УДК 581.3

Informatics and Mathematical Methods in Simulation
Vol. 7 (2017), No. 3, pp. 178-186

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДІВ ПОШУКУ ОБЕРНЕНОГО ЕЛЕМЕНТА ЗА МОДУЛЕМ

М.М. Касянчук, І.З. Якименко, С.В. Івасьєв, О.В. Момотюк

Тернопільський національний економічний університет,
вул. Львівська, 11, Тернопіль, 46020, Україна; e-mail: kasyanchuk@ukr.net , iyakymenko@ukr.net, stepan.ivasiev@gmail.com, momotjuk98@gmail.com

Знаходження мультиплікативного оберненого елемента за модулем дуже часто є необхідною умовою для розв'язування багатьох задач сучасної теорії чисел, обчислювальної та прикладної математики, асиметричної криптографії, зокрема, криптосистем RSA та Ель-Гамаля. У роботі проведено експериментальне дослідження часових характеристик програмної реалізації пошуку оберненого елемента за модулем на основі класичного методу розширеного алгоритму Евкліда та запропонованих методів додавання модуля та додавання залишку із застосуванням мови програмування високого рівня C++. Для дослідження використовувалися числа різної розрядності. Показано, що в переважній більшості розглянутих випадків метод додавання модуля характеризується більш високою швидкодією в порівнянні з двома іншими. Представлено графічні залежності середнього часу пошуку оберненого елемента різними методами від розрядності вибраних чисел. Для нівелювання випадкових впливів на час роботи усі обчислення повторювалися 100 разів. Запропоновані методи ефективно можна використовувати для пошуку оберненого елемента за модулем.

Ключові слова: алгоритм Евкліда, обернений елемент за модулем, асиметрична криптографія, додавання модуля, додавання залишку, середній час, часові характеристики.

Вступ

Операція пошуку мультиплікативного оберненого елемента за модулем на даний час є однією з найважливіших і одночасно найбільш обчислювально складних в сучасній теорії чисел [1-2]. Поширеність цієї операції зумовлена її застосуванням в сучасній асиметричній криптографії (криптосистеми RSA, Ель-Гамаля [3], шифрування на основі математичного апарату еліптичних кривих [4], протоколи електронного цифрового підпису та обміну ключами [5] тощо), системі залишкових класів [6-7] та її різних модифікаціях [8-9], кодуванні даних на основі модулярної арифметики [10], інших застосуваннях прикладної та дискретної математики. Тому задача дослідження програмної реалізації різних методів пошуку мультиплікативно оберненого елемента за модулем є надзвичайно актуальною.

Мультиплікативно оберненим елементом до числа a за модулем n називається таке число b, для якого виконується рівність $a-b \bmod n=1$, тобто $b=a^{-1} \bmod n$. Натуральні числа a та n при цьому повинні бути взаємно простими. В [11] детально описані методи пошуку оберненого елемента, з яких найбільш поширеними є такі:

1. Перебором всіх можливих варіантів;
2. За допомогою теореми Ейлера;
3. На основі розширеного алгоритму Евкліда.

Перший метод характеризується високою обчислювальною складністю, оскільки повний перебір вимагає значних часових затрат.

При використанні теореми Ейлера $a^{\varphi(n)} \bmod n = 1$ отримується $a^{\varphi(n)-1} \bmod n = a^{-1} \bmod n$ [12]. Це передбачає виконання операції модулярного експоненціювання [13-14], що може привести до переповнення розрядної сітки процесора і ускладнює пошук оберненого елемента для багаторозрядних чисел.

