УДК 581.3

Informatics and Mathematical Methods in Simulation
Vol. 5 (2015), No. 4, pp. 376-382

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДІВ МОДУЛЯРНОГО ЕКСПОНЕНЦІЮВАННЯ

М.М. Касянчук, І.З. Якименко, Т.М. Долинюк, Н.А. Рендзеняк

Тернопільський національний економічний університет,
вул. Львівська, 11, Тернопіль, 46020, Україна; e-mail: kasyanchuk@ukr.net, jiz@tneu.edu.ua

Виконання арифметичних операцій над багаторозрядними числами є досить важливою задачею сучасної теорії чисел та асиметричної криптографії. У роботі проведено експериментальне дослідження часової складності модулярного експоненціювання з використанням середовища програмування Python 3.4.0 різними методами: пониження степеня за допомогою виділення квадратів (бінарний), кубів (3-арний), системи залишкових класів та її модифікованої досконалої форми. Операції виконувалися над числами різної розрядності та з різною вагою Хемінга. Показано, що при малих розрядностях найбільшою швидкодією характеризується метод пониження степенів за допомогою кубів. Починаючи з 1024 біт найменший час для виконання операції модулярного піднесення до степеня витрачається при застосуванні методу модифікованої досконалої форми системи залишкових класів. При дослідженні часової складності модулярного експоненціювання в залежності від ваги Хемінга встановлено, що найбільший час затрачається, коли вага Хемінга максимальна. При її зменшенні час виконання різко зменшується і надалі залишається майже постійним.

Ключові слова: асиметрична криптографія, модулярне експоненціювання, пониження степеня, модифікована досконала форма системи залишкових класів, часова складність, вага Хемінга.

Вступ

Розробка стійких систем захисту інформаційних потоків забезпечується високою продуктивністю при реалізації на основі обчислювальних термінальних пристроїв в комп'ютерних мережах існуючих мережевих протоколів [1], забороною функціонування системи в обхід підсистем захисту та розмежуванням доступу [2]. Найбільш трудомісткими з точки зору обчислювальної реалізації протоколів мережового обміну є виконання операцій модулярної арифметики, зокрема, модулярного експоненціювання, над багаторозрядними числами [3], розрядність яких значно перевищує розрядність процесорів. Особливо це стосується асиметричної криптографії [4], наприклад, криптосистем RSA, Ель-Гамаля, протоколів обміну ключем та електронного цифрового підпису [5] тощо.

Складність використання протоколів обміну ключами та рівень захищеності прямо пропорційна розрядності чисел, якими вони оперують. На сьогоднішній день для досягнення прийнятного рівня захищеності вказаних криптографічних застосувань необхідна розрядність чисел становить щонайменше 1024 біт [6] з перспективою її зростання в найближчі роки до 2048 та 4096 біт. Це неминуче призведе до зменшення швидкодії виконання алгоритмів модулярного експоненціювання та виникнення небезпеки переповнення розрядної сітки. З особливою гостротою ця проблема стоїть для малорозрядних вбудованих мікропроцесорів та мікроконтролерів [7]. Тому постає

задача вибору найбільш швидкого алгоритму піднесення до степеня великорозрядних чисел за модулем при використанні комп'ютерів загального та спеціального призначення.

Слід зазначити, що на даний момент розроблено багато методів різної обчислювальної складності для модулярного множення та експоненціювання [3] (стандартний, бінарний, Монтгомері, Карацуби-Офмана тощо). Однак теоретичні дослідження обчислювальної складності не завжди показують реальну картину щодо швидкодії обчислень, оскільки не враховують усіх факторів, які впливають на роботу комп'ютера, зокрема, звертань до пам'яті, особливостей завантаженості процесора, його розрядності та розпаралелення обчислень тощо.

Тому *метою* нашої роботи є знаходження, порівняння та аналіз часу модулярного експоненціювання, виконаних за допомогою різних алгоритмів для чисел різної розрядності на комп'ютері із заданими параметрами.

