УДК 681.3.06

Ю.В. СТАСЕВ, д-р техн. наук, ***В.Ю. КОВТУН,***
А.А. КУЗНЕЦОВ, канд. техн. наук. ***В.Ю. ТРУБАЧОВ***

ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЛОЖЕНИЯ ДИВИЗОРОВ В ЯКОБИАНЕ ГИПЕРЭЛЛИПТИЧЕСКОЙ КРИВОЙ НАД ПОЛЯМИ ХАРАКТЕРИСТИКИ 2

Розглядаються перспективний напрямок несиметричних криптографічних перетворень у якобіані гіпереліптичної кривої. Приводяться алгоритми, що застосовуються для реалізації криптосистем. Порівнюється стійкість та складність перетворень на еліптичній та гіпереліптичній кривій.

Введение. В начале XXI века во многих государствах были приняты стандарты цифровой подписи [1-9], в число которых вошла и Украина [10]. В основе национального стандарта лежит задача дискретного логарифма в группе точек эллиптической кривой (ЭК). Интенсивное развитие вычислительной техники, значительные успехи в разработке новых методов решения задачи дискретного логарифмирования, приводит к необходимости подтверждения стойкости стандарта с указанием срока безопасной эксплуатации.

В настоящее время ведутся интенсивные исследования по определению «преемницы» задачи дискретного логарифма в группе точек эллиптической кривой (ЭК). Наиболее перспективной можно назвать задачу дискретного логарифмирования в якобиане дивизоров гиперэллиптической кривой (ЗДЛГЭК) [11]. Основным преимуществом применения гиперэллиптической кривой (ГЭК), перед ЭК, является то, что первая является более богатым источником конечных абелевых групп.

Целью данной работы является исследование эффективности криптопреобразований ГЭК (КПГЭК), по соотношению сложность реализации/производительность, а также поиск путей повышения производительности КПГЭК.

Общие положения теории ГЭК. Исследованию ЗДЛ в алгебраических многообразиях, посвящена работа [12]. Основополагающими в области алгебраических кривых являются работы [13-15]. Впервые была приведена в 1986 году Нилом Коблицем криптосистема на основе проблемы дискретного логарифма в якобиане ГЭК [11]. Общие теоретические положения приведены в [11, 12].

Пусть K - поле и пусть $\overline{K}$ - алгебраическое замыкание для k. *Гиперэллиптическая кривая C рода g над K* $(g \ge 1)$ имеет уравнение вида

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

где $h(u) \in K[u]$ - полином степени g, $f(u) \in K[v]$ - монический полином степени $2g+1$. Причем не существует таких решений $(u, v) \in \overline{K} \times \overline{K}$, которые одновременно удовлетворяют и (1), и частным производным по соответствующим переменным: $2u + h(u) = 0$, $h'(u)v - f'(u) = 0$.

Алгоритм сложения дивизоров. Для задания группового закона на множестве почти приведенных дивизоров порядка ноль ГЭК, необходимо определить операцию сложения почти приведенных дивизоров. Алгоритм сложения состоит из двух шагов, первый – композиция почти приведенных дивизоров и второй шаг – приведение почти приведенного дивизора.

Даны $D_1 = \mathrm{div}(a_1, b_1)$ и $D_2 = \mathrm{div}(a_2, b_2)$ - два приведенных дивизора над K, в форме Мамфорда, $a_1, a_2, b_1, b_2 \in K[u]$. На шаге композиции находится почти приведенный дивизор $D = \mathrm{div}(a, b)$ с $a, b \in K[u]$, таких, что $D \sim D_1 + D_2$. На втором шаге осуществляется приведение D к эквивалентному приведенному дивизору D'.

Шаги 1 и 2 были представлены Коблицем [11, 16] и являются обобщением алгоритма описанного Кантором, с ограничениями $h(u) = 0$ и $\mathrm{char}(K) \ne 2$.

Алгоритм 1. Композиция дивизоров

Вход: Приведенный дивизор $D_1 = \mathrm{div}(a_1, b_1)$ и $D_2 = \mathrm{div}(a_2, b_2)$ над K.

Выход: Почти приведенный дивизор $D = \mathrm{div}(a, b)$ определенный над K, такой, что $D \sim D_1 + D_2$.

1. Используется расширенный алгоритм Эвклида для нахождения полиномов $d_1, e_1, e_2 \in K[u]$, где $d_1 = \gcd(a_1, a_2)$, $d_1 = e_1 a_1 + e_2 a_2$.