Найбільш поширеним є метод пошуку мультиплікативного оберненого елемента за модулем на основі розширеного алгоритму Евкліда [15]: спочатку потрібно записати прямий алгоритм Евкліда, згідно якого при $n>a=r_0$, де n і a –цілі числа, виконується така послідовність операцій:

$$
\begin{aligned}
& n = r_0 \cdot q_1 + r_1, \;\; q_1 = a, \;\; 0 \le r_1 < r_0; \\
& r_0 = r_1 \cdot q_2 + r_2, \;\; 0 \le r_2 < r_1; \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
& r_{n-3} = r_{n-2} \cdot q_{n-1} + r_{n-1}, \;\; 0 \le r_{n-1} < r_{n-2}; \\
& r_{n-2} = r_{n-1} \cdot q_n + r_n, \;\; 0 \le r_n < r_{n-1}; \\
& r_{n-1} = r_n \cdot q_{n+1} + 0.
\end{aligned}
$$

Оскільки a і n є взаємно простими, то r_n=1. Розширений алгоритм Евкліда передбачає зворотній порядок проходження прямого алгоритму:

$$
\begin{aligned}
& 1 = r_n = r_{n-2} - q_n r_{n-1} = r_{n-2} - q_n\left(r_{n-3} - q_{n-1} r_{n-2}\right) = r_{n-2} - q_n r_{n-3} + q_n q_{n-1} r_{n-2} = -q_n r_{n-3} + \\
& + \left(1 + q_n q_{n-1}\right) r_{n-2} = -q_n r_{n-3} + \left(1 + q_n q_{n-1}\right)\left(r_{n-4} - q_{n-2} r_{n-3}\right) = \ldots
\end{aligned}
$$

Описана процедура продовжується до тих пір, поки не буде отриманий вираз $v \cdot n + t \cdot r_0 = 1$, де величина $b = t \bmod n = a^{-1} \bmod n$ і буде шуканим оберненим елементом. В таблиці 1 наведено приклад пошуку оберненого елемента на основі розширеного алгоритму Евкліда.

Таблиця 1.

Пошук оберненого елемента $41^{-1} \bmod 157$ на основі розширеного алгоритму Евкліда

Алгоритм Евкліда	Розширений алгоритм Евкліда
157=41·3+34 41=34·1+7 34=7·4+6 7=6·1+1	1=7-1·6=7-1·(34-4·7)=-1·34+5·7=-1·34+5·(41-1·34)=5·41-6·34= =5·41-6·(157-3·41)=-6·157+23·41; $41^{-1} \bmod 157 = 23$

Даний метод характеризується великою кількістю ділень з остачею, перемножень і підстановок, хоча він володіє найменшою часовою складністю в порівнянні з іншими двома, наведеними вище.

Метою даної роботи є розробка нових методів пошуку оберненого елемента за модулем без виконання обчислювально складних операцій множення та ділення з остачею, а також експериментальне порівняння програмної реалізації пошуку оберненого елемента за модулем на основі розширеного алгоритму Евкліда та запропонованими методами.

Основна частина

З теорії чисел відомо [16], що вираз $a \cdot b \bmod n=1$ можна переписати таким чином: $a \cdot b=k \cdot n+1$, де k – деяке ціле число. Звідси слідує, що для пошуку оберненого елемента необхідно до модуля додати 1 і перевірити, чи ділиться націло отримане число на a. Якщо не ділиться, то далі до отриманого числа потрібно послідовно додавати модуль до тих пір, поки результатом ділення не буде ціле число. Математично це записується так:

$$n_0 = n+1;\ \ b_0 = (n+1)/a;$$
$$n_1 = 2 \cdot n+1;\ \ b_1 = (2 \cdot n+1)/a;$$
$$\ldots.$$
$$n_i = (i+1) \cdot n+1;\ \ b_i = ((i+1) \cdot n+1)/a;\ b_i \in Z.$$

Блок-схема алгоритму пошуку оберненого елемента на основі додавання модуля представлена на рисунку 1, а в таблиці 2 – приклад його застосування.

