УДК 681.3.06

Г.В. Неласа

Запорізький національний технічний університет

МЕТОДИ СКАЛЯРНОГО МНОЖЕННЯ НА ЕЛІПТИЧНИХ КРИВИХ

Розглянуто методи виконання однієї з важливих криптографічних операцій – операції скалярного множення на еліптичній кривій. Запропоновано паралельні двократні методи виконання цієї операції.

The methods of one of the relevant cryptography operations - operation of a scalar multiplication on an elliptic curve are considered. The parallel 2th-order methods of execution of this operation are offered.

Вступ

Еліптичні криві застосовуються у криптографії з відкритим ключем. Зокрема, на них побудовані стандарти цифрового підпису [1; 2; 3]. Сьогодні криптосистеми на основі еліптичних кривих з довжиною ключа 160 біт мають однакову стійкість із криптосистемами ElGamal і RSA з довжиною ключа 1024 біт. Однак розвиток методів криптоаналізу і збільшення потужності обчислювальної техніки приводять до того, що вимоги, пропоновані до швидкості обчислень і стійкості, неухильно зростають. Отже, актуальним є завдання розробки нових ефективних алгоритмів, що використовуються у рамках методів криптографічних перетворень на еліптичних кривих.

Еліптична крива – це математичний об'єкт, що може бути визначений над будь-яким полем, зокрема над полем раціональних чисел чи над скінченним полем Галуа. У загальному вигляді рівняння еліптичної кривої у формі Вейерштрасса над полем K має вигляд

$$E(K): y^2 + axy + by = x^3 + cx^2 + dx + e,$$

де $a, b, c, d, e \in K$.

У криптографії застосовується особлива форма еліптичної кривої, визначеної над скінченним полем Галуа $GF(q)$, де модуль q є або простим числом, або степенем простого числа. Найчастіше вживаються криві над простим полем $GF(p)$, де p – велике просте число і криві над полем $GF(2^m)$:

$$E(GF(\text{p})): y^2 = x^3 + ax + b \bmod p,$$

де $a, b \in GF(p)$, $4a^3+27b^2 \neq 0 \bmod p$; p – просте число. $E(GF(2^m)): y^2 + xy = x^3 + ax^2 + b \bmod (f(x),2)$, де $a, b \in GF(2^m)$, $f(x) \in GF(2^m)$ – незвідний.

Множина точок еліптичної кривої, визначеної над скінченним полем, містить у собі, крім точок (x_i, y_i), що задовольняють рівнянню еліптичної кривої, також точку на нескінченності, що позначається як O. На такій множині можна визначити операцію додавання точок, наприклад, для кривої над простим полем, у такий спосіб:

$$P_3 = P_1 + P_2 = (x_1, y_1) + (x_2, y_2) = (x_3, y_3),$$

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p,$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p.$$

Якщо $P_1 \neq P_2$, то $\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\bmod p)$.

Якщо $P_1 = P_2$, то $\lambda = \frac{3x_1^2 + a}{2y_1} (\bmod p)$.

Ця множина з операцією додавання точок утворює адитивну абелеву групу з нулем у точці O. Основною криптографічною операцією на еліптичній кривій є операція скалярного множення

$$Q = c \times P = \underbrace{P + P + \cdots + P}_{c \text{ раз}},$$

де точка P – генератор групи; множник c – ціле число – секретний ключ; точка Q – відкритий ключ.

Основою криптографічної стійкості перетворень на еліптичних кривих є велика обчислювальна складність задачі знаходження числа c для відомих P і Q, що називається проблемою дискретного логарифма для еліптичної кривої. За результатами сучасних досліджень найкращим методом знаходження дискретного логарифма на еліптичній кривій є розподілений алгоритм ρ -Полларда.

Алгоритми скалярного множення на еліптичних кривих

Стандартним алгоритмом скалярного множення на еліптичній кривій є так званий "бінарний" алгоритм. У ньому використовується бінарне подання множника. Якщо поточний біт дорівнює 1, подвоєння доповнюється додаванням з базовою точкою.