Основна частина

Для досліджень було обрано комп'ютер HP ProBook 4540S, з процесором Intel i5, тактова частота 2,3 GHz, оперативна пам'ять 6 GB, операційна система Linux Mint 17.1, середовище програмування Python 3.4.0. Вибір останнього зумовлений його модульністю та можливістю повторного використання коду [8]. Це є інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формах на всіх основних платформах. В мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована [9]. Безсумнівною її перевагою є можливість роботи з великорозрядними числами (до 4096 біт).

Вибір методів зумовлювався часом модулярного експоненціювання, який для різних методів мав бути приблизно одного порядку, та можливістю виконання операцій над великорозрядними числами (до 4096 біт). В зв'язку з цим не досліджувалися методи прямого піднесення до степеня, при якому вже для 16-бітних чисел відбувалося переповнення розрядної сітки, та множення справа наліво або зліва направо, що вимагало виконання x-1 операцій множення (x – показник степеня) і, відповідно, збільшувався час обчислень. З цієї ж причини не був використаний метод системи залишкових класів (СЗК), коли модуль p є добутком декількох простих співмножників ($p = p_1 \cdot p_2 \cdot \ldots \cdot p_n$), і його потрібно факторизувати, що приводило до істотного збільшення часової складності. Отже, для дослідження були вибрані чотири методи:

- бінарний або метод пониження степеня за допомогою квадратів [3], який описується виразом:

$$N = a^x \bmod p = \left(a^{x-2x_1} a_1^{x_1-2x_2} \ldots a_i^{x_i-2x_{i+1}} \ldots\right) \bmod p = \left(\prod_{i=0}^{[\log_2 x]} a_i^{x_i-2x_{i+1}}\right) \bmod p, \qquad (1)$$

де $a_0 = a$, $x_0 = x$, $a_i = a_{i-1}^2 \bmod p$; $x_i = \left[\frac{x_{i-1}}{2}\right]$;

- метод пониження степеня за допомогою кубів (3-арний) [2], який можна записати аналогічно (1):

$$N = a^x \bmod p = \left(a^{x-3x_1} a_1^{x_1-3x_2} \ldots a_i^{x_i-3x_{i+1}} \ldots\right) \bmod p = \left(\prod_{i=0}^{[\log_3 x]} a_i^{x_i-3x_{i+1}}\right) \bmod p,$$

де $a_0 = a$, $x_0 = x$, $a_i = a_{i-1}^3 \bmod p$; $x_i = [x_{i-1}/3]$;

- використання СЗК [10], коли основа степеня розбивається на залишки від ділення на попарно взаємно прості модулі, далі відбувається пониження степеня бінарним методом і відновлення десяткового запису числа. Всю процедуру можна описати виразом:

$$N = a^x \bmod p = \left(\sum_{i=1}^{k} m_i M_i a_i\right) \bmod p, \qquad (2)$$

де $a_i = (a \bmod p_i)^x \bmod p_i$, $p = \prod_{i=1}^{k} p_i$, $M_i = \frac{p}{p_i}$, $m_i = M_i^{-1} \bmod p_i$, $a_0 = a$, k – кількість модулів. Пошук оберненого елемента відбувається за допомогою алгоритму Евкліда та його наслідку;

- використання модифікованої досконалої форми (МДФ) СЗК [11], згідно якої модулі підібрані таким чином, що $m_i = M_i \bmod p_i = \pm 1$. Це дозволяє уникнути виконання громіздкої операції пошуку оберненого елемента за модулем та множення в (2) на базисні числа m_i, що, відповідно, приводить до зменшення часу обчислень.