2. Используется расширенный алгоритм Эвклида для нахождения полиномов $d, c_1, c_2 \in K[u]$, где $d = \gcd(d_1, b_1 + b_2 + h)$, $d = c_1 d_1 + c_2(b_1 + b_2 + h)$.

3. Вычисляется $s_1 = c_1 e_1$, $s_2 = c_2 e_2$, $s_3 = c_2$, так, что

$$d = s_1 a_1 + s_2 a_2 + s_3(b_1 + b_2 + h).$$

4. Вычисляется

$$a = a_1 a_2 / d^2,$$

$$b = \left(s_1 a_1 b_2 + s_2 a_2 b_1 + s_1(b_1 b_2 + f)\right)\left(d^{-1} \bmod a\right),$$

5. Return (a, b).

Для вычисления сложности, необходимо сделать следующее допущения: $\mathrm{div}(a, b)$ выбран равновероятно из множества приведенных дивизоров, то и a и b будут выбраны равновероятно из множества полиномиальных функций степени $g-1$. Таким образом, если $D_1 = \mathrm{div}(a_1, b_1)$ и

$D_1 = \mathrm{div}(a_1, b_1)$ равновероятно выбранные из множества приведенных дивизоров, то и a_1, b_1, a_2, b_2 будут равновероятно выбраны среди множества полиномиальных функций степени $g-1$. На выходе алгоритма, шаг 5, полиномиальные функции a, b будут иметь степени $2g$ и $2g-1$ соответственно.

В результате шага 1 алгоритма 1, будут получены полиномиальные функции e_1 и e_2, которые будут согласно определению НОД, в среднем, обладать степенью значительно меньшей g. Аналогичными рассуждениями необходимо руководствоваться и для определения степеней c_1 и c_2. Вычисляемая на шаге 3 алгоритма 1, полиномиальная функция d, также будет обладать степенью значительно меньшей g. Таким образом, для вычислений на шаге 1, 2, 3 необходимы алгоритмы, работающие с полиномиальными функциями степени g (одинарной точности), в то время как на шаге 4 используются алгоритмы, работающие с полиномиальными функциями степени $2g$ (удвоенной точности).

Сложность алгоритма композиции составит:

$$I(A_{compos}) = 2I^{PF}_{\gcd} + 4I^{PF}_{add} + 3I^{2PF}_{add} + 10I^{PF}_{mul} + I^{2PF}_{mul} + I^{PF}_{inv} + I^{PF}_{sqr} + I^{PF}_{div},$$

где I^{PF}_{add}, I^{2PF}_{add} - сложность алгоритма сложения двух полиномиальных функций степени g и $2g$, соответственно, $I^{PF}_{\gcd}$ - сложность алгоритма нахождения НОД двух полиномиальных функций степени g, I^{PF}_{mul}, I^{2PF}_{mul} - сложность алгоритма умножения двух полиномиальных функций степени g и $2g$, соответственно; I^{PF}_{inv} - сложность алгоритма мультипликативного инвертирования полиномиальной функции степени g, I^{PF}_{sqr} - сложность алгоритма возведения в квадрат полиномиальной функции степени g, I^{PF}_{div} - сложность алгоритма деления двух полиномиальных функций степени g.

Рассмотрим алгоритм приведения почти приведенных дивизоров.

Алгоритм 2. Приведение почти-приведенного дивизора

Вход: Почти приведенный дивизор $D = \mathrm{div}(a, b)$ определенный над K.

Выход: Приведенный дивизор $D' = \mathrm{div}(a', b')$, такой, что $D' \sim D$.

1. Вычисляется

$$a' = \left(f + bh + b^2\right)/a$$
$$b' = (h + b) \bmod a'$$

2. Если $\deg_u(a') > g$, тогда $a \leftarrow a'$, $b \leftarrow b'$ и переход к п.1.

3. c - старший коэффициент a', вычисляется $a' \leftarrow a'/c$.

4. Return (a', b').

Для оценки количества итераций цикла пп. 1-2, воспользуемся доказательством теоремы 51 из [11], из которой следует, что если $m = \deg_u a$ и $n = \deg_u b$, $m > n$ и $m \geq g+1$, то $\deg_u a' = \max\{2g+1, 2n\} - m$. Если $m > g+1$, то $\max\{2g+1, 2n\} \leq 2(m-1)$, следовательно $\deg_u a' \leq m-2 < \deg_u a$.