Алгоритм. Бінарне скалярне множення на еліптичній кривій.

Вихідні дані: число $c \neq 0$, точка P, еліптична крива $E = <a, b, p>$.

Результат: точка $Q = c \times P$.

1. Якщо $c = 1$, то $Q := P$; закінчити роботу алгоритму.
2. $k := l_c - 2$; $Q := P$, де l_c – довжина числа c у бітах;
3. Для i, що приймає значення від k до 0, виконати кроки 4–5.
4. $Q := Q + Q$.
5. Якщо i-й біт c дорівнює 1, то $Q := Q + P$.
6. Закінчити роботу алгоритму.

Обчислювальна складність цього алгоритму визначається в такий спосіб:

$$(\lfloor \log_2 c \rfloor + H(c) - 1) \times t_o ,$$

де $c = (c_{N-1} c_{N-2} \ldots c_2 c_1 c_0)$ – бінарне подання множника; $N = \lfloor \log_2 c \rfloor + 1$ – довжина множника в бітах; $H(c) = \sum_{i=0}^{N-1} c_i$ – вага Хеммінга множника; t_o – час виконання однієї операції (додавання чи подвоєння точок) на еліптичній кривій. Причому вважаємо час додавання точок приблизно таким, що дорівнює часу подвоєння.

У M-арних алгоритмах як основу системи числення для подання множника часто використовують різні ступені двійки. Це дає змогу трохи прискорити обчислення за рахунок використання додаткової пам'яті для таблиці передобчислень [4].

Ще один різновид алгоритмів скалярного множення на еліптичній кривій – це методи вікон. Вони є розвитком ідеї M-арних алгоритмів. Підвищення швидкості досягається за рахунок того, що розмір вікна вибирається динамічно [4].

Способами підвищення швидкості обчислень також є подання множника в спеціальному вигляді з метою зменшення одиничних розрядів. Наприклад, при множнику $c=15$ для одержання відкритого ключа $Q=cP$ необхідно обчислити $15P$. У стандартному поданні це виглядає так: $15P = 1111_2 P = (((2P+P)2+P)2+P$, тобто для обчислення необхідними будуть три операції подвоєння і три додавання. Якщо ж записати $15P = (16-1)P = (10000_2-1)P = (2P)\,2\,2\,2-P$, то одержимо чотири операції подвоєння й одну віднімання. З врахуванням того, що операції подвоєння виконуються швидше від додавання, алгоритм прискорює обчислення, особливо за великих значень числа c.

Для прискорення обчислень можна також використовувати трійкове подання множника $c_i \in \{-1,0,1\}$ і той факт, що, якщо P=(x,y), то –P=(x,-y). Однак на відміну від бінарного, таке подання є неоднозначним. Існують різні алгоритми оптимального подання множника в трійковій системі числення [5].

У [5] проведено порівняльний аналіз бінарного методу, знакового бінарного методу, методу Якобі, методу вікна, Bos-Coster's-методу і методу знакового бінарного вікна. Аналіз показав, що найменшу кількість операцій має метод знакового бінарного вікна – у середньому в 1,27 раза менше, ніж стандартний бінарний алгоритм.

Задача пошуку оптимального алгоритму обчислення $Q=cP$ у загальному вигляді зводиться до математичної задачі пошуку найкоротшого адитивного ланцюжка [6]. Адитивним ланцюжком для n називається послідовність цілих чисел

$$1 = a_0, a_1, a_2, \ldots, a_r = n,$$

які мають властивість, що

$$a_i = a_j + a_k \text{ для деяких } k \le j < i,$$

для всіх $i=1,2,\ldots,r$.