Для модулярного піднесення до степеня $a^x \bmod p$ в залежності від розрядності n та кількості одиничок у двійковому записі числа (ваги Хемінга) параметри a, x та p визначалися таким чином: $a = r(n-3, \Delta)$, $x = r(n, \Delta)$, де цілочисельна функція

$$r(n, \Delta) = 2^n - 1 - [RND \cdot [\Delta \cdot n/100]],$$

n – розрядність чисел, над якими виконуються операції, $0 < RND < 1$ – випадкова величина, заданий параметр Δ набуває дискретних значень 0, 10, 30, 50 та 80.

Останній вказує, що при $\Delta = 0$ вага Хемінга чисел $a = 2^n - 1$ та $x = 2^{n-3} - 1$ дорівнює їх розрядності. Збільшення цього параметра означає, що Δ% молодших розрядів у двійковому записі a та x може змінитися випадковим чином під дією функції RND.

Для кожного методу піднесення до степеня використовувався один і той самий модуль, який залежить від розрядності числа x і є добутком трьох попарно взаємно простих співмножників $p = p_1 \cdot p_2 \cdot p_3$, що утворюють МДФ СЗК: $p_1 = 2^{\left[\frac{n}{3}\right]+1}$, $p_2 = 2p_1 - 1$, $p_3 = 2p_1 + 1$. Слід відмітити, що модуль p перевищує діапазон обчислень, який визначається відповідною розрядністю n, приблизно у 8-16 разів. Усі розрахунки для кожного випадку проводилися 10 разів і визначався середній час виконання модулярного експоненціювання (в мікросекундах). Отримані результати наведені в таблиці 1 (у першому стовпчику 1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – СЗК, 4 – МДФ СЗК).

Як слідує з таблиці, швидкість піднесення до степеня за модулем істотно залежить від методу, яким воно виконується. Так, при малих розрядностях (до 256 включно) методи СЗК та МДФ СЗК істотно поступаються в часі двом іншим (приблизно в 2-4 рази), а найшвидше модулярне експоненціювання виконується методом пониження степеня за допомогою кубів. Це ж стосується випадку, коли $n = 512$ і $\Delta = 0$. Для інших Δ і тієї ж розрядності мінімальний час буде при використанні бінарного методу, а СЗК та МДФ СЗК вже відстають приблизно в 1,5 раза і, починаючи з $n = 1024$ вони дають мінімальний час. На рисунку 1 в логарифмічній шкалі представлено графіки залежності часу виконання операції модулярного

експоненціювання різними методами від розрядності чисел для $\Delta = 0$, тобто для чисел з максимальним значенням ваги Хемінга.

Таблиця 1.

Час виконання операції модулярного експоненціювання різними способами

Розрядність	8	16	32	64	128	256	512	1024	2048	4096
$\Delta = 0$										
1	14	24	55	108	254	746	2251	10727	64628	405196
2	12	23	44	91	200	655	2162	11987	75411	466016
3	58	83	159	299	628	1430	3430	10366	37564	216596
4	62	72	150	266	610	1340	3195	9956	36861	208496
$\Delta = 10$										
1	10	18	39	81	179	487	1955	9452	61290	404806
2	9	17	34	66	148	445	1997	10055	68190	458023
3	43	58	114	211	456	997	2858	9266	34831	199364
4	40	53	105	193	451	969	2563	8417	34413	198145
$\Delta = 30$										
1	10	18	38	78	177	487	1919	9353	60907	403917
2	9	17	35	68	146	441	2038	10097	68163	457405
3	43	61	115	211	458	998	2848	9009	34768	199534
4	40	53	103	193	443	974	2546	8409	34378	197433
$\Delta = 50$										
1	11	18	37	79	177	494	1954	9465	61187	404641
2	8	17	33	67	146	442	1978	10224	68165	457931
3	44	61	113	211	459	1004	2847	9108	34826	199136
4	40	53	103	193	442	965	2553	8443	34361	197780
$\Delta = 80$										
1	10	17	37	78	181	492	1918	9419	60949	404217
2	9	18	33	67	146	444	2015	10074	68148	458093
3	42	61	112	209	458	1001	2856	9086	34738	199704
4	40	51	101	194	440	974	2560	8394	34355	197588

Слід зазначити, що при інших значеннях Δ залежності мають приблизно такий самий характер, тому вони не наведені. З рисунка видно, що графіки для СЗК та МДФ СЗК практично збігаються. Це пов'язано з тим, що при переведенні чисел із СЗК в десяткову систему числення для модулів, які утворюють МДФ СЗК і використовуються в дослідженні, потрібна мала кількість ітерацій при застосуванні алгоритму Евкліда, його наслідку та китайської теореми про залишки.