Положим, что $m = 2g$ и $n = 2g-1$, то $\max\{2g+1, 2n\} = \max\{2g+1, 2(2g-1)\} = 4g-2$, то $\deg_u a' = 4g-2-2g = 2g-2$. Другими словами, за одну итерацию произойдет уменьшение степени дивизора на 2.

Положим, что $m = g+1$, то $\max\{2g+1, 2n\} = \max\{2g+1, 2n\} = 2g+1$, то $\deg_u a' = 2g+1-(g+1) = g$. Другими словами, за одну итерацию произойдет уменьшение степени дивизора на 1.

Если рассмотрим случай, при котором $\Pr\{\deg_u a - \deg_u a' == 2\} = \Pr\{\deg_u a - \deg_u a' == 1\}$. Таким образом, количество итераций алгоритма в среднем составит $\lfloor \frac{2}{3} g \rfloor$.

Для вычисления a' и b' воспользуемся алгоритмами удвоенной точности.

Сложность алгоритма приведения составит:

$$I(A_{reduct}) = \lfloor \tfrac{2}{3} g \rfloor \left(I^{2PF}_{mul} + I^{2PF}_{sqr} + 3I^{2PF}_{add} + 1I^{2PF}_{div} + 1I^{PF}_{div}\right) + 1I^{F}_{inv} + gI^{F}_{mul},$$

где I^{F}_{mul} - сложность алгоритма умножения двух элементов поля F_{2^m}, I^{F}_{inv} - сложность алгоритма мультипликативного инвертирования элемента поля F_{2^m}, I^{2PF}_{sqr} - сложность алгоритма возведения в квадрат полиномиальной функции степени $2g$, I^{2PF}_{div} - сложность алгоритма деления двух полиномиальных функций степени $2g$.

Алгоритмы полиномиальной арифметики. Как видно из приведенных алгоритмов, основными операциями являются сложение, возведение в квадрат, умножение, $\gcd(a, b)$, приведение по модулю и мультипликативное инвертирование. В качестве базового поля рассматривается F_{2^m}.

В табл. 1 сведены показатели сложности реализации алгоритмов [18-20], используемых для построения криптосистем на основе гиперэллиптических кривых рода $k \leq g$.

Таблица 1

Алгоритм	Сложность
Сложение	$I(A_{add}) = kI^{F}_{add}$

Возведение в квадрат	$I\left(A_{sqr}\right)=kI_{sqr}^{F}$
Умножение (Карацубы)	$I\left(A_{mul}^{Kar}\right)=8I_{add}^{F}\left(3^{h}-2^{h}\right)+3^{h}I_{mul}^{F}$
Умножение	$I\left(A_{mul}\right)=kn\left[I_{mul}^{F}+I_{add}^{F}\right]$
НОД	$I\left(A_{\gcd}\right)=3k\left(2k+1\right)I_{mul}^{F}+3k^{2}I_{add}^{F}+3I_{inv}^{F}$
Деление $\left(k=2n\right)$	$I\left(A_{div}\right)=\left(k-n\right)\left[nI_{mul}^{F}+I_{mul}^{PF}+2nI_{add}^{F}\right]+I_{inv}^{F}$
Инвертирование	$I\left(A_{inv}\right)=\left(4k^{2}+1\right)I_{mul}^{F}+3k^{2}I_{add}^{F}+1I_{inv}^{F}$

где I_{add}^{F} - сложность алгоритма сложения в поле $\mathbf{F}_{2^m}$; I_{sqr}^{F} - сложность алгоритма возведения в квадрат в поле $\mathbf{F}_{2^m}$; I_{mul}^{F} - сложность алгоритма умножения в поле $\mathbf{F}_{2^m}$, $h=\lceil\log_2 k\rceil$ – показатель степени в выражении $2^h\geq k$, 2^h – количество элементов поля - коэффициентов полиномиальной функции степени k, для которого применим алгоритм Карацубы (без модификаций); $LC\left(A\left(u\right)\right)$ - функция выделения старшего коэффициента полиномиальной функции $A\left(u\right)$; r - количество итераций цикла, число соизмеримо с k, I_{inv}^{F} - сложность алгоритма инвертирования в поле $\mathbf{F}_{2^m}$.