Дотепер не існує поліноміального алгоритму, що розв'язує задачу пошуку найкоротшого адитивного ланцюжка. Ця задача належить до класу складності NP. Однак відомо, що знизу довжина адитивного ланцюжка обмежена числом [6; 4]

$$\log_2 n + \log_2 H(n) - 2{,}13 .$$

У [7] запропоновано p-кратний бінарний метод для розподілу операції скалярного множення на еліптичних кривих на p процесорів. Суть методу полягає в тому, що множник в певний спосіб за допомогою швидкої операції маскування розбивається на суму p чисел, що мають вагу Хеммінга в p разів меншу від вихідного множника. Потім у p процесорів обчислюють відповідно добуток базової точки на кожне з отриманих чисел. На цьому етапі відсутні пересилання даними між процесорами. Вони з'являються лише на кінцевому етапі, коли необхідно скласти p отриманих точок.

За теоремою з [7] час виконання цього алгоритму в кращому випадку становить

$$(\lfloor \log_2 c \rfloor + \lceil H(c)/p \rceil + \lceil \log_2 p \rceil - 1) \times t_o ,$$

у гіршому

$$(\lfloor \log_2 c \rfloor + H(c) - 1) \times t_o .$$

Як можна побачити з наведених вище формул, кількість операцій над точками еліптичної кривої у розглянутих методах тією чи іншою мірою залежить від кількості одиниць в бінарному поданні множника, тобто від $H(c)$.

Двократні паралельні методи скалярного множення на еліптичних кривих

У [8] запропоновано паралельний двократний бінарний алгоритм скалярного множення на еліптичних кривих. Суть запропонованого методу полягає в тому, що один із двох паралельно працюючих процесорів виконує послідовне подвоєння базової точки, а другий у цей самий час додає точки відповідно до положення одиничних бітів бінарного подання множника.

Алгоритм. Паралельне двократне бінарне скалярне множення на еліптичній кривій.

Вихідні дані: число $c = (c_{k+1}, \ldots, c_1, c_0) \neq 0$, точка P, еліптична крива $E = <a, b, p>$.

Результат: точка $Q = c \times P$.

1. Якщо $c = 1$, то $Q := P$; закінчити роботу алгоритму.
2. $k := l_c - 2$; , де l_c – довжина числа c у бітах;
3. Якщо $c_0 = 1$, то $Q_1 := P$, інакше $Q_1 = O$.
4. $Q_2 := P + P$.
5. Для i, що приймає значення від 1 до k, виконати:

Процесор A: якщо $c_i = 1$, то $Q_1 := Q_1 + Q_2$.

Процесор B: $Q_2 := Q_2 + Q_2$.

6. Якщо $Q_1 \neq O$, $Q := Q_1 + Q_2$, інакше $Q := Q_2$.
7. Закінчити роботу алгоритму.

Перевага запропонованого методу полягає в тому, що кількість операцій над точками не залежить від кількості одиниць $H(c)$ у бінарному поданні множника, а визначається тільки його довжиною. Однак кількість операцій пересилання даних між потоками дорівнює $H(c)$-2. Загальний час виконання алгоритму

$$(\lfloor \log_2 c \rfloor + 1) \times t_o + (H(c) - 2) \times t_p ,$$

де t_p – час, необхідний на виконання одного пересилання між процесорами.

Запропонований алгоритм ефективний у тому разі, коли $t_p << t_o$. Виграш є особливо значний для таких значень множника, бінарні подання яких містять багато одиночних розрядів. Якщо орієнтуватися на середнє значення кількості одиниць в словах, що обробляються, рівною половині їхньої довжини, то одержимо виграш у підвищенні швидкодії процедури скалярного множення в 1,5 раза.

Через нерівномірне завантаження процесора А, запропонований алгоритм уразливий до часового аналізу з завантаження процесора А. Щоб уникнути цієї уразливості, під час програмної реалізації рекомендується в циклах, що відповідають нульовим бітам множника, виконувати неробочий хід. Наприклад: додавання двох випадкових точок кривої без наступного використання результату.

Розроблений алгоритм був реалізований мовою програмування Microsoft Visual C++6.0 з використанням бібліотеки багатократної арифметики [9] та бібліотеки системних функцій [10] і протестований на комп'ютері з двома процесорами x86 Family 6 Model 8 Stepping 6 Genuine Intel ~ 933 Mhz і загальною фізичною пам'яттю – 512 М під керуванням операційної системи Microsoft Windows 2000 Advanced Server.