На рисунку 2 наведено графіки залежності часу виконання модулярного піднесення до степеня різними методами від значення ваги Хемінга Δ при найбільш поширеній на даний час розрядності $n = 1024$.

Як видно з рисунка, всі графіки мають приблизно однаковий характер. Найбільший час затрачається при максимальній вазі Хемінга ($\Delta = 0$). При зменшенні останньої час різко зменшується і надалі залишається майже постійним, здійснюючи невеликі коливання відносно деякого значення.

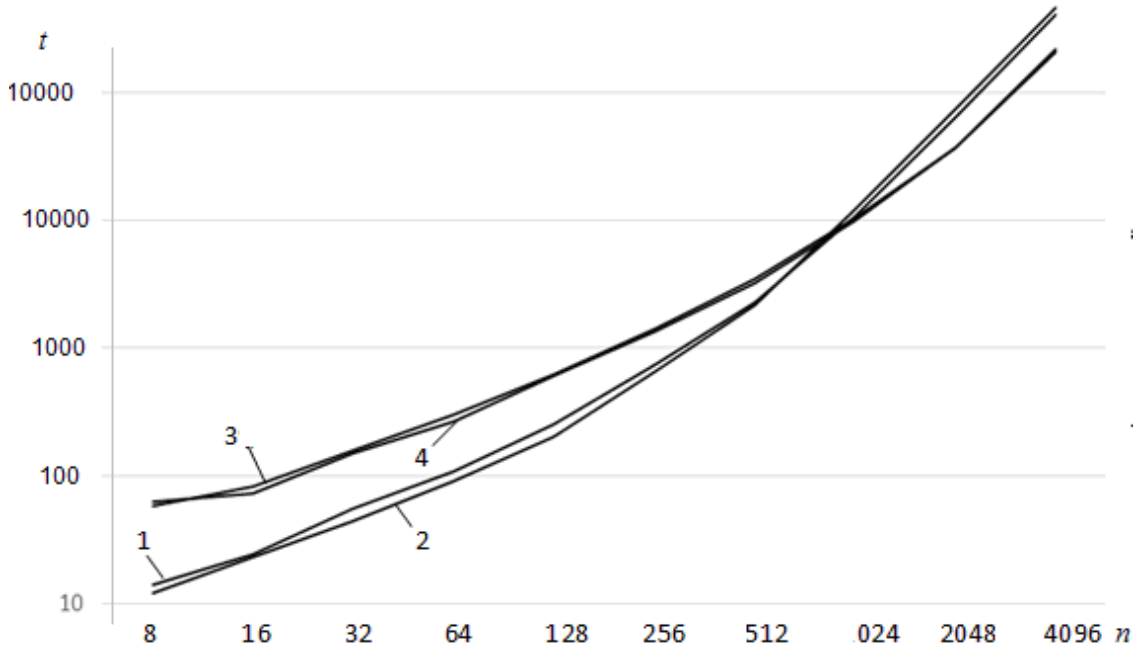

Рис. 1. Графіки залежності часу виконання операції модулярного експоненціювання різними методами від розрядності чисел для $\Delta = 0$ (1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – використання СЗК, 4 – використання МДФ СЗК**)**