Для выполнения сравнения производительности КПГЭК и КПЭК, определим зависимость рода кривой и размера поля КПГЭК от размера поля КПЭК. При фиксированной стойкости криптосистемы, проведем сравнительную оценку сложности криптопреобразований.

Известно [1, 2, 16], что сложность алгоритма решения ПДЛЭК [2] и ПДЛГЭК [16] составляет $O\left(\sqrt{\pi\left(q-2\sqrt{q}\right)/2}\right)$ и $O\left(g^{5}p^{2-4/(2g+1)+\varepsilon}\right)$ групповых операций. Размер групп определяется согласно теореме Хассе-Вейля об интервальной оценке количества точек алгебраической кривой [12] $q-2\sqrt{q}$ и p^{g}, соответственно.

Зафиксируем размеры групп, криптосистемы на КПЭК над полем $\mathbf{F}_q$ и КПГЭК рода $g>1$ над полем $\mathbf{F}_p$, имеют сравнимые размеры групп при соотношении размеров полей (характеристики 2) как $2^m\approx 2^{ng}$, т.е. $m\approx ng$.

Реализации алгоритмов, применяемых для построения криптосистемы на КПГЭК с кривой рода 2 над полем $GF\left(2^{83}\right)$, обеспечивающую сравнимую стойкость с криптосистемой КПЭК с кривой B-163 над полем $GF\left(2^{163}\right)$ [22]. В качестве исходных данных для проведения анализа была определена сложность арифметических операций в поле полиномиальных функций над полем (табл. 2).

Таблица 2

Алгоритм	Сложность								
	$g=2$			$g=3$			$g=4$		
	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}
Сложение	2			3			4		
Удвоенное сложение	4			6			8		
Возведение в квадрат	2			3			4		
Удвоенное возведение в квадрат	4			6			8		
Умножение (Карацубы)	8	3		26	7		40	9	
Удвоенное умножение (Карацубы)	40	9		98	19		152	27	
НОД	12	30	3	27	63	3	48	108	3
Деление	24	10	1	96	30	1	192	52	1
Удвоенное деление	192	52	1	660	150	1	1344	280	1
Инвертирование	12	17	1	27	37	1	48	67	1

Полученные в табл. 2 данные позволяют произвести расчет сложности алгоритма сложения почти приведенных дивизоров: композиции и приведения дивизоров в полевых операциях.

Таблица 3

Алгоритм	Сложность								
	$g=2$			$g=3$			$g=4$		
	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}	I_{add}^{F}	I_{mul}^{F}	I_{inv}^{F}
Композиция	164	120	8	487	270	8	808	434	8
Приведение	300	165	5	1861	655	7	2348	925	15
Сложение дивизоров	464	285	13	3648	1539	9	4456	1973	17

Соответствующая операция сложения в группе точек эллиптической кривой над полем в проективных координатах Лопеса-Дахаба [23] составляет

$$I\left(A_{add}^{point}\right)=14I_{mul}^{F}+8I_{add}^{F}+3I_{sqr}^{F}.$$

Для проведения сравнительного анализа сложностей КПГЭК и КПЭК, преобразуем сложности алгоритмов сложения дивизоров (табл. 3) и точек (табл. 5) к элементарным операциям, согласно табл. 4, которые выполняются на процессорах с архитектурой IA-32, с использованием результатов изложенных в [21].

В табл. 4 представлены сложности операций в $GF\left(2^{83}\right)$ с неприводимым полиномом $f(x)=x^{83}+x^7+x^4+x^2+1$ и $GF\left(2^{163}\right)$ с неприводимым полиномом $f(x)=x^{163}+x^7+x^6+x^3+1$ на процессорах с архитектурой IA-32.

Таблица 4.

Алгоритм	$GF\left(2^{83}\right)$			$GF\left(2^{163}\right)$		
	I^w_{add}	I^w_{shift}	I_{tbl}	I^w_{add}	I^w_{shift}	I_{tbl}
Сложение	3		-	6		-
Умножение	89	51	-	300	102	-
Умножение с приведением	119	79	-	350	126	-
Возведение в квадрат	6	6	12	12	12	24
Возведение в квадрат с приведением	36	34	12	62	36	24
Приведение по модулю	30	14	-	50	24	
Инвертирование	332	-	-	1141	-	-

