет необходимости в приведении к единому знаменателю координаты U_{1j} и U_{2j}, экономиться 2 умножения в $\mathbf{GF}(p)$.

Шаг А.2.4. При вычислении w_1 и w_2, также нет необходимости в приведении к одному знаменателю координат V_{1j} и V_{2j}, что также позволяет уйти от 2 умножений в $\mathbf{GF}(p)$.

Шаг А.2.5. Количество предвычислений, уменьшается на 3 умножения в $\mathbf{GF}(p)$ за счет изменения порядка вычисления коэффициентов l_0, l_1 и l_2 многочлена l на шаге А.2.6.

Алгоритм А.3. Сложение приведенных дивизоров Co-Z методом	
Вход:	$[U_{11}, U_{10}, V_{11}, V_{10}, Z]$, $[U_{21}, U_{20}, V_{21}, V_{20}, Z]$
Выход:	$[U_1', U_0', V_1', V_2', Z'] = [U_{11}, U_{10}, V_{11}, V_{10}, Z] + [U_{21}, U_{20}, V_{21}, V_{20}, Z]$, $weight(D_1) = weight(D_2) = 2$

#	Выражение	Сложность
1	Вычисление результанты r для u_1, u_2: $y_1 = U_{11} - U_{21}$, $y_2 = U_{20} - U_{10}$, $y_3 = U_{11} \cdot y_1 + y_2 \cdot Z$, $r = y_2 \cdot y_3 + y_1^2 \cdot U_{10}$	1S, 4M
2	Вычисление почтиз-инверсии $inv = r/u_2 \bmod u_1$, $inv = inv_1 x + inv_0$: $inv_1 = y_1$, $inv_0 = y_3$	
3	Вычисление $s = (v_1 - v_2)\, inv \bmod u_1$, $s = s_1 x + s_0$: $w_0 = V_{10} - V_{20}$, $w_1 = V_{11} - V_{21}$, $s_0 = y_3 \cdot w_0 - U_{10} \cdot y_1 \cdot w_1$ $s_1 = y_2 Z \cdot w_1 + y_1 \cdot Z \cdot w_0$, Если $s_1 = 0$ тогда следует рассмотреть особый случай	6M
4	Предвычисления: $R = r \cdot Z$, $s_2 = s_0 \cdot Z$, $s_3 = s_1 \cdot Z$, $\tilde{R} = R \cdot s_3$, $w_3 = s_1 \cdot y_1$, $w_5 = w_3 + s_1 \cdot U_{21}$	6M
5	Вычисление $l = su_2$, $l = x^3 + l_2 x^2 + l_1 x + l_0$: $l_0 = s_0 \cdot U_{20}$, $l_2 = s_1 U_{21}$, $l_1 = (w_1 + w_0) \cdot (U_{21} + U_{20}) - l_0 - l_2$, $l_2 = l_2 + s_0$	2M
6	Вычисление $u' = \left(s(l + h + 2v_1) - k\right) u_1^{-1}$, $k = \left(f - v_1 h - v_1^2\right)/u_1$, $u' = x^2 + u_1' x + u_0'$: $\tilde{U}_1' = s_3 \cdot (2s_2 - w_3 + h_2 R) - R^2$, $\tilde{U}_0' = s_2^2 + w_3 \cdot (w_5 - 2s_2) + s_3 \cdot (y_2 \cdot s_1 + 2r \cdot V_{21} + h_1 R) +$ $+ R \cdot \left[h_2 (s_2 - w_5) + r \cdot (U_{11} + U_{21} - f_4 \cdot Z) \right]$	2S, 8M
7	Коррекция: $U_0' = \tilde{U}_0' \cdot \tilde{R}$, $U_1' = \tilde{U}_1' \cdot \tilde{R}$, $Z' = s_3^2 \cdot \tilde{R} \left(= s_3^3 \cdot r \cdot Z\right)$	1S, 3M
8	Вычисление $v' \equiv -(h + s_1 l + v_2) \bmod u'$, $v' = v_1' x + v_0'$: $V_1' = \tilde{U}_1' \cdot \left((l_2 + h_2 R) \cdot s_3 - \tilde{U}_1' \right) + s_3^2 \cdot \left(\tilde{U}_0' - s_3 \cdot (h_1 R + r V_{21} + l_1) \right)$, $V_0' = \tilde{U}_0' \cdot \left((l_2 + h_2 R) s_3 - \tilde{U}_1' \right) - s_3^2 \cdot s_3 \cdot (l_0 + h_0 R + r \cdot V_{20})$	8M
		4S, 37M
9	Коррекция: $Z_c = s_3^3 \cdot r$, $U_{20} = U_{20} \cdot Z_c$, $U_{21} = U_{21} \cdot Z_c$, $V_{20} = V_{20} \cdot Z_c$, $V_{21} = V_{21} \cdot Z_c$	5M
		4S, 42M