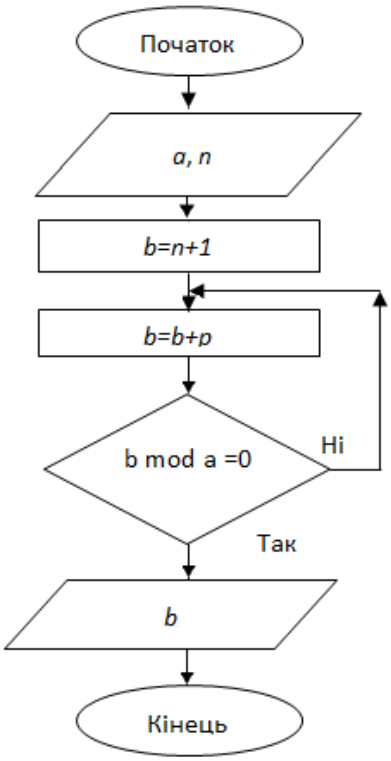


Рис. 1. Блок-схема пошуку оберненого елемента на основі додавання модуля

Таблиця 2.

Пошук оберненого елемента $41^{-1} \bmod 157$ на основі додавання модуля

i	0	1	2	3	4	5
n_i	158	315	472	629	786	943
b_i	3,85…	7,68…	11,51…	15,34…	19,17…	23

Отже, $41^{-1} \bmod 157=23$. Результат отриманий без використання громіздких операцій ділення з остачею та множення.

Для зменшення чисел, які використовуються в даній процедурі, можна додавати не модуль (дослідження часових характеристик апаратної реалізації цього методу

проведено, зокрема, в [17]), а залишок $n_{00} = n \bmod a$ до тих пір, поки остача від ділення отриманого результату на число a не буде дорівнювати 0. Математичний запис даного алгоритму матиме такий вигляд:

$$b_{01} = (n_{00} + 1) \bmod a;$$
$$b_{11} = (b_{01} + n_{00}) \bmod a;$$
$$b_{21} = (b_{11} + n_{00}) \bmod a;$$
$$\ldots.$$
$$b_{i1} = (b_{i-1\ 1} + n_{00}) \bmod a = 0.$$

Обернений елемент шукається за формулою $b = a^{-1} \bmod n = (i+1)n/a$.

Блок-схема алгоритму пошуку оберненого елемента на основі додавання залишку представлена на рисунку 2, а в таблиці 3 – приклад його застосування з врахуванням, що n_{00}=157 mod 41=34.

Отже, 41^{-1}mod 157=(6·157+1)/41=23. Кількість ітерацій даного алгоритму така ж сама, як і в попередньому, однак переважна більшість операцій виконується над значно меншими числами.

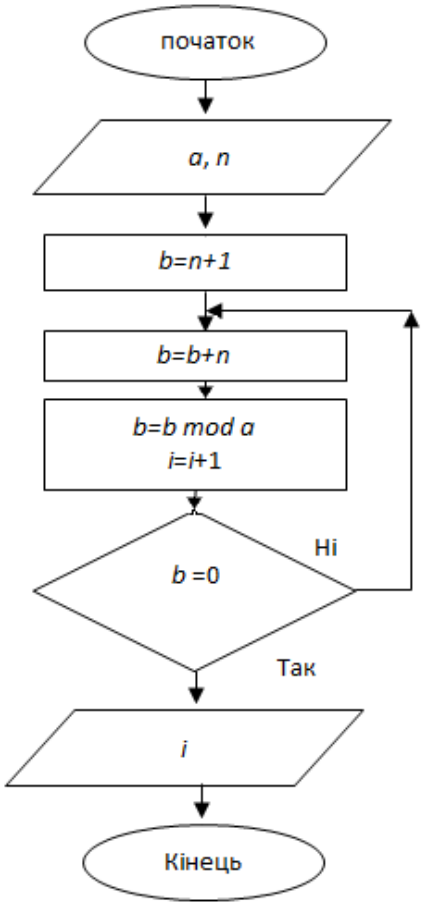


Рис. 2. Блок-схема пошуку оберненого елемента на основі додавання залишку

Таблиця 3.