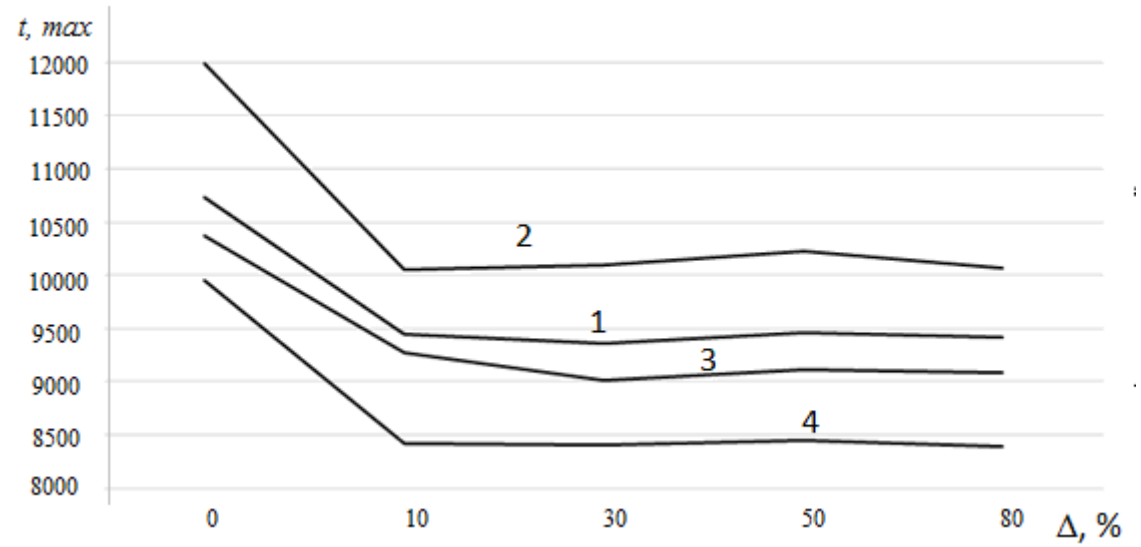

Рис. 2. Графіки залежності часу виконання модулярного піднесення до степеня різними методами від значення ваги Хемінга Δ при $n = 1024$ (1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – використання СЗК, 4 – використання МДФ СЗК)

Висновки

У роботі проведено експериментальне дослідження програмної реалізації різних методів модулярного експоненціювання. Показано, що при малих розрядностях найбільшою швидкодією характеризується метод пониження степенів за допомогою кубів. Починаючи з 1024 біт найменший час для виконання операції модулярного

піднесення до степеня витрачається при застосуванні методу МДФ СЗК. Досліджено часову складність модулярного експоненціювання в залежності від ваги Хемінга. Встановлено, що найбільший час затрачається, коли вага Хемінга максимальна. При її зменшенні час виконання різко зменшується і надалі залишається майже постійним.

Список літератури

1. Горбенко, І. Д. Захист інформації в інформаційно- телекомунікаційних системах / І.Д. Горбенко, Т.О.Гриненко. - Харків: ХНУРЕ, 2004. - 368 с.
2. Норткатт, С. Обнаружение вторжений в сеть / С. Норткатт, Д. Новак. – М.: Лори, 2001. – 384 с.
3. Задірака, В.К. Комп'ютерна арифметика багаторозрядних чисел. / В.К. Задірака, О.С. Олексюк. – К.: КИ, 2003. – 264 с.
4. Задірака, В.К. Комп'ютерна криптологія / В.К. Задірака, О.С. Олексюк. – Тернопіль, Київ, 2002. – 504 с.
5. Вербіцький, О.В. Вступ до криптології / О.В. Вербіцький. – Львів: ВНТЛ, 1998. – 247 с.
6. Петренко, А.Б. Аналіз часових атак на апаратний шифратор персонального засобу криптографічного захисту інформації «Шипка» / А.Б. Петренко, О.С. Шматок, С.А. Шматок, Є.О. Агеєнко // Наукоємні технології. - 2014. - № 2 (22). - С. 187-191.
7. Schindler, W. A timing attack against rash with the chinese remainder theorem / W. Schindler // Cryptographic Hardware and Embedded System. - 2000. - № 2. - P. 109–124.
8. Stewar,t, J. Python for Scientists / J. Stewart. - Cambridge: Cambridge University Press, 2014 - 230 p.
9. Downey, A. How to Think Like a Computer Scientist. Learning with Python / A. Downey, J. Elkner, C. Meyers. - Wellesley: Green Tea Press, 2012. - 243 p.
10. Николайчук, Я.М. Теорія джерел інформації / Я.М. Николайчук. – Тернопіль: ТзОВ «Терно–граф», 2010. – 536 с.
11. Касянчук, М. Алгоритм знаходження системи модулів модифікованої досконалої форми системи залишкових класів / М.М. Касянчук, Я.М. Николайчук, І.З. Якименко, Л.М. Тимошенко, Т.М. Долинюк // Матеріали Міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології». - Одеса, 26 — 30 травня 2014 р. – С. 115-116.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ МЕТОДОВ МОДУЛЯРНОГО ЭКСПОНЕНЦИИРОВАНИЯ

