УДК 681.32

Н.А. Стефурак, С.В Івасьєв., Р.Б Димашевський., О.Я Лотоцький., Ю.П. Молявчик

Тернопільський національний економічний університет

ЕФЕКТИВНИЙ АЛГОРИТМ ВИЗНАЧЕННЯ ЗАЛИШКУ БАГАТОРОЗРЯДНОГО ДВІЙКОВОГО ЧИСЛА

Вступ. Знаходження залишків від ділення чисел великої розрядності є важливою фундаментальною задачею теорії чисел, успішне вирішення якої дозволяє вдосконалити алгоритми широкого класу прикладних задач [algebra]. Особливо це стосується задач захисту інформаційних потоків в комп'ютерних системах з використанням асиметричної криптографії, зокрема, алгоритмів RSA [1], Рабіна [2], Ель-Гамаля [1], електронного цифрового підпису, використанням математичних основ еліптичних кривих [кінець]. Значну роль операція пошуку залишків відіграє при використанні системи залишкових класів (СЗК) (або модулярної арифметики) [1] та проектуванні відповідних пристроїв [2]. СЗК володіє високим паралелізмом і часто використовується при опрацюванні багаторозрядних даних для пришвидшення процесу обчислень.

Формування модулярного базису може відбуватися на основі різних підходів. Наприклад, в [3] розроблено теоретичні основи досконалої форми СЗК, в [4] – модифікованої досконалої, у яких уникається обчислювально складна процедура пошуку оберненого елемента за модулем [5]. В [6-9] здійснюється використання спеціальних модулів типу $2^n \pm k$ або чисел Мерсена та Ферма, які володіють перевагами при проектуванні операцій модулярної арифметики. Ще одним підходом є вибір великого числа модулів малої розрядності, особливо для багаторозрядних вхідних даних [9]. Для довільних модулів розроблено універсальне рішення для реалізації прямого перетворювача з позиційної системи в модулярний код [9].

Найбільш розповсюдженим для пошуку залишків у позиційній системі числення є алгоритм, згідно якого виділяється ціла частина від ділення багаторозрядного числа на модуль. Отриманий результат множиться на модуль і добуток віднімається від заданого числа. Інший спосіб полягає у послідовному відніманні модуля від заданого

багаторозрядного числа. [10] Їх недоліками є також велика кількість ітерацій, бітових операцій та порівнянь.

Вказані алгоритми пошуку залишку є ефективними лише при використанні апаратних засобів та чисел, що відповідають розрядності процесора. При створенні складних програмних систем, виникає необхідність опрацювання багаторозрядних чисел з допомогою використання спеціалізованих бібліотек[10], в яких операція знаходження залишку має значну обчислювальну складність.

У зв'язку з цим актуальною задачею є розробка алгоритмів пошуку залишків багаторозрядних чисел, які дозволяють досягнути покращення часових характеристик та зменшити часову складність пошуку залишку за рахунок зменшення кількості арифметичних операцій.

Метою роботи є реалізація алгоритмів знаходження залишків багаторозрядних чисел на основі використання властивостей модулярної арифметики, який дозволяє зменшити розрядність чисел, над якими виконуються арифметичні операції, в порівнянні з класичним підходом та, відповідно, зменшити часові складності пошуку залишків.

1. Алгоритм знаходження залишку довільного багаторозрядного числа

В основу запропонованого методу пошуку залишку $Y \bmod P = H$ покладено представлення багаторозрядного числа Y та модуля P у двійковому коді $P = \sum_{i=0}^{k-1} p_i 2^i$, $Y = \sum_{i=0}^{n-1} y_i 2^i$, де p_i, y_i=0, 1.

На першому етапі до молодших розрядів модуля P дописується n-k-2 нулів, в результаті чого отримується двійковий вектор $S = (p_{k-1}, p_{k-2}, \ldots, p_0, 0, \ldots, 0)$. Якщо k-1 старших розрядів Y не перевищує P, то знаходиться $Y \bmod S$ шляхом віднімання $Y - S = M$ і записується $M = \sum_{i=1}^{n-k-2} M_i 2^i, M_i = 0,1$, відповідно у двійковій формі $M = (M_{n-k-2}, M_{n-k-3}, \ldots, M_1, M_0)$.

Якщо $M \geq P$, то формується двійковий вектор L, в якому n-$2k$-3 молодших розрядів є нулями, а старші являють собою двійкове представлення числа P:

$$L = (p_{k-1}, p_{k-2}, \ldots, p_0, 0, \ldots, 0). \quad (1)$$

Далі якщо виконується нерівність $M > L$, то обчислюється значення U=M mod L=M − L, яке записується у двійковій формі

$U=\sum_{l=1}^{n-k-2} U_l 2^l, U_l=0,1$ або у вигляді двійкового вектора $U=(U_{n-2k-3},U_{n-2k-4},...,U_1,U_0)$. Причому, якщо $U \geq P$, то в молодший розряд модуля P дописується n-$3k$-4 нулів і знову ж формується такий двійковий вектор:

$$F=(p_{k-1},p_{k-2},...,p_1,p_0,0,...,0), \quad (2)$$

У разі виконання нерівності $U \geq F$ знаходиться значення $U \bmod F = U-F=H$ і формується двійковий вектор $H=(H_{n-3k-4},H_{n-3k-5},...,H_1,H_0)$. Дана процедура продовжується доти, поки двійковий вектор H не буде меншим за P.

Слід відмітити, що реалізація представленої процедури пошуку залишку в двійковій системі числення зводиться до обчислення різниці двох чисел, розрядність яких на кожному кроці зменшується вдвічі, що дозволяє суттєво зменшити часову складність виконання зазначеної операції:

$$Y \bmod P = U \bmod F = H. \quad (3)$$

На рисунку 1 подано розроблену блок-схему алгоритму пошуку залишку багаторозрядних чисел запропонованим методом.

Основними перевагами даного алгоритму в порівнянні з описаними в роботах [1, 2] є зменшення надлишкового використання пам'яті та кількості порівнянь.

Нехай, наприклад, потрібно обчислити $10989_{10} \bmod 7_{10} = 6_{10}$ або відповідно у двійковій формі

$$10101011101101_2 \bmod 111_2 = 110_2.$$

На першому етапі здійснюється побітовий зсув значення залишку до тих пір, поки кількість розрядів модуля не буде перевищувати розрядність числа:

$$10101011101101_2 - 1110000000000_2 = 111011101101_2.$$

На наступній ітерації від отриманого значення віднімається значення залишку, яке формується за рахунок побітового зсуву, кількість бітів якого менша розрядності проміжного результату віднімання:

$$111011101101_2 - 111000000000_2 = 11101101_2.$$

Аналогічна процедура здійснюється на насупному кроці:

$$11101101_2 - 11100000_2 = 1101_2.$$

В результаті обчислень запропонованим методом отримується шукане значення залишку:

$$1101_2 - 111_2 = 110_2,\ 110_2 < 111_2.$$

Особливістю даного алгоритму є використання властивостей залишків та модулярних операцій, що приводить до зменшення кількості ітерацій.

2. Експериментальні дослідження алгоритму пошуку залишку довільного багаторозрядного числа

Досліджено експериментальні результати часу виконання операції знаходження залишку багаторозрядного числа розробленим та класичним методами для відомих багаторозрядих чисел Мерсена.

Для нівелювання випадкових впливів на час роботи усі обчислення значення залишку для чисел різної розрядності повторювалися 10000 разів, що дало змогу отримати масштабовану різницю часових характеристик та виділити швидший алгоритм.

В результаті чисельного експерименту побудовано графіки часових характеристик розробленого та відомого алгоритмів для 32– (рисунок 2) та 1024-бітних (рисунок 1) чисел, де q – порядковий номер чисел, s – час в секундах.

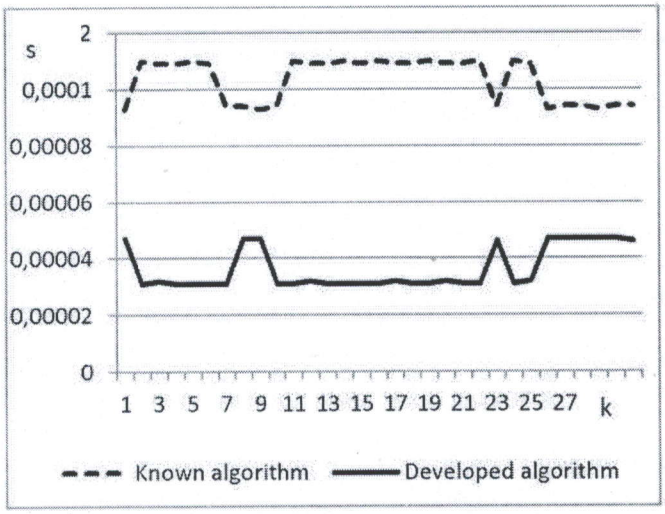

Рисунок 1. Часові харакетеристики розробленого та відомого алгоритму для 32 – бітних чисел

Дослідження показали, що із збільшенням розрядності числової вибірки часові характеристики мають лінійний характер, а швидкодія обчислення операції залишку залежить від хемінгової ваги числа.