Пошук оберненого елемента 41^{-1}mod 157 на основі додавання залишку

i	0	1	2	3	4	5
b_{i1}	35	28	21	14	7	0

На відміну від розширеного алгоритму Евкліда, запропоновані методи дозволяють розпаралелити процес пошуку оберненого елемента на декілька потоків. Початок обчислень в кожному потоці для методів додавання модуля та залишку відповідно визначається з таких формул:

$$N_0 = \left(\left[\frac{(j-1)a}{z} \right] + 1 \right) n + 1; \quad (1)$$

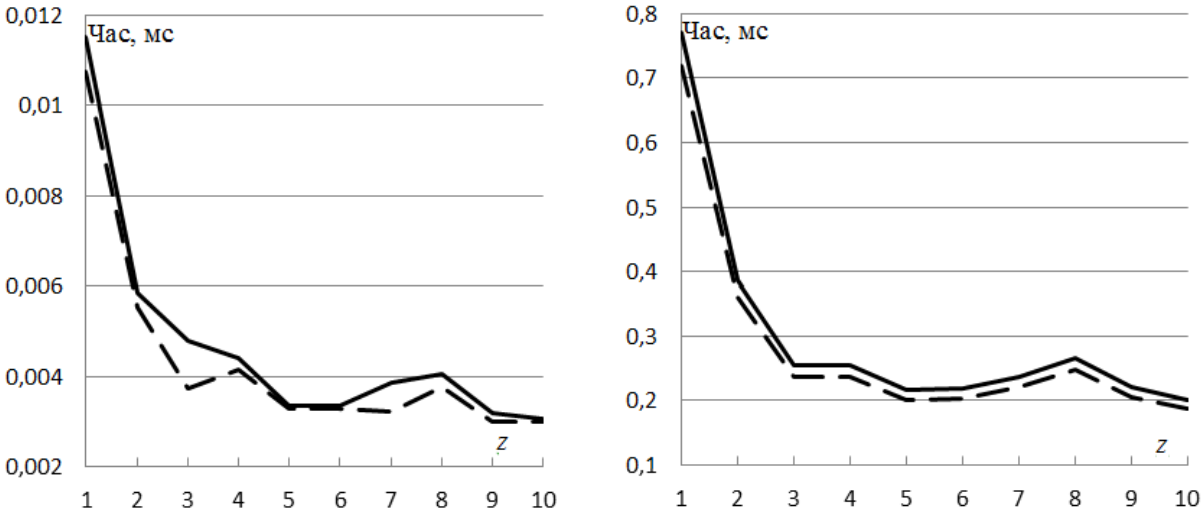
$$N_1 = \left(\left(\left[\frac{(j-1)a}{z} \right] + 1 \right) n_{00} + 1 \right) \bmod a \quad (2)$$

де j - номер потоку, z - кількість потоків. Максимальна кількість ітерацій в кожному потоці становитиме $a/z+1$.

Для експериментальних досліджень було обрано портативний комп'ютер Lenovo B50-70 з процесором Intel Pentium 3558U (1.7 ГГц). Об'єм оперативної пам'яті в пристрої становив 4 GB. При проектуванні програмного комплексу, що забезпечував обчислення, було обрано мову програмування високого рівня C++, яка дозволяє трансформувати коди під різні архітектури та операційні системи.

Даний комп'ютер не дозволяє розпаралелювати обчислення на будь-яку кількість потоків, тому після визначення меж усі обчислення виконувалися послідовно, а час пошуку оберненого елемента за модулем вибирався в тому потоці, де знаходився шуканий результат.

На рисунку 3 показано графічну залежність середнього часу пошуку оберненого елемента на основі додавання модуля (штрихова лінія) та залишку (суцільна лінія) від кількості паралельних потоків z при n=1021 та n=65521 (рис. 3). Останні є найбільшими простими десяти- та шістнадцятирозрядними числами. Число a пробігає значення від 2 до n-1 з кроком 1.



а б

Рис. 3. Графічна залежність середнього часу пошуку оберненого елемента на основі додавання модуля (штрихова лінія) та залишку (суцільна лінія) від кількості паралельних потоків при: а – n=1021; б – n=65521

З рисунка 3 видно, що в обох випадках метод додавання модуля володіє більшою швидкодією, ніж додавання залишку. Графіки мають приблизно однаковий характер. При розпаралелюванні на два та три потоки час обчислень різко зменшується

порівняно з послідовним виконанням усіх операцій (z=1). При подальшому збільшенні z інтенсивність зменшення часу спадає. Незначне збільшення часу спостерігається, коли z=7, 8, і надалі графік повільно спадає. Тому для подальших досліджень було вибрано z=6.