Реалізація виконувалась двома способами: асинхронним та синхронним. За асинхронного способу процес, що передає дані, записує їх у тимчасовий динамічний список і продовжується далі, незалежно від того, було прийнято дані другим процесом, чи ні. Другий процес вибирає дані із створеного першим процесом списку у міру їх надходження. Така реалізація потребує деякої кількості додаткової пам'яті для динамічного списку, але виконується скоріше. За синхронної реалізації додаткова пам'ять не потрібна, але на синхронізацію витрачається трохи більше часу.

Результати тестування для множника завдовжки 256 біт наведені в табл. 1. В табл. 2 наведено значення прискорення $S = \frac{T_1}{T_2}$, де T_1, T_2 – час виконання алгоритмів, що порівнюються.

Таблиця 1

Час виконання алгоритмів скалярного множення на еліптичній кривій

H(c)%	Стандартний бінарний, мкс	Паралельний p-кратний на 2-х процесорах, мкс	Паралельний двократний бінарний (асинхронна реалізація), мкс	Паралельний двократний бінарний (синхронна реалізація), мкс	% часу, що втрачається на пересилання для двократного бінарного (асинхронна реалізація)	% часу, що втрачається на пересилання та синхронізацію для двократного бінарного (синхронна реалізація)
1	2	3	4	5	6	7
0,39	49985,06	50356,27	50784,97	50768,48	0,00	0,00
1,56	50480,25	50991,47	51059,21	51166,67	0,54	0,78
6,25	52760,26	53773,47	51189,96	51671,76	0,80	1,78
12,50	56028,65	54075,68	51377,01	51600,17	1,17	1,64
18,75	58961,51	55416,76	51554,50	53148,64	1,52	4,69
25,00	61862,26	57048,77	52000,33	52162,84	2,39	2,75
31,25	65375,69	58932,82	51813,26	52624,08	2,02	3,66
37,50	68320,78	60228,80	52184,67	52862,54	2,76	4,12
43,75	71421,80	61885,37	52297,93	53181,82	2,98	4,75
50,00	74165,47	64894,76	52319,96	53699,26	3,02	5,77
56,25	77387,14	65607,62	52776,21	54446,37	3,92	7,24
62,50	80397,00	66751,05	52855,66	54947,95	4,08	8,23
68,75	83512,27	68309,84	52847,10	55429,31	4,06	9,18
75,00	86798,67	69862,52	53249,09	55778,25	4,85	9,87
81,25	89820,27	71297,53	53220,42	55359,50	4,80	9,04
87,50	92836,91	72913,73	53536,53	55880,31	5,42	10,07
93,75	95988,92	74703,68	53748,65	56089,36	5,84	10,48
100,0	98909,34	75618,94	53668,82	56498,52	5,68	11,29

Таблиця 2

Прискорення алгоритмів скалярного множення на еліптичних кривих

Не(с) %	Прискорення паралельного р-кратного на 2-х процесорах порівняно із стандартним бінарним	Прискорення паралельного двократного бінарного порівняно із стандартним бінарним (асин. реалізація)	Прискорення паралельного двократного бінарного порівняно із р-кратним на 2-х процесорах (асин. реалізація)	Прискорення паралельного двократного бінарного порівняно із стандартним бінарним (син. реалізація)	Прискорення паралельного двократного бінарного порівняно із р-кратним на 2-х процесорах (син. реалізація)
0,39	0,99	0,98	0,99	0,98	0,99
1,56	0,99	0,99	1,00	0,99	1,00
6,25	0,98	1,03	1,05	1,02	1,04
12,50	1,04	1,09	1,05	1,09	1,05
18,75	1,06	1,14	1,07	1,11	1,04
25,00	1,08	1,19	1,10	1,19	1,09
31,25	1,11	1,26	1,14	1,24	1,12
37,50	1,13	1,31	1,15	1,29	1,14
43,75	1,15	1,37	1,18	1,34	1,16
50,00	1,14	1,42	1,24	1,38	1,21
56,25	1,18	1,47	1,24	1,42	1,20
62,50	1,20	1,52	1,26	1,46	1,21
68,75	1,22	1,58	1,29	1,51	1,23
75,00	1,24	1,63	1,31	1,56	1,25
81,25	1,26	1,69	1,34	1,62	1,29
87,50	1,27	1,73	1,36	1,66	1,30
93,75	1,28	1,79	1,39	1,71	1,33
100,0	1,31	1,84	1,41	1,75	1,34