Сложность алгоритма сложения составляет:

$$I\left(A^{divisor}_{add}\right)=39623I^w_{add}+18525I^w_{shift}, \quad (3)$$

соответствующая операция сложения в группе точек эллиптической кривой над полем $GF\left(2^{163}\right)$, в проективных координатах Лопеса-Дахаба, согласно [24], составляет:

$$I\left(A^{point}_{add}\right)=5134I^w_{add}+1872I^w_{shift}+72I_{tbl}. \quad (4)$$

Анализ выражений (3) и (4) показывает, что криптосистема на КПЭК обладает, приблизительно, в 8-10 раз большей производительностью, чем КПГЭК. Это объясняется тем, что для оценки сложности КПГЭК использовались универсальные алгоритмы операции сложения дивизоров и операций в поле полиномиальных функций.

Известно [7-9], что сложность алгоритма решения ПДЛЭК [2] и ПДЛГЭК [16] составляет $O\left(\sqrt{\pi\left(q-2\sqrt{q}\right)}/2\right)$ и $O\left(g^5q^{2-4/(2g+1)+\varepsilon}\right)$ групповых операций. Зафиксируем стойкости криптосистем, т.е. алгоритмы криптоанализа будут требовать соизмеримые количества групповых операций $O\left(\frac{1}{2}\sqrt{\pi\left(q_1-2\sqrt{q_1}\right)}\right)\approx O\left(g^5q_2^{2-4/(2g+1)+\varepsilon}\right)$, то есть $\sqrt{\pi\left(q_1-2\sqrt{q_1}\right)}/2\approx g^5q_2^{2-4/(2g+1)+\varepsilon}$, т.к. q_1 и q_2, то можно отбросить слагаемые, которые являются несоизмеримо малыми по отношению q_1, q_2 и не оказывают существенное влияние на равенство. Получим $\sqrt{\pi q_1}/2\approx g^5q_2^{2-4/(2g+1)}$, выразим q_2, $q_2=\left(\frac{\pi}{4g^{10}}q_1\right)^{\frac{2g+1}{2(4g-2)}}$. Согласно [16], криптографический интерес представляет кривые родов 2, 3 и 4, т.к. для родов больших 4, существуют субэкспоненциальный алгоритм решения ПДЛГЭК. В табл. 5 представим зависимость размера поля q_2, степени расширения m_2 поля характеристики 2 ГЭК обеспечивающие соизмеримую стойкость с ЭК $E\left(GF\left(2^{m_1}\right)\right)$ от рода g ГЭК.

Таблица 5

g=2	g=3	g=4
$q_2=\left(\frac{\pi}{4\cdot2^{10}}q_1\right)^{0.42}$	$q_2=\left(\frac{\pi}{4\cdot3^{10}}q_1\right)^{0.35}$	$q_2=\left(\frac{\pi}{4\cdot4^{10}}q_1\right)^{0.32}$
$m_2=0.42\left(\log_2\pi+m_1-\right.$ $\left.-12\right)$	$m_2=0.35\left(\log_2\pi+m_1-\right.$ $\left.-\left(2+10\log_2 3\right)\right)$	$m_2=0.32\left(\log_2\pi+m_2-\right.$ $\left.-22\right)$

В [22] рассматриваются рекомендуемые кривые для использования в криптосистемах на КПЭК. Вычислим размеры полей КПГЭК, которые будут обеспечивать соизмеримую стойкость с КПЭК [22] и представим их в табл. 6. Учитывая требование [1-10], что степень расширения поля характеристики 2 обязана быть простой, в. табл. 6 приведем также такие простые числа, что являются ближайшими большими к полученным.

Таблица 6

g	ЭК (2^{163})	ЭК (2^{233})	ЭК (2^{283})	ЭК (2^{409})	ЭК (2^{571})
2	64,11	93,51	114,51	167,43	235,47
	67	97	117	171	237
3	51,38	75,88	93,38	137,48	194,18
	53	79	97	139	197
4	45,64	68,04	84,04	124,37	176,21
	47	71	87	127	179

На рис. 2 представлены степени расширений полей характеристики 2 ГЭК обеспечивающие соизмеримую стойкость с ЭК [22].

Расчеты проводились для конечных полей $GF\left(2^m\right)$, для значений m из табл. 6 по соответствующим им полиномам.