На рисунку 4 показано графічну залежність часу пошуку оберненого елемента на основі додавання модуля (штрихова лінія) та залишку (суцільна лінія) при z=6 та n=65521, а в таблиці 4 представлений фрагмент з частиною отриманих результатів.

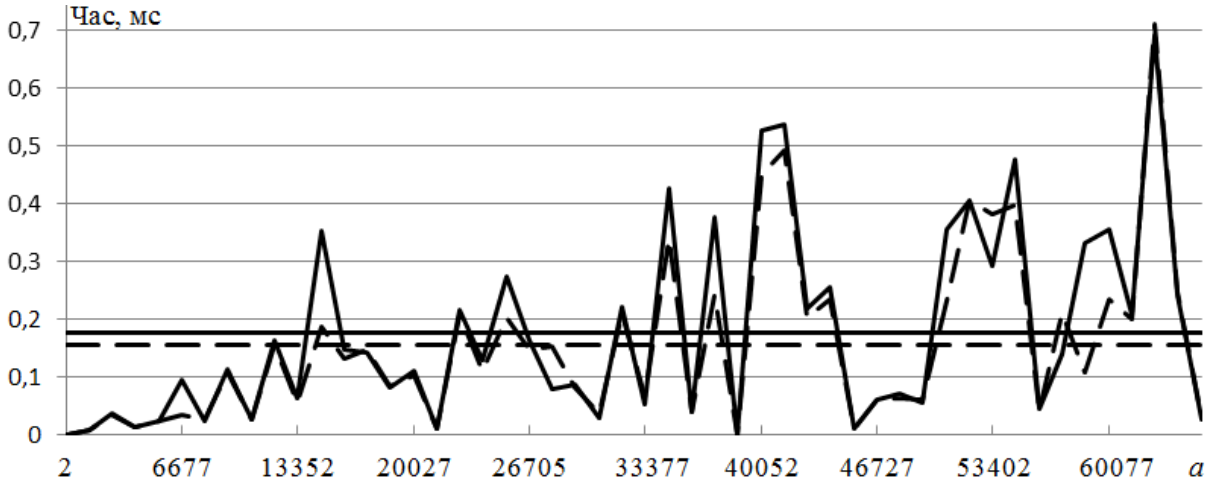


Рис. 4. Графічна залежність часу пошуку обернених елементів на основі додавання модуля (штрихова лінія) та залишку (суцільна лінія) при z=6 та n=65521 та середні часи обчислень (штрихова та суцільна прямі відповідно)

Таблиця 4.

Фрагмент з частиною отриманих результатів

a	b	Час пошуку, мс	
		Додавання модуля	Додавання залишку
17357	55744	0,146275	0,142446
18692	49968	0,08318	0,08077
20027	29906	0,10303	0,111585
21362	22510	0,009311	0,009736
22697	35106	0,207195	0,216458
24032	50643	0,109552	0,129591
25367	33547	0,202232	0,272227
26702	30557	0,146747	0,164092
28037	25648	0,152182	0,078502

Число a змінюється від 2 до 65417 з кроком 1335, який був вибраний з міркувань того, щоб розрахунки оберненого елемента здійснювалися для 50-ти значень. Крім того, наведено середній час обчислень обома методами (штрихова пряма – додавання модуля, суцільна пряма – додавання залишку). Графіки 1 і 2 носять осцилюючий характер, однак загальний тренд вказує на збільшення часу із ростом числа a. В переважній більшості випадків метод додавання модуля є швидший (усереднений час обчислень розглянутих прикладів становить 0,15499 мс), ніж метод додавання залишку (середній час – 0,176144 мс), в 1,136 рази.