На рис. 1 – 3 по осі X відзначена кількість одиниць у множнику у відсотках від 0 до 100 %.

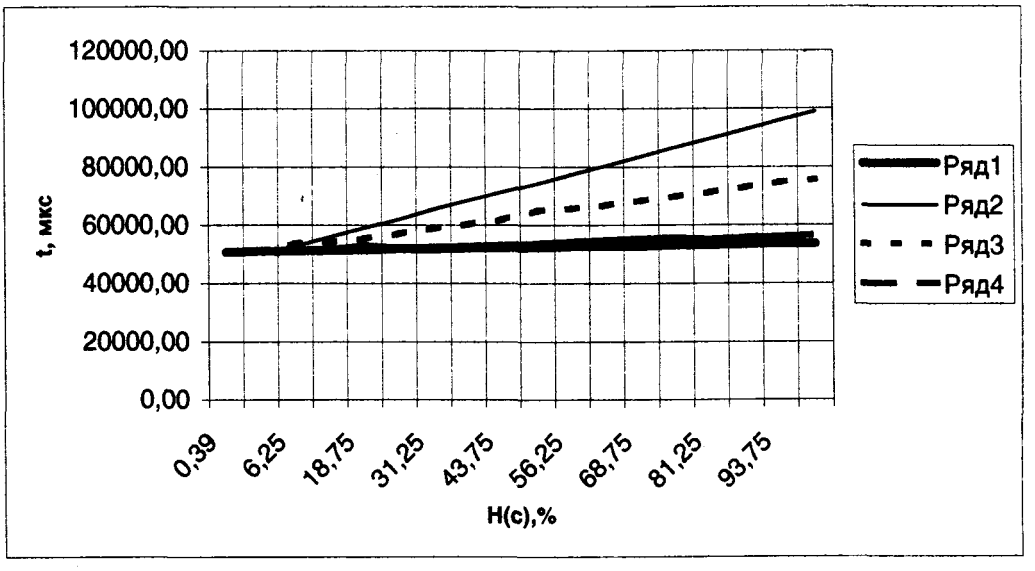

Рис. 1. Час виконання операції скалярного множення:
ряд 1 – паралельний двократний алгоритм (асинхронна реалізація);
ряд 2 – стандартний бінарний алгоритм;
ряд 3 – паралельний р-кратний алгоритм на двох процесорах;
ряд 4 – паралельний двократний алгоритм (синхронна реалізація)