Таблица 7

Алгоритм	$GF\left(2^{47}\right)$		$GF\left(2^{53}\right)$		$GF\left(2^{67}\right)$	
	I^w_{add}	I^w_{shift}	I^w_{add}	I^w_{shift}	I^w_{add}	I^w_{shift}
Композиция	25312	16492	21686	12420	15116	7560
Приведение	88593	58482	51995	30130	19400	10395
Сложение	113905	74974	73681	42550	34516	17955
	$GF\left(2^{71}\right)$		$GF\left(2^{79}\right)$		$GF\left(2^{97}\right)$	
Композиция	45926	34738	31259	15390	23016	9120
Приведение	159705	87723	73950	37335	29035	12540
Сложение	205631	112461	105209	52725	52051	21660

	$GF(2^{87})$		$GF(2^{97})$		$GF(2^{117})$	
Композиция	53382	25606	47408	20520	29096	10560
Приведение	184905	90801	111709	49780	36795	14520
Сложение	238287	116407	159117	70300	65891	25080
	$GF(2^{127})$		$GF(2^{139})$		$GF(2^{171})$	
Композиция	106830	35154	79847	28350	54000	15120
Приведение	369660	124659	186753	68775	67515	20790
Сложение	476490	159813	266600	97125	121515	35910
	$GF(2^{179})$		$GF(2^{197})$		$GF(2^{237})$	
Композиция	180660	56420	143187	39690	92536	20160
Приведение	621063	200070	332564	96285	115035	27720
Сложение	801723	256490	475751	135975	207571	47880

Вычислим сложности алгоритмов композиции дивизоров для ГЭК родов и над полями $GF(2^m)$, для значений m из табл. 6. В табл. 7 представлены показатели сложности операций над дивизорами ГЭК над полями из табл. 6 на процессорах с архитектурой IA-32.

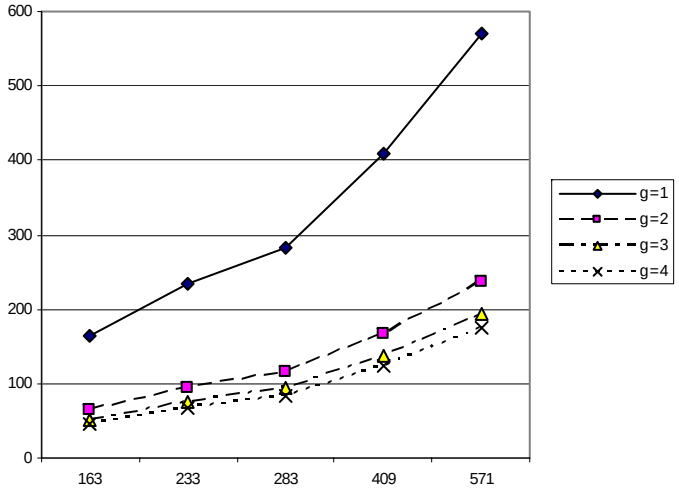

Рис. 2. Зависимость степени расширений полей характеристики 2 ГЭК от рода, при соизмеримой стойкости с ЭК (по вертикали – степени расширения базовых полей ГЭК с соизмеримой стойкостью ЭК, по горизонтали - степени расширения базовых полей ЭК)

В табл. 8 приведены сложности алгоритмов операций сложения точек ЭК над полями [22] в проективных координатах Лопеса-Дахаба [23].

Таблица 8

$GF(2^{163})$		$GF(2^{233})$		$GF(2^{283})$		$GF(2^{409})$		$GF(2^{571})$	
I^w_{add}	I^w_{shift} / I_{tbl}	I^w_{add}	I^w_{shift} / I_{tbl}	I^w_{add}	I^w_{shift} / I_{tbl}	I^w_{add}	I^w_{shift} / I_{tbl}	I^w_{add}	I^w_{shift} / I_{tbl}
5134	1872 / 72	8306	2224 / 96	11600	2876 / 108	21552	3648 / 156	41188	5886 / 228

На графике рис. 3 представлена зависимость количества элементарных операций IA-32 процессора от рода ГЭК, которые обладают соизмеримой стойкостью с ЭК [22].

Проведенный анализ сложности операций над дивизорами в якобиане ГЭК, позволяет сделать вывод, что сложность операций над дивизорами ГЭК значительно выше сложности операции с точками ЭК. Это подтверждается результатами проведенных исследований, которые представлены в табл. 7 и на рис. 3.