На рисунку 5 показано графічну залежність середнього часу пошуку оберненого елемента на основі розширеного алгоритму Евкліда (крива 1) та запропонованих

методів додавання модуля (штрихові лінії 3, 5) та залишку (суцільні лінії 2, 4) при z=1 (криві 2, 3) і z=6 (криві 3,5) від розрядності модуля. Модуль n вибирався найбільшим простим, але меншим за 2^k, де k змінювалося від 10 до 19,5 з кроком 0,5. Число a набувало 1000 різних значень. Крок його зміни вибирався таким, щоб охопити весь діапазон від 1 до n.

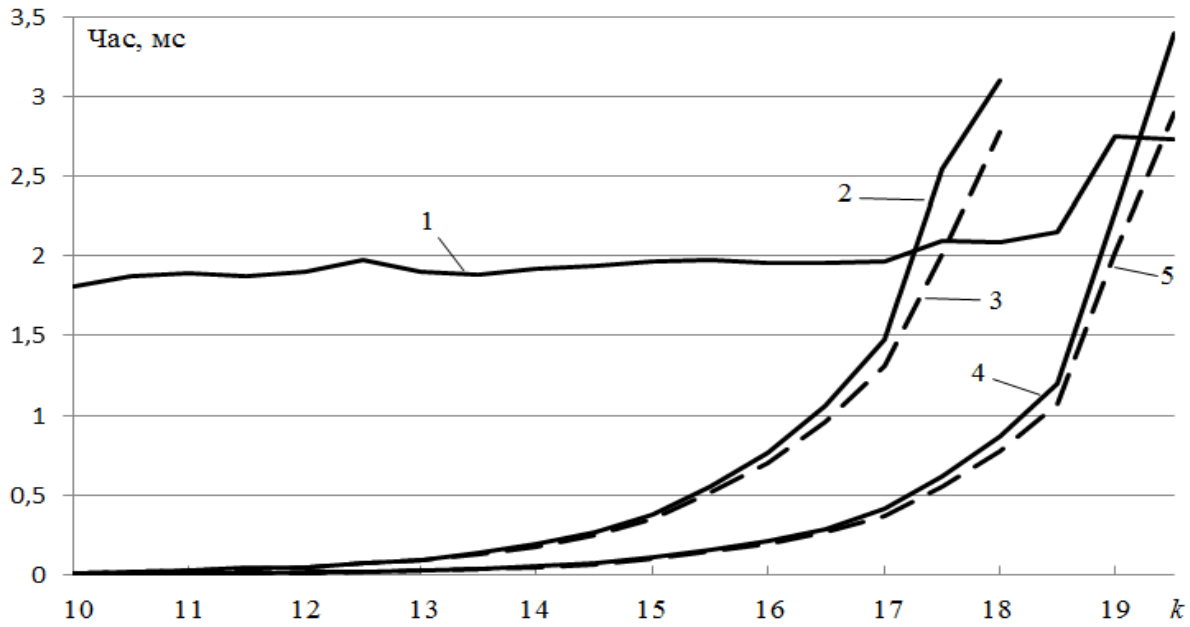


Рис. 5. Графічна залежність часу пошуку оберненого елемента від розрядності модуля на основі: 1 – розширеного алгоритму Евкліда; 2 – запропонованих методів додавання модуля при z=1; 3 – запропонованих методів додавання модуля при z=6 ; 4 – залишку при z=1; 5 – залишку при z=6 ;

Як видно з рисунка 5, при використанні алгоритму Евкліда із збільшенням розрядності чисел час збільшується дуже повільно. При k=19 час різко зростає і далі знову наступає практично горизонтальна ділянка графіка. В обох запропонованих методах середній час збільшується приблизно параболічно, причому метод додавання модуля має більшу швидкодію. При z=1, тобто без розпаралелення, алгоритм Евкліда випереджає запропоновані алгоритми при k=17,5, а для z=6 – при k=19,5. Це вказує на необхідність розпаралелення процесу обчислень для зменшення часу пошуку оберненого елемента за модулем.

Для нівелювання випадкових впливів на час роботи усі експерименти повторювалися 100 разів.