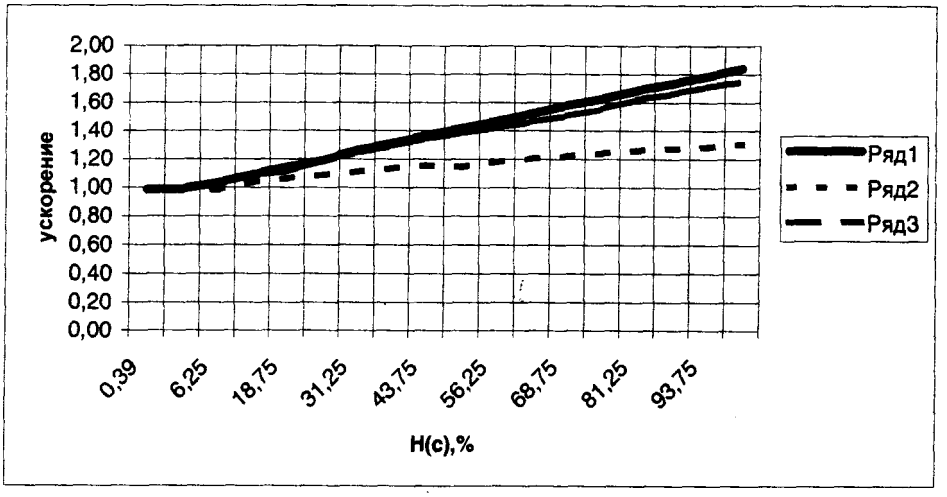

Рис. 2. Прискорення операції скалярного множення порівняно із стандартним бінарним алгоритмом:
ряд 1 – прискорення паралельного двократного алгоритму (асинхронна реалізація);
ряд 2 – прискорення паралельного р-кратного на 2 процесорах;
ряд 3 – прискорення паралельного двократного алгоритму (синхронна реалізація)

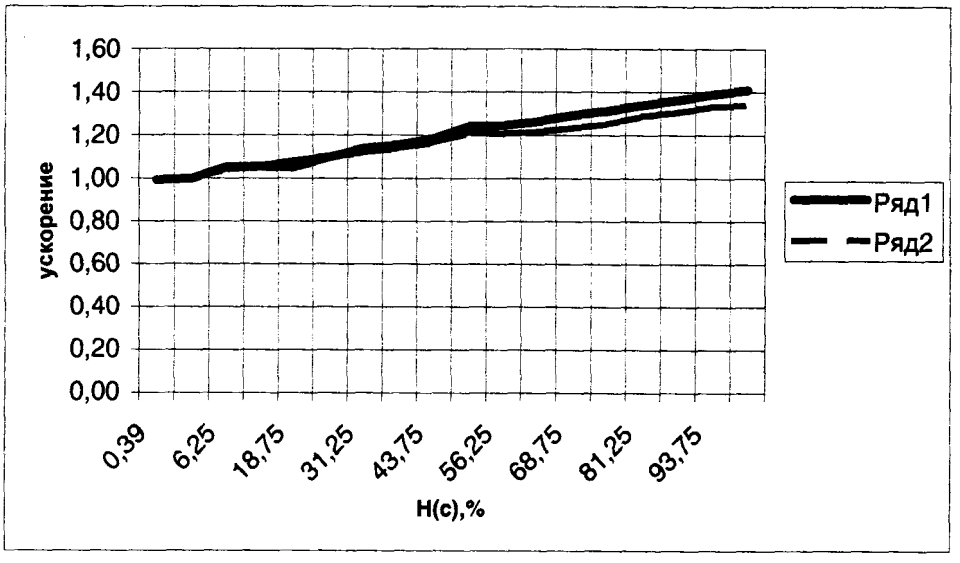

Рис. 3. Прискорення паралельного двократного порівняно із паралельним р-кратним на 2 процесорах: ряд 1 – асинхронна реалізація; ряд 2 – синхронна реалізація

Як бачимо з результатів тестування, зі збільшенням кількості одиниць у множнику час роботи стандартного бінарного алгоритму зростає майже вдвічі, р-кратного паралельного алгоритму на двох процесорах – у 1,5 раза, а час роботи запропонованого двократного паралельного алгоритму зростає незначно тільки за рахунок збільшення кількості пересилань.

Якщо ж час пересилань t_p значний, з метою зменшення кількості пересилань автором пропонується М-арний паралельний двократний метод скалярного множення на еліптичних кривих. У цьому методі множник подається в системі числення з основою М. Для ефективної програмної й апаратної реалізації зручно використовувати $M = 2^p$.

Алгоритм. Паралельне М-арне двократне скалярне множення на еліптичній кривій.

Вихідні дані: $M=2^p$ – основа системи числення, множник $c=(c_{k-1},\ldots,c_1,c_0)\neq 0$, поданий у системі числення з основою М; точка P, еліптична крива $E = <a, b, p>$.

Результат: точка $Q = c\times P$.

1. Якщо $c = 1$, то $Q := P$; закінчити роботу алгоритму.
2. k=кількість М-ичних цифр числа c.

3. Q: = O, $Q_2 = P$.

4. Для i, що приймає значення від 0 до k-1, виконати:

Процесор A: якщо $c_i > 0$, то Q:= $Q + c_i \cdot Q_2$.

Процесор B: для j приймаючого значення від 1 до p: Q_2:= $Q_2 + Q_2$.

5. Закінчити роботу алгоритму.

Але в цьому алгоритмі за рахунок зменшення кількості пересилань збільшується кількість операцій над точками еліптичної кривої, тому що на кроці 4 алгоритму процесор А обчислює $ci \cdot Q2$ за допомогою стандартного бінарного алгоритму або його будь-якої з описаних вище однопроцесорних модифікацій. Його можна використовувати, коли t_p порівняно з t_o

На рис. 4 показаний час виконання паралельного двократного М-арного алгоритму порівняно із стандартним бінарним методом для тестового множника з [2] завдовжки 256 біт. По осі X відзначено степені 2 для основи системи числення $M = 2^p$ для подання множника c.

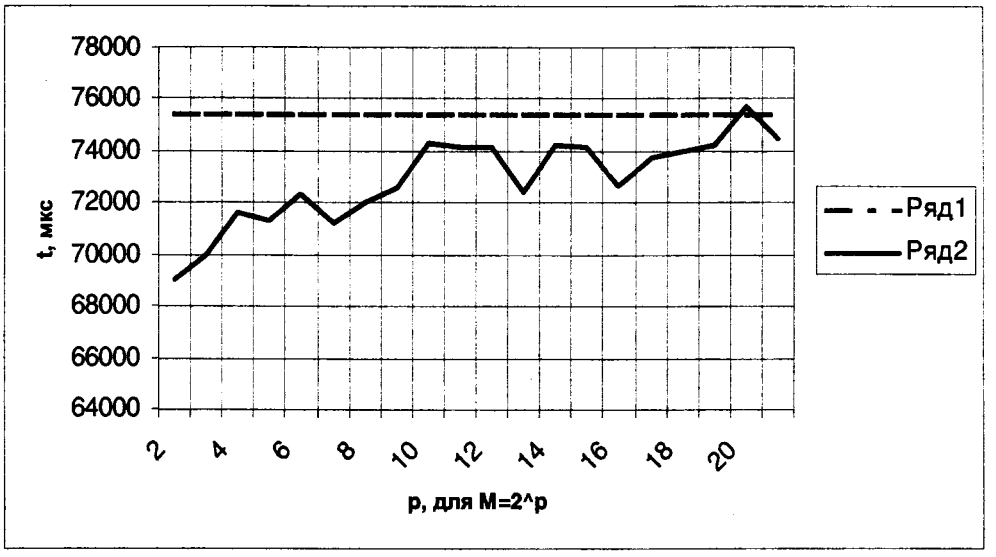

Рис. 4. Час виконання алгоритмів скалярного множення на еліптичних кривих:
ряд 1 – стандартного бінарного алгоритму;
ряд 2 – паралельного двократного М-арного алгоритму

Як зрозуміло з рис. 4, зі збільшенням p час виконання паралельного двократного М-арного алгоритму поступово, але нерівномірно, зростає. Ця нерівномірність зумовлена випадковою кількістю одиниць в часткових множниках-цифрах в М-ічному поданні. Оскільки в системі, в якій тестувались алгоритми, час на пересилання даними між процесами замалий, цей алгоритм виявився не таким ефективним, як розглянутий вище, паралельний двократний бінарний алгоритм.

Висновки

Запропоновані паралельні двократні методи скалярного множення на еліптичних кривих забезпечують значне зменшення часу обчислень. Вони не залежать від типу поля, над яким визначена крива, тобто можуть застосовуватися і для кривих над розширеними полями.

1. ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка. – Київ: Держстандарт України, 2003. 2. ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки цифровой подписи. – М.: Госстандарт России, 2001. 3. ANSI X9.62. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999. 4. Вельшенбах М. Криптография на Си и С++ в действии: Учеб. пособие.– М.: Издательство Триумф, 2004. 5. Kenji Koyama, Yukio Tsuruoka.

Speeding up elliptic cryptosystems by using a signed binary window method, Advances in Cryptology. – CRYPTO'92, LNCS 740. – 1993. – P. 345–357. 6. Кнут Д. Искусство программирования для ЭВМ. Том 2. Получисленные алгоритмы. – М.: Мир, 1976. 7. Juan Manuel Garcia Garcia, Rolando Mechaca Garcia Parallel Algorithm for Multiplication on Elliptic Curves. 8. Nelasa A.V., Dolgov V.I. Multisequencing of operation of a scalar multiplication on elliptic curve//Proceedings of the International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics" (CADSM'2005). – Lviv-Polyana, 2005. – P. 122–123. 9. Библиотека многократной арифметики MMATH v. 1.12 (12 августа 1996 г.) © АО "ИИТ" (г. Харьков) 10. Пинчук В.П. Библиотека VP / C++. Модуль syst.h. – Запорожье: ЗНТУ, 2003.

УДК 519.8766.5

М.П. Дивак, Ю.П. Франко, М.Я. Шпінталь

Тернопільська академія народного господарства

ОЦІНЮВАННЯ ДОПУСТИМИХ ЗНАЧЕНЬ ПАРАМЕТРІВ БАГАТОЕЛЕМЕНТНОЇ СТАТИЧНОЇ СИСТЕМИ НА ОСНОВІ АНАЛІЗУ ЇЇ ІНТЕРВАЛЬНИХ ХАРАКТЕРИСТИК

Розглянуто задачу допустимого оцінювання параметрів статичної багатоелементної системи. Запропоновано метод допустимого оцінювання параметрів складних систем, в якому допустимі області оцінок параметрів подані у вигляді багатовимірних еліпсоїдів.

The task of tolerance parameters estimation of the static multi-element systems based on the interval analysis is considered. Method of tolerance estimation of the parameters of the complex systems is proposed. In the method the tolerance estimation sets in form of multidimensional ellipsoids is represented.

Постановка проблеми

Однією із проблем синтезу складних багатоелементних статичних систем є проблема знаходження таких значень її параметрів, які забезпечать задані характеристики – виходи усієї системи [1]. Надалі під параметрами системи розумітимемо характеристики її складових елементів, підсистем, модулів. Враховуючи, що під час виготовлення елементів системи виникають випадкові технологічні відхилення їх характеристик від розрахункових, то природним є ризик, що побудована система може бути функціонально непридатною. Як відомо, достатньо часто функціональна придатність системи задається імовірністю її працездатності [2]. За цих умов синтез складних статичних систем полягає у знаходженні допустимих значень параметрів підсистеми, які забезпечуватимуть допустимі характеристики системи із заданою імовірністю її працездатності [1].

Для синтезу допустимих значень параметрів системи використовують гарантований та стохастичний підходи [2]. Перший полягає у знаходженні таких допустимих значень параметрів системи, які забезпечують повну гарантію працездатності. Однак при цьому отримуватимемо надзвичайно жорсткі вимоги до технологічного процесу синтезу елементів системи. У разі стохастичного підходу традиційні методи вимагають надзвичайно складних процедур розв'язування задач нелінійного програмування високої розмірності. Враховуючи, що значення виходів системи є встановленими в певних межах допустимих значень, тобто у вигляді числових інтервалів, для вирішення вказаної проблеми можливим є застосування методів аналізу інтервальних даних [2].

Під час розв'язування задач синтезу допустимих значень параметрів елементів математичні моделі статичних систем зображають системами інтервальних рівнянь [2]. Для оцінювання розв'язків інтервальних систем можливим є застосовування відомих методів допустимого оцінювання.