Шаг А.2.6. В отличие от алгоритма А.2, в А.3. предлагается учесть множитель s_3, который присутствует в каждом коэффициенте l_0, l_1 и l_2 многочлена l, непосредственно при использовании коэффициентов l_i, $i = \overline{0,2}$ на шагах А.2.7 и А.2.9. Это позволит вынести общий множитель s_3 за скобки, и таким образом экономить 3 умножения в $\mathbf{GF}(p)$. Далее рассмотрим применение предложенного алгоритма для смешанного сложения дивизоров D_1 и D_2, таких что $[U_{11}, U_{10}, V_{11}, V_{10}, Z]$ и $[U_{21}, U_{20}, V_{21}, V_{20}, 1]$ (аффинное представление). Для этого необходимо привести D_2 к единой Z-координате, т.е. $[U_{21} \cdot Z, U_{20} \cdot Z, V_{21} \cdot Z, V_{20} \cdot Z, Z]$, что потребует 4 умножения в $\mathbf{GF}(p)$. После чего следует использовать предложенный алгоритм А.3 для сложения подготовленных дивизоров.

Отметим, что кроме вычисления суммы дивизоров $D_3 = D_1 + D_2$, следует привести одно из слагаемых, например дивизор D_2, к такому виду, чтобы D_2 и D_3 обладали идентичными Z-координатами. Для этого, на шаге А.3.9, следует вычислить величину Z_c, такую, что $Z' = Z \cdot Z_c$, т.е. $Z_c = s_3^3 \cdot r$, это дополнительно потребует 1 умножение в $\mathbf{GF}(p)$. Другими словами, приведение дивизора к единой Z-координате, потребует 5 дополнительных умножений в $\mathbf{GF}(p)$, итого, для выполнения шага сложения дивизоров необходимо 42M+4S

(M - умножение, S - возведение в квадрат) операций и для шага Fibonacci необходимо 46M+4S операций. Согласно оценкам сложности [20], описанный в данной работе подход не является эффективным, т.к. сложность смешанного сложения составляет 39M+4S. Однако можно говорить об альтернативном подходе к сложению дивизоров для реализации скалярного умножения. Проведем сравнение вычислительной сложности алгоритмов скалярного умножения, рассмотренных в [20] и предложенных в данной работе на основе изложенных в [22].

Сравнение с другими методами

Сделаем предположение, что S=0,8M и скалярный множитель является 80-битным целым. За основу оценок сложности алгоритмов скалярного умножения возьмем оценки [20, 22]. Результаты сравнения приведем в таблице 1.

Таблица 1. Сложность операций скалярного умножения

#	Алгоритм скалярного умножения	Сложность, M
	Сложение в смешанных координатах [20]	
1	Двоичный (left-to-right)	5192
2	NAF	4630
3	w-NAF, $w=4$	4350
	Сложение предложенным Co-Z методом	
4	Fibonacci-and-add	6773
5	Window Fibonacci-and-add	5970

Вычислительная сложность алгоритма скалярного умножения Fibonacci-and-add обладает большей вычислительной сложностью, чем двоичный алгоритм, на 23% и алгоритмом Window Fibonacci-and-add на 13%.

Выводы

В работе предложен новый алгоритм сложения дивизоров веса, с идентичными Z-координатами, который требует большее количество операций в $\mathbf{GF}(p)$, чем алгоритм [20], однако позволит уменьшить вычислительную сложность алгоритмов скалярного умножения на основе Фибоначчи представления скалярного множителя, и вплотную приблизить ее к сложности двоичного алгоритма.