Висновки

У роботі проведено експериментальне дослідження часових характеристик програмної реалізації пошуку оберненого елемента за модулем на основі класичного методу розширеного алгоритму Евкліда та запропонованих методів додавання модуля та додавання залишку із застосуванням мови програмування високого рівня C++. Показано, що в переважній більшості розглянутих випадків метод додавання модуля характеризується більш високою швидкодією в порівнянні з двома іншими. Представлено графічні залежності середнього часу пошуку оберненого елемента різними методами від розрядності вибраних чисел. Запропоновані методи ефективно можна використовувати для розпаралелення процесу обчислень та зменшення часу пошуку оберненого елемента за модулем.

Список літератури

1. Lorencz, R. New Algorithm for Classical Modular Inverse / R.Lorencz // Cryptographic Hardware and Embedded Systems. International Workshop, 2002. – Pp. 57-70.
2. Parthasarathy, S. Multiplicative inverse in mod(m) / S. Parthasarathy // Algologic Technical Report, 2012. – №1. – Pp. 1-3.
3. Kasyanchuk, M.M. Algorithms theory of RSA and El Gamal in differentiated notation of Rademacher-Krestenson basis / M.M. Kasyanchuk, I.Z. Yakymenko, O.I. Volynskiy, I.R. Pituh // Reports of Khmelnitsky National University. Technical sciences, 2011. – №3. – Pp. 265-273.
4. Hankerson, D. Guide to EllipticCurve Cryptography / D. Hankerson, A. Menezes, S. Vanstone.- New York, USA: Springer-Verlag, 2004. – 311 p.
5. Stallings, W._Cryptography and Network Security: Principles and Practice / W. Stallings. - New York, USA: Prentice Hall Press Upper Saddle River, 2010. – 719 p.
6. Omondi, A. Residue number systems: theory and implementation / A. Omondi, B. Premkumar. - London: Imperial College Press, 2007. – 296 p.
7. Kasyanchuk, M.M. Foundations for the Analytical Computation of Coefficients of Basic Numbers of Krestenson's Transformation / M.M. Kasyanchuk, Ya. M. Nykolaychuk, I. Z. Yakymenko // Cybernetics and Systems Analysis, 2014. – Vol 50, №5. – Pp. 649-654.
8. Nykolaychuk, Ya.M. Theoretical Foundations of the Modified Perfect Form of Residue Number System / Ya.M.Nykolaychuk, M.M.Kasianchuk, I.Z.Yakymenko // Cybernetics and Systems Analysis, 2016. – Vol. 52, №2. – Pp. 219-223.
9. Kasianchuk, M. Conception of theoretical bases of the accomplished form of Krestenson's transformation and its practical application / M. Kasianchuk // Proceedings of the 4-th International Conference "Advanced Computer Systems and Networks: Design and Application" (ACSN–2009). – L'viv, Ukraine, 2009. – Pp. 299 - 301.
10. Яцків, В.В. Модифіковані коректуючі коди системи залишкових класів та їх застосування / В.В. Яцків // Інформаційні технології та комп'ютерна інженерія, 2013. – №2. – С. 39-45.
11. Zhengbing, Hu. The Analysis and Investigation of Multiplicative Inverse Searching Methods in the Ring of Integers Modulo M / Hu Zhengbing, I. A. Dychka, M. Onai, A. Bartkoviak // International Journal of Intelligent Systems and Applications (IJISA), 2016. – Vol. 8, №11. – Pp. 9-18.
12. Онай, М.В. Пошук мультиплікативно оберненого елементу у кільці лишків за довільним модулем методами, що грунтуються на модулярному піднесенні до степеня / М.В. Онай, А.Ю. Бартков'як // Матеріали шістнадцятої міжнародної наукової конференції імені академіка Михайла Кравчука. Т.2: Алгебра. Геометрія. Математичний аналіз. – Київ, 2015. – С. 139-141.
13. Kozaczko, D. Vector Module Exponential in the Remaining Classes System / D.Kozaczko, M.Kasianchuk, I.Yakymenko, S.Ivasiev // Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS–2015),1, September, Warsaw, Poland, 2015. – Pp.161-163.
14. Касянчук, М.М. Експериментальне дослідження програмної реалізації методів модулярного експоненціювання / М.М.Касянчук, І.З.Якименко, Т.М.Долинюк, Н.А.Рендзеняк // Інформатика та математичні методи в моделюванні, 2015. – Т.5, №4. – С. 376 -382.
15. Дичка, І.А. Застосування *k*-арного методу Евкліда для пошуку мультплікативно оберненого елемента у кільці лишків за модулем *m* / І.А. Дичка, М.В. Онай, А.Ю. Бартков'як // Матеріали статей П'ятої Міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія». - Івано-Франківськ, 2015. – С. 151-153.
16. Виноградов, И.М. Основы теории чисел / И.М.Виноградов. – Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2003. – 176 с.
17. Rajba, *T.* Research of Time Characteristics of Search Methods of Inverse Element by the Module / T. Rajba, A. Klos-Witkowska, S. Ivasiev, I. Yakymenko, M. Kasianchuk // Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS–2017),1, September, Bucharest, Romania, 2017.– Pp.82 - 85.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ МЕТОДОВ ПОИСКА ОБРАТНОГО ЭЛЕМЕНТА ПО МОДУЛЮ