М.Н. Касянчук, И.З. Якименко, Т.Н. Долынюк, Н.А. Рендзеняк

Тернопольский национальный экономический университет,
ул. Львовская, 11, Тернополь, 46020, Украина; e-mail: kasyanchuk@ukr.net, jiz@tneu.edu.ua

Выполнение арифметических операций над многоразрядными числами является весьма важной задачей современной теории чисел и асимметричной криптографии. В работе проведено экспериментальное исследование временной сложности модулярного экспоненциирования с использованием среды программирования Python 3.4.0 различными методами: понижение степени посредством выделения квадратов (бинарный), кубов (3-арный), системы остаточных классов и ее модифицированной совершенной формы. Операции выполнялись над числами различной разрядности и с разным весом Хемминга. Показано, что при малых разрядностях наибольшей скоростью характеризуется метод понижение степеней с помощью кубов. Начиная с 1024 бит наименьшее время для выполнения операции модулярного возведения в степень расходуется при применении метода модифицированной совершенной формы системы остаточных классов. При исследовании временной сложности модулярного экспоненциирования в зависимости от веса Хемминга установлено, что наибольшее время затрачивается, когда вес Хемминга максимален. При его уменьшении время выполнения резко уменьшается и в дальнейшем остается почти постоянным.

Ключевые слова: асимметричная криптография, модульное экспоненциирование, понижение степени, модифицированная совершенная форма системы остаточных классов, временная сложность, вес Хемминга.

EXPERIMENTAL STUDY OF PROGRAM IMPLEMENTATION OF METHODS OF MODULAR EXPONENTIAL

M.M. Kasianchuk, I.Z. Yakymenko, T.M. Dolynyuk, N.A. Rendzenyak

Ternopil National Economic University,
11, Lvivska Str., Ternopil, 46020, Ukraine; e-mail: kasyanchuk@ukr.net, jiz@tneu.edu.ua

The arithmetic operations on multi-digital numbers performing is assigned as the mportant task of modern numerical theory and asymmetric cryptography. Present work was devoted to experimental study of time complexity of modular exponential with using of Python 3.4.0 programming platform by various methods, among them depleted of degree using selection of squares (binary), cubes (tertiary) residual classes system and its modified perfect shape. The operations were performed on different bit numbers and with different Hemets' weight. It is shown that at low bit the greatest rate of reduction method is characterized by using the powers of cubes. Since 1024 bit the least time to perform of operation of modular exponentiation power was exploited under the application of method of modified form of perfect system of residual classes. Under the evaluation of time complexity of modular exponential depending on Hemets' weight it was shown that most time was spent, when the Hemets'weight is maximal. With its reducing runtime drastically reduced and remained almost constant.

Keywords: asymmetric cryptography, modular exponential, degree depleted, residual classes system and its modified perfect shape, time complexity, Hemets' weight.