Список литературы

1. N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48(177), 1987, pp. 203–209.
2. I. V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, Advances in Cryptology - CRYPTO'85, volume 218 of LNCS, Springer, 1985, pp. 417–426.
3. N. Koblitz. Hyperelliptic cryptosystems. Journal of cryptology, No 1. 1989. pp.139-150.
4. Menezes A., Wu Y.H., Zuccherato R. An elementary introduction to hyperelliptic curves / In: Koblitz N. ed. // Algebraic aspects of cryptography. –Berlin, Heidelberg, New York: Springer–Verlag, 1998. -pp. 28–63.
5. Cantor D.G. Computing in Jacobian of a Hyperelliptic curve // Mathematics of Computation. –Vol. 48. -No.177. -1987. -pp.95–101.
6. Gaudry P., Harley R. Counting points on hyperelliptic curves over finite fields / In W. Bosma, ed. // ANTS IV. –LNCS 1838. –Berlin: Springer–Verlag, 2000. –pp.297–312.
7. 140. Harley R. Fast arithmetic on genius two curves. –2000. Available at: http://cristal.inria.fr/harley/hyper/, adding.txt and doubling.c
8. Jacobson M. (Jr), Menezes A., Stain A. Hyperelliptic curves and cryptography // Report of Fields Institute Communications. –Vol.7. –2002. –28p.
9. Lange T. Efficient arithmetic on hyperelliptic curves: PhD thesis: Mathematics and Informatics. –University of Essen: Institute for experimental mathematics. – Germany: Essen, 2001. –122p.
10. 143. Matsuo K., Chao J., Tsujii S. Fast genius two hyperelliptic curve cryptosystem // Technical report IEICE. –ISEC2001–31. –IEICE`2001. –2001. –8p.
11. Miyamoto Y., Doi H., Matsuo K., Chao J., Tsujii S. A fast addition algorithm of genius two hyperelliptic curve // In the 2002 Symposium on cryptography and information security. –SCIS`2002. Japan: IEICE, 2002. –pp.497–502. (In Japanese)
12. Takahashi M. Improving Harley algorithms for jacobians of genius 2 hyperelliptic curves // In the 2002 Symposium on cryptography and information security. –SCIS`2002. Japan: IEICE, 2002. –pp.155–160. (In Japanese)
13. Sugizaki H., Matsuo K., Chao J., Tsujii S. An extension of Harley addition algorithm for hyperelliptic curves over finite fields of characteristic two // Technical report IEICE. –ISEC2002–09. –IEICE`2002. –2002. –8p.
14. Lange T. Efficient arithmetic on genius 2 hyperelliptic curves over finite fields via explicit formulae // Cryptology ePrint Archive. –Report 2002/121. –2002. –13p. Available http://eprint.iacr.org.
15. Lange T. Inversion–free arithmetic on genius 2 hyperelliptic curves // Cryptology ePrint Archive. –Report 2002/147. –2002. –7p. Available http://eprint.iacr.org.
16. Lange T. Formulae for arithmetic on genius 2 hyperelliptic curves. September 2003. Available http://www.ruhr–uni–bochum.de/itsc/tanja/preprints/expl_sub.pdf.
17. Lange T. Weighted coordinates on genius 2 hyperelliptic curves // Cryptology ePrint Archive. –Report 2002/153. –2002. –20p. Available http://eprint.iacr.org.
18. Wollinger T. Software and hardware implementation of hyperelliptic curve cryptosystems: PhD dissertation: Electronics and informatics. –Worchester Polytechnic Institute. –Germany: Bochum, 2004. –218p.
19. Chudnovsky D.V., Chudnovsky G.V. Sequence of number generated by addition in formal group and new primality and factorization test // Advanced in Applied Math. –№8. –1986. –pp.385–434.
20. Ковтун В.Ю., Збитнев С.И. Арифметические операции в якобиане гиперэллиптической кривой рода 2 в проективных координатах с уменьшенной сложностью // Восточно–Европейский журнал передовых технологий. –2004. –Вып. №½ (13). –С. 14–22.

21. Cohen H., Miyaji A., Ono T. Efficient elliptic curve exponentiation using mixed coordinates // Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology. –CRYPTO`98. –LNCS 1514. –Berlin: Springer–Verlag, 1998. –pp. 51–65.

22. N. Meloni. New point addition formul. for ECC applications. In C. Carlet and B. Sunar, editors, Arithmetic of Finite Fields (WAIFI 2007), LNCS 4547, Springer, 2007, pp. 189–201.

Рецензент: д.т.н., проф. Прокопенко І.Г.

Надійшла 26.05.2010 р.