М.Н. Касянчук, И.З. Якименко, С.В. Ивасьев, О.В. Момотюк

Тернопольский национальный экономический университет,
ул. Львовская, 11, г.Тернополь, 46020, Украина; E-mail: kasyanchuk@ukr.net, jiz@tneu.edu.ua, stepan.ivasiev@gmail.com, momotjuk98@gmail.com

Нахождение мультипликативного обратного элемента по модулю очень часто является необходимым условием для решения многих задач современной теории чисел, вычислительной и прикладной математики, асимметричной криптографии, в частности, криптосистем RSA и Эль-Гамаля. В работе проведено экспериментальное исследование временных характеристик программной реализации поиска обратного элемента по модулю на основе классического метода расширенного алгоритма Евклида и предложенных методов добавления модуля и добавления остатка с использованием языка программирования высокого уровня С++. Для исследования использовались числа разной разрядности. Показано, что в подавляющем большинстве рассмотренных случав метод добавления модуля характеризуется более высоким быстродействием в сравнении с двумя остальными. Представлены графические зависимости среднего времени поиска обратного элемента разными методами от разрядности выбранных чисел. Для нивелирования случайных влияний на время работы все вычисления повторялись 100 раз. Предложенные методы можно эффективно использовать для поиска обратного элемента по модулю.

Ключевые слова: алгоритм Евклида, обратный элемент по модулю, асимметричная криптография, добавление модуля, добавление остатка, среднее время, временные характеристики.

EXPERIMENTAL RESEARCH OF PROGRAM IMPLEMENTATION OF METHODS OF THE INVERSE ELEMENT SEARCH BY MODULE

M.M Kasianchuk, I.Z. Yakymenko, S.V. Ivasiev, O.V. Momotjuk

Ternopil National Economic University,
11, Lvivska Str., Ternopil, 46020, Ukraine; e-mail: kasyanchuk@ukr.net, jiz@tneu.edu.ua, stepan.ivasiev@gmail.com, momotjuk98@gmail.com

Finding a multiplicative inverse element by module is very often a necessary condition for solving many tasks of modern theory of numbers, computational and applied mathematics, asymmetric cryptography, in particular RSA and El Gamal cryptosystems. Experimental study of time characteristics of program implementation for finding the inverse element by module on the basis of the classical method of the extended Euclidean algorithm and the offered methods for adding a module and remainder addition with the use of high-level programming language C ++ was conducted in the work. The numbers of different bit capacity were used for the research. It was shown that in the majority of the considered cases the method of adding a module is characterized with higher performance compared to the other two. Graphic dependencies of average time of the inverse element search by different methods on bit capacity of the selected numbers were presented. All calculations were repeated 100 times for leveling random effects on the operating time. The proposed methods can be effectively used to find the inverse element by module.

Keywords: Euclid's algorithm, inverse element by module, asymmetric cryptography, addition module, remainder addition, average time, time characteristics.

.