Однако, поскольку КПГЭК обладает потенциально большей стойкостью, вопросу понижения сложности групповых операций уделяется достаточно большое внимание [24-34]. Так предлагаются следующие способы снижения сложности групповых операций за счет:

- выбора кривых специального вида;
- рассмотрения кривых четко заданного рода;
- формул сложения соответствующие весу дивизоров [26-31];
- однородных координат для представления [32-34];
- использования метода Монтгомери;
- использования расширенного набора SIMD инструкций процессоров, для параллельного выполнения операций в базовом поле кривой [25].

Выводы. Таким образом, проведенный анализ производительности криптосистем на ЭК и ГЭК, и существующих способов понижения сложности групповых операций, позволяет сделать следующие выводы:

1. Основным достоинством криптосистемы на ГЭК является предоставление большего уровня стойкости при одинаковых размерах полей и роде большем 1 (ЭК), что позволяет при фиксированной длине ключа и кратности групповой операции, получить значительный прирост стойкости криптосистемы на основе ГЭК.

2. Недостатком криптосистемы на ГЭК является высокая сложность групповых операций, что накладывает ограничение на практическое использование несимметричных криптосистем с повышенными требованиями к производительности.

3. Перспективным направлением дальнейшего развития криптосистем на ГЭК является разработка способов снижения сложности преобразований, что

снимет основное ограничение для практического использования – высокую вычислительную сложность.

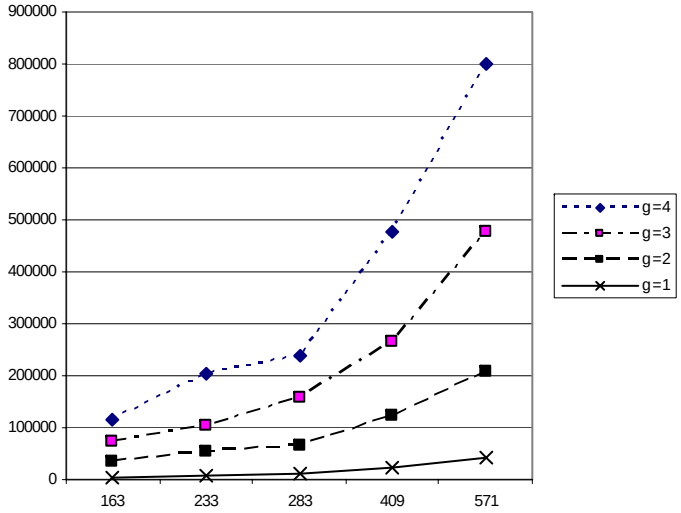

Рис. 3. Зависимость показателя сложности операций над дивизорами ГЭК от рода и операций над точками ЭК на процессорах с архитектурой IA-32 (по вертикали – усредненное количество операций, по горизонтали – степень расширения базового поля)

4. Наиболее перспективными способами снижения сложности преобразований являются, на наш взгляд: выбор кривых специального вида, рассмотрение кривых четко заданного рода, использования формул сложения соответствующие весу дивизоров [26-31], использования однородных координат для представления дивизоров [32-34], использования расширенного набора SIMD инструкций процессоров, для параллельного выполнения операций в базовом поле кривой [25].

Известные способы не позволяют в полной мере снизить сложность преобразований до уровня сложности криптосистем на ЭК. Научно-техническая задача разработки эффективного метода снижения сложности криптопреобразований на ГЭК за счет модификации известных методов, является актуальной.

