УДК 581.3

Informatics and Mathematical Methods in Simulation
Vol. 7 (2017), No. 1-2, pp. 29-36

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СУМІСНОГО ВИКОНАННЯ АЛГОРИТМУ ЕВКЛІДА ТА МНОЖЕННЯ

М.М. Касянчук, І.З. Якименко, І.Р. Паздрій, С.В. Івасьєв

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

Сумісне виконання алгоритму Евкліда та перемноження двох багаторозрядних чисел є досить важливою задачею сучасної теорії чисел, обчислювальної математики та асиметричної криптографії, зокрема, криптосистеми Рабіна. У роботі проведено експериментальне дослідження часових характеристик програмної реалізації вказаних операцій класичним та запропонованим методами із застосуванням мови програмування високого рівня C++. У запропонованому методі передбачено використання проміжних результатів алгоритму Евкліда та звертання до наявної у пам'яті комп'ютера таблиці квадратів. Для дослідження використовувалися числа різної розрядності. Показано, що в переважній більшості розглянутих випадків запропонований метод характеризується більш високою швидкодією, середній час виконання операцій зменшується приблизно в 1.3 рази. Для нівелювання випадкових впливів на час роботи усі обчислення повторювалися 5000 разів. Запропонований метод ефективно можна використовувати для сумісного виконання алгоритму Евкліда та перемноження двох багаторозрядних чисел.

Ключові слова: алгоритм Евкліда, множення, асиметрична криптографія, прості числа, середній час, часові характеристики, розрядність чисел

Вступ

На даний час виконання арифметичних операцій над багаторозрядними числами дуже широко використовується в різних галузях науки і техніки, зокрема, при розв'язуванні задач обчислювальної, прикладної та дискретної математики [1]. Кожен з відповідних алгоритмів має свою область ефективного використання залежно від розрядності, моделі обчислень, мови програмування, апаратної або програмної реалізації. Особливо це стосується проблем криптографії [2], де найбільш поширеними є операції множення [3], піднесення до степеня [4], пошуку найбільшого спільного дільника (НСД), використання китайської теореми про залишки [5] тощо. Поєднання перших двох з рештою операцій вимагає строго послідовної реалізації, що істотно зменшує швидкодію обчислювальних систем. Прикладом може бути криптосистема Рабіна [6], де для застосування китайської теореми про залишки використовується алгоритм Евкліда, а для формування відкритого ключа ті ж самі числа необхідно перемножити. Тому досить гостро постає питання про можливість розпаралелення виконання подібних операцій [7, 8], зокрема, застосування системи залишкових класів [9, 10], її досконалої [11, 12] та модифікованої досконалої форм [13, 14].

Сучасний математичний запис алгоритму Евкліда має такий вигляд [15]: для будь-якого $a > b = r_0$, де a і b – цілі числа, виконується система рівнянь

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

НСД (a, b), обчислений за допомогою алгоритму Евкліда зі складністю $\underline{O}\left(2\log_2 \max(a,b)\right)$ [16], дорівнюватиме r_n, тобто останньому ненульовому члену послідовності r_i. Далі відбувається перемноження чисел a і b. Для множення розроблено багато алгоритмів (стандартний, бінарний, Монтгомері, Карацуби-Офмана тощо), які характеризуються різною обчислювальною складністю [17].

Однак теоретичні дослідження 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* роботи є визначення, порівняння та аналіз часу сумісного виконання алгоритму Евкліда та перемноження двох багаторозрядних чисел, виконаних за допомогою класичного та запропонованих методів для чисел різної розрядності на комп'ютері із заданими параметрами.

Основна частина

У криптосистемі Рабіна задані числа a і b є простими, тому відомо, що в такому випадку НСД $(a,b) = r_n = 1$. На відміну від виконання алгоритму Евкліда з наступним перемноженням цих самих чисел пропонується, згідно (1), у множенні використати проміжні та кінцеві результати алгоритму Евкліда таким чином:

$$a \cdot b = r_0^2 q_1 + r_1^2 q_2 + r_2^2 q_3 + \ldots + r_{n-1}^2 q_n + r_n. \tag{2}$$

Слід відмітити, що кількість доданків у (2) відповідає кількості кроків у алгоритмі Евкліда. Хоча даний метод передбачає виконання більшої кількості арифметичних операцій, однак вони виконуватимуться над числами меншої розрядності. Суттєвим кроком для підвищення швидкодії буде наявність таблиці квадратів у пам'яті комп'ютера, хоча це приводить до збільшення використання ресурсів обчислювальної системи. Блок-схема алгоритму наведена на рис. 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++. Для роботи з багаторозрядними числами використовувалась бібліотека спеціального призначення, написана А. Ленстрою. Це дозволило забезпечити гнучкі можливості роботи з багаторозрядними числами, розмір яких залежить лише від доступних ресурсів системи. Обрана мова програмування та кросплатформенність спецбібліотеки дозволяють трансформувати коди під різні архітектури та операційні системи.

Найбільш розповсюдженим методом програмної реалізації алгоритму Евкліда є послідовне віднімання меншого числа від більшого, поки різниця не стане меншою від'ємника. Тоді цю ж саму процедуру потрібно виконати з від'ємником та різницею.

Процес віднімання буде тривати до тих пір, поки від'ємник та різниця не стануть однакові.

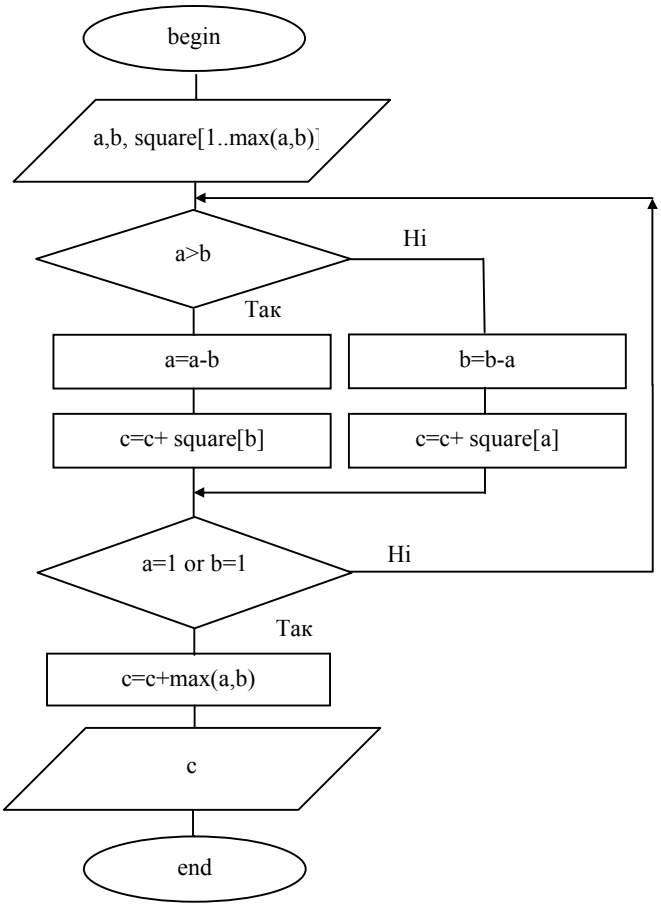

Рис. 1. Блок-схема запропонованого алгоритму

В табл. 1 представлено час сумісного виконання алгоритму Евкліда та перемноження чисел класичним (t_1) та запропонованим (t_2) методами для b=2, 3, …, 70 при $a = 71$, а на рис. 2 – відповідна графічна залежність (t_1 та $t_{1\text{сер}}$ – пунктирна лінія, t_2 та $t_{2\text{сер}}$ – суцільна).

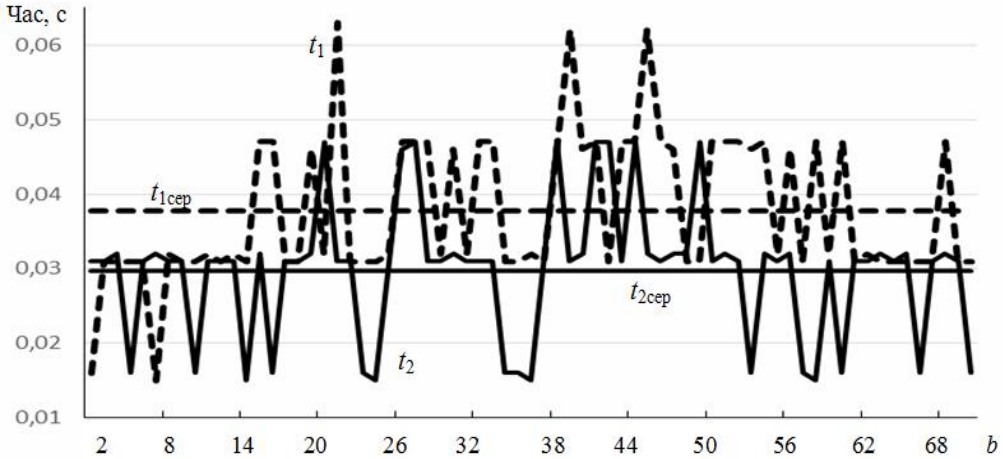

Рис. 2. Графічна залежність часу сумісного виконання алгоритму Евкліда та перемноження чисел класичним (t_1) та запропонованим (t_2) методами

Таблиця 1.

Час сумісного виконання алгоритму Евкліда та перемноження чисел класичним (t_1) та запропонованим (t_2) методами

b	2	3	4	5	6	7	8	9	10	11	12
t_1, с	0.016	0.031	0.031	0.031	0.031	0.015	0.032	0.031	0.031	0.032	0.031
t_2, с	0.031	0.031	0.032	0.016	0.031	0.032	0.031	0.031	0.016	0.031	0.031
b	13	14	15	16	17	18	19	20	21	22	23
t_1, с	0.032	0.031	0.047	0.047	0.031	0.031	0.046	0.032	0.063	0.031	0.031
t_2, с	0.031	0.015	0.032	0.016	0.031	0.031	0.032	0.047	0.031	0.031	0.016
b	24	25	26	27	28	29	30	31	32	33	34
t_1, с	0.031	0.032	0.047	0.047	0.047	0.032	0.046	0.032	0.047	0.047	0.031
t_2, с	0.015	0.031	0.046	0.047	0.031	0.031	0.032	0.031	0.031	0.031	0.016
b	35	36	37	38	39	40	41	42	43	44	45
t_1, с	0.031	0.032	0.031	0.047	0.062	0.046	0.047	0.031	0.047	0.047	0.062
t_2, с	0.016	0.015	0.032	0.047	0.031	0.032	0.047	0.047	0.031	0.047	0.032
b	46	47	48	49	50	51	52	53	54	55	56
t_1, с	0.047	0.046	0.031	0.031	0.047	0.047	0.047	0.046	0.047	0.032	0.046
t_2, с	0.031	0.032	0.032	0.047	0.031	0.032	0.031	0.016	0.032	0.031	0.032
b	57	58	59	60	61	62	63	64	65	66	67
t_1, с	0.031	0.047	0.032	0.047	0.031	0.032	0.031	0.031	0.031	0.031	0.031
t_2, с	0.016	0.015	0.031	0.016	0.031	0.031	0.032	0.031	0.032	0.016	0.031
b	68	69	70								
t_1, с	0.047	0.031	0.031	Середній час: $t_{1\text{сер}}$=0.037783 с.							
t_2, с	0.032	0.031	0.016	Середній час: $t_{2\text{сер}}$=0.029754 с.							

Графіки (рис. 2) носять осцилюючий характер, що пояснюється різною кількістю кроків у алгоритмі Евкліда для різних чисел. В 44 випадках із 69, що становить 64%, обчислення запропонованим методом виконуються швидше, у 10 (14%) – повільніше, в 15 випадках (22%) час виконання обома методами однаковий. Середні значення часу становлять відповідно $t_{1\text{сер}}$=0.037783 с та $t_{2\text{сер}}$=0.029754 с, що також представлено на графіку. Отже, швидкодія збільшилася в середньому в 1.27 рази.

В табл. 2 представлено середній час сумісного виконання алгоритму Евкліда та множення класичним (t_3) та запропонованим (t_4) методами у випадку, коли прості числа a перебувають в межах однієї розрядності від 67 до 127, а на рис. 3 – відповідні графіки залежності від номера числа. При цьому b змінюється від 2 до $a-1$.

Таблиця 2.

Середній час сумісного виконання алгоритму Евкліда та перемноження класичним (t_3) та запропонованим (t_4) методами

№	1	2	3	4	5	6	7
a	67	71	73	79	83	89	97
t_3, с	0.036369	0.037783	0.037268	0.038416	0.038519	0.038299	0.037137
t_4, с	0.03	0.029754	0.030099	0.029987	0.030926	0.030195	0.030947
№	8	9	10	11	12	13	
a	101	103	107	109	113	127	
t_3, с	0.038798	0.038485	0.040181	0.039243	0.039541	0.039544	
t_4, с	0.032222	0.031248	0.030429	0.030822	0.030568	0.03144	

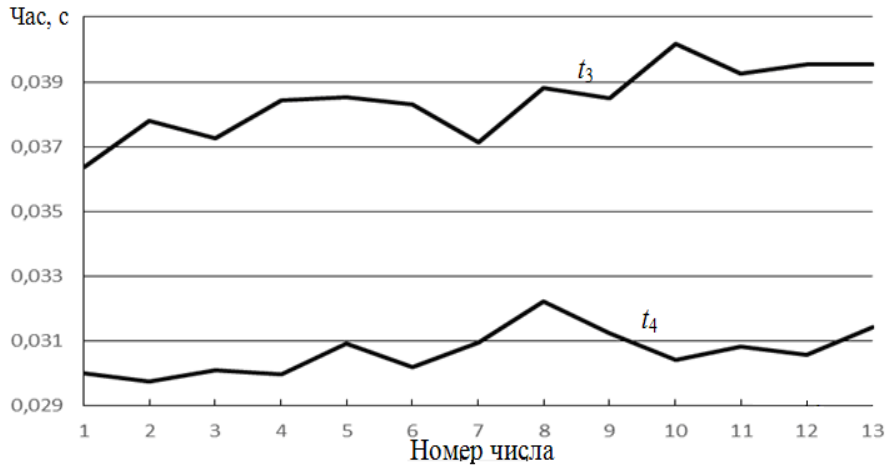

Рис. 3. Графічна залежність середнього часу сумісного виконання алгоритму Евкліда та перемноження класичним (t_3) та запропонованим (t_4) методами від номера числа згідно табл. 2

З рис. 3 видно, середній час роботи запропонованим методом у всіх випадках менший, ніж класичним. Загальний тренд показує зростання часу при збільшенні заданих чисел, причому графік для класичного методу зростає інтенсивніше.

В табл. 3 представлено середній час сумісного виконання алгоритму Евкліда та перемноження класичним (t_5) та запропонованим (t_6) методами у випадку, коли розрядність n простих чисел a перебуває в межах від 7 до 16 біт, а на рис. 4 – відповідні графічні залежності у логарифмічній шкалі. Число b змінюється аналогічно до попереднього випадку.

Таблиця 3.

Середній час сумісного виконання алгоритму Евкліда та перемноження класичним (t_5) та запропонованим (t_6) методами

a	131	181	257	359	521	727
$\log_2 a$	7	7.5	8	8.5	9	9.5
t_5, с	0.040612	0.039525	0.041859	0.043527	0.044844	0.046127
t_6, с	0.031426	0.032223	0.032906	0.034521	0.034865	0.035659
a	1031	1447	2053	2897	4099	5791
$\log_2 a$	10	10.5	11	11.5	12	12.5
t_5, с	0.048333	0.049314	0.051011	0.051922	0.05336	0.054791
t_6, с	0.037083	0.037722	0.039205	0.039943	0.040651	0.041505
a	8209	11587	16411	23173	32771	46337
$\log_2 a$	13	13.5	14	14.5	15	15.5
t_5, с	0.056475	0.05788	0.059334	0.060874	0.062263	0.063511
t_6,с	0.042528	0.043577	0.044608	0.045555	0.04639	0.047953

Видно, що усереднений час роботи збільшується майже лінійно із збільшенням розрядності числа a, причому графік для класичного методу зростає інтенсивніше.

В табл. 4 представлено середній час сумісного виконання алгоритму Евкліда та перемноження класичним (t_7) та запропонованим (t_8) методами у випадку, коли розрядність n простих чисел a (параметру a присвоювалося значення найменшого простого числа, яке перевищувало 2^n) перебуває в межах від 16 до 44 біт, а на рис. 5 –

відповідні графічні залежності у логарифмічній шкалі. Число b набувало 10000 різних значень.

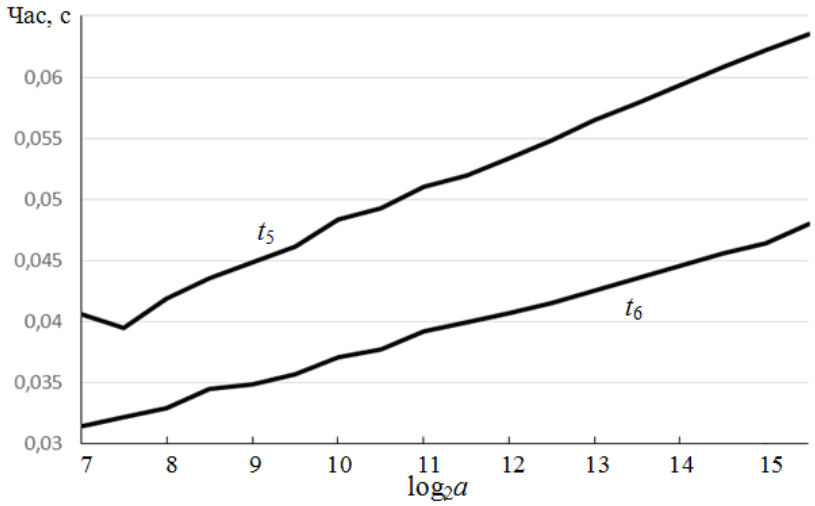

Рис. 4. Графічна залежність середнього часу сумісного виконання алгоритму Евкліда та перемноження класичним (t_5) та запропонованим (t_6) методами від розрядності числа a

Таблиця 4.

Середній час сумісного виконання алгоритму Евкліда та перемноження класичним (t_7) та запропонованим (t_8) методами

$\log_2 a$	16	20	24	28	32	36	40	44
t_7, с	0.06722	0.07799	0.08828	0.09779	0.10631	0.11857	0.12602	0.13206
t_8, с	0.05012	0.05857	0.06734	0.07577	0.08644	0.09782	0.10525	0.10915

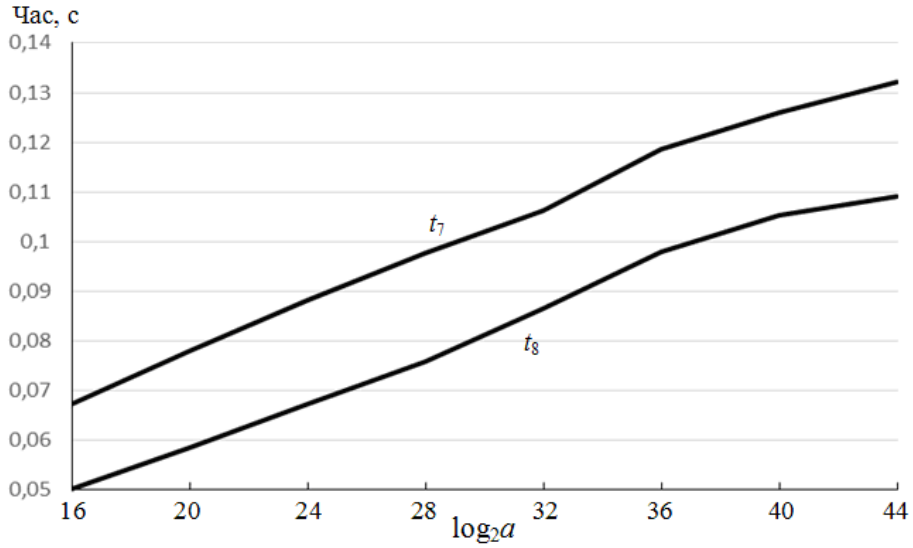

Рис. 5. Графічна залежність середнього часу сумісного виконання алгоритму Евкліда та перемноження класичним (t_7) та запропонованим (t_8) методами від розрядності числа a

Як видно з рисунка, графіки розміщені практично паралельно. При великих розрядностях ($n \geq 40$) інтенсивність зростання t_8 зменшується.

Для нівелювання випадкових впливів на час роботи усі обчислення повторювалися 5000 разів.

Висновки

У роботі проведено експериментальне дослідження часових характеристик програмної реалізації сумісного виконання алгоритму Евкліда та перемноження двох багаторозрядних чисел класичним та запропонованим методами із застосуванням мови програмування високого рівня C++. У запропонованому методі передбачено використання проміжних результатів алгоритму Евкліда та звертання до наявної у пам'яті комп'ютера таблиці квадратів. Для дослідження використовувалися числа різної розрядності. Показано, що в переважній більшості розглянутих випадків запропонований метод характеризується більшою швидкодією, середній час виконання операцій зменшується приблизно в 1.3 рази.

Список літератури

1. Задірака, В.К. Комп'ютерна арифметика багаторозрядних чисел. / В.К. Задірака, О.С. Олексюк. – К.: 2003. – 264 с.
2. Задірака, В.К. Комп'ютерна криптологія / В.К. Задірака, О.С. Олексюк. – Тернопіль, Київ, 2002. – 504 с.
3. 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). - Warsaw, Poland. - V.1, September – 2015. – PP.161–163.
4. 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.
5. Николайчук, Я.М. Теорія джерел інформації / Я.М. Николайчук. – Тернопіль: ТзОВ „Терно–граф", 2010. – 536 с.
6. Jeffrey, H. An Introduction to Mathematical Cryptography / H. Jeffrey, P. Jill, H. Joseph. – Berlin: Springer, 2008. – 540 p.
7. Чернобровкин, В.В. Распараллеливание представлений многоразрядных чисел на модулярных структурах данных / В.В. Чернобровкин // Фундаментальные исследования. – 2013. – №11. – С. 910-914.
8. Iakymenko, I. Construction of distributed thermal or piezoelectric sensor based on residue systems / I. Iakymenko, M. Kasianchuk, Ia. Kinakh, M. Karpinski // Przeglad Elektrotechniczny. – 2017. – №1. – PP. 290-294.
9. Omondi, A. Residue number systems: theory and implementation / A. Omondi, B. Premkumar. – London: Imperial College Press, 2007. – 296 p.
10. Kasianchuk, M. Algorithms of findings of perfect shape modules of remaining classes system / M. Kasianchuk, I. Yakymenko, I. Pazdriy, O. Zastavnyy // Proceedings of the XIII-th International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2015)". – Polyana-Svalyava (Zakarpattya), Ukraine. - 2015. – PP. 168-171.
11. Касянчук, М.М. Теорія та математичні закономірності досконалої форми системи залишкових класів / М.М. Касянчук // Праці Міжнародного симпозіуму „Питання оптимізації обчислень (ПОО–XXXV)". Т.1. – Київ–Кацивелі.– 2009.– С. 306–310.
12. 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. – 2009. – PP. 299–301.
13. 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.
14. Kasianchuk, M.N. Theory and Methods of Constructing of Modules System of the Perfect Modified Form of the System of Residual Classes / M.N. Kasianchuk, Y.N. Nykolaychuk, I.Z. Yakymenko // Journal of Automation and Information Sciences. – 2016. – Vol.48, №8. – PP. 56-63.
15. Виноградов, И.М. Основы теории чисел / И.М. Виноградов. – Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2003. – 176 с.
16. Ишмухаметов, Ш.Т. Методы факторизации натуральных чисел: учебное пособие / Ш.Т. Ишмухаметов. – Казань: Казан. ун., 2011. – 190 с.

17. Николайчук, Я.М. Ефективний метод модулярного множення в теоретико-числовому базисі Радемахера-Крестенсона / Я.М. Николайчук, М.М. Касянчук, І.З. Якименко, С.В. Івасьєв // Вісник Національного університету «Львівська політехніка» «Комп'ютерні системи та мережі». – 2014. – №806.– С. 195–199.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ СОВМЕСТНОГО ВЫПОЛНЕНИЯ АЛГОРИТМА ЕВКЛИДА И УМНОЖЕНИЯ

М.Н. Касянчук, И.З. Якименко, И.Р. Паздрий, С.В. Ивасьев

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

Совместное выполнение алгоритма Евклида и умножения двух многоразрядных чисел является весьма важной задачей современной теории чисел, вычислительной математики и асимметричной криптографии, в частности, криптосистемы Рабина. В работе проведено экспериментальное исследование временной сложности программной реализации указанных операций классическим и предложенным методами с применением языка программирования высокого уровня С ++. В предложенном методе предусмотрено использование промежуточных результатов алгоритма Евклида и обращение к имеющейся в памяти компьютера таблицы квадратов. Для исследования использовались числа различной разрядности. Показано, что в подавляющем большинстве рассмотренных случаев предложенный метод характеризуется более высоким быстродействием, среднее время выполнения операций уменьшается примерно в 1.3 раза. Для нивелирования случайных воздействий на время работы все вычисления повторялись 5000 раз. Предложенный метод эффективно можно использовать для совместного выполнения алгоритма Евклида и умножения двух многоразрядных чисел.

Ключевые слова: алгоритм Евклида, умножение, асимметричная криптография, простые числа, среднее время, временная сложность, разрядность чисел.

EXPERIMENTAL STUDY OF SOFTWARE IMPLEMENTATION OF COMBINE REALIZATION OF THE EUCLID ALGORITHM AND MULTIPLICATION

M.M. Kasianchuk, I.Z. Yakymenko, I.R. Pazdriy, S.V. Ivasiev

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

The joint realization of the Euclid algorithm and multiplication of two numbers is important task of modern number theory, computational mathematics and asymmetric cryptography, including Rabin cryptosystem. The present work provide results of experimental investigation of time characteristics of software implementation these operations by standard and proposed methods using high-level programming language C++. The proposed method provides intermediate results using Euclidean algorithm and reference to computer memory available to the table of squares. In present study we used numbers of different digits. It has been shown that in most cases the proposed method is characterized by a higher speed and reduce average time of operations by 1.3 times. For leveling random effects on the working time all calculations were repeated 5000 times. The proposed method should be used effectively for compatible using Euclidian algorithm and multiplying of two multidigital numbers.

Keywords: Euclidean algorithm, multiplication, asymmetric cryptography, simple numbers, the average time, time characteristics, bit numbers.