Список литературы. **1**. Update on the section of algorithms for further investigation during the second round. NESSIE report. D18. 11.03.2002. **2**. IEEE P1363-2000. Standard Specifications for Public Key Cryptography. **3**. X9.62-1998 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). **4**. ANSI X9.42-1998, Public Key Cryptography for The Financial Service Industry: Agreement of Symmetric Keys on Using Diffie-Hellman and MQV Algorithms. **5**. X9.63-199x Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. **6**. International Organization for Standardization. ISO/IEC FCD 15946-4. Information technology – Security techniques –Cryptographic techniques based on elliptic curves – Part2: Digital signatures. **7**. International Organization for Standardization. ISO/IEC FCD 15946-2. Information technology – Security techniques –Cryptographic techniques based on elliptic curves – Part2: Digital signatures **8**. International Organization for Standardization. ISO/IEC 14888-3. Information technology – Security techniques – Digital signatures with appendix. 1999. **9**. ГОСТ Р 34.10-2001. Государственный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной подписи. М. Росстандарт. **10**. ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що грунтується на еліптичних кривих. Формування та перевірка. **11**. *A.J. Menezes, Y. Wu, R.J. Zuccherrato.* An elementary introduction to hyperelliptic curves. Technical report CORR96-19, Department of combinatorics and optimization, University of Waterloo, Waterloo, Ontario, 1996. In: Koblitz, N.: Algebraic aspects of cryptography, Springer-Verlag, Berlin Heidelberg New York. 1998. **12**. *G. Frey.* Applications of arithmetical geometry to cryptographic constructions. **13**. *И.Р. Шафаревич.* Основы алгебраической геометрии. –М.: Наука. Гл. ред. Физ.-мат. Лит., 1972. -568 с. **14**. *С.А. Степанов.* Арифметика алгебраических кривых. –М.: Наука. Гл. ред. Физ.-мат. Лит., 1991. -368 с. –ISBN 5-02-014607-2. **15**. *W. Fulton.* Algebraic curves. Benjamin. New York. 1969. **16**. *M. Jacobson, A. Menezes, A. Stein.* Hyperelliptic curves and cryptography. Fields Institute Communications. **17**. *R.M. Avanzi.* Aspects of hyperelliptic curves over large prime fields in software implementations. December 17, 2003. **18**. *T. Wollinger.* Computer architectures for cryptosystems based on hyperelliptic curves. A thesis submitted to the faculty of the Worcester polytechnic institute. In partial fulfillment of the requirements for the degree of master of science in electrical engineering. April 2001. **19**. *T. Wollinger, C. Paar.* Hardware architecture proposed for cryptosystems based on hyperelliptic curves. **20**. *A. Enge.* The extended Euclidean algorithm on polynomials, and the computational efficiency of hyperelliptic cryptosystems. November, 1999. **21**. *Ю.В. Стасев, С.А. Головашич, В.Ю. Ковтун* Сравнительный анализ алгоритмов умножения и приведения по модулю в поле $GF(2^m)$ // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2003. Вып. 135. С. 129-141. **22**. Recommended elliptic curves for federal government use. National institute of standards and technology. Digital signature standard. FIPS publication 186-2. 2000. **23**. *Ю.В. Стасев, В.Ю. Ковтун, А.А. Смирнов, Я.Ю. Стасева.* Анализ методов представления точек эллиптической кривой над двоичными полями // Системы обработки информации. -2002. -Вып. 5(21). **24**. *Yasuyuki Sakai, Kouichi Sakurai.* On the practical performance of hyperelliptic curve cryptosystems in software implementation. IEICE Trans. Fundamentals, vol. E83-A, No. 4, April 2000. **25**. *P. K. Mishra, P. Sarkar.* Parallelizing explicit formula for arithmetic in the jacobian of hyperelliptic curves (full version). India. **26**. *H Sugizaki, K. Matso, J. Chao, S. Tsujii.* A generalized Harley algorithm for genius two hyperelliptic curves. Technical report of IEICE. SCIC 2003, Japan. **27**. *H. Sugizaki, K. Matsuo, J. Chao, S. Tsujii.* An extension of Harley addition algorithm for hyperelliptic curves over finite fields of characteristic two. Technical report of IEICE. ISEC2002-9. Japan. **28**. *P. Gaudry, R. Harley.* Counting points on hyperelliptic curves over finite fields. ANTS-IV LNCS 1838, Springer-Verlag, 2000, pp. 297-312. **29**. *K. Matsuo, J. Chao, S. Tsujii.* Fast genius two hyperelliptic curve cryptosystems. Technical report of IEICE, ISEC2001-07. Japan. **30**. *R. Harley*, adding.c, available at: http://crystal.infra.fr/~harley/hyper/, 2000. **31.** *R. Harley*, doubling.c, available at: http://crystal.infra.fr/~harley/hyper/, 2000. **32**. *T. Lange.* Efficient arithmetic on genius 2 hyperelliptic curves over finite fields via explicit formulae. December 15, 2003. Available at: http://eprint.iacr.org. **33**. *T. Lange.* Weighted coordinates on genius 2 hyperelliptic curves. October 11, 2002. Available at: http://eprint.iacr.org. **34**. *T. Lange.* Inversion-free arithmetic on genius 2 hyperelliptic curves. September 22, 2002. Available at: http://eprint.iacr.org.

Поступила в редакцию 06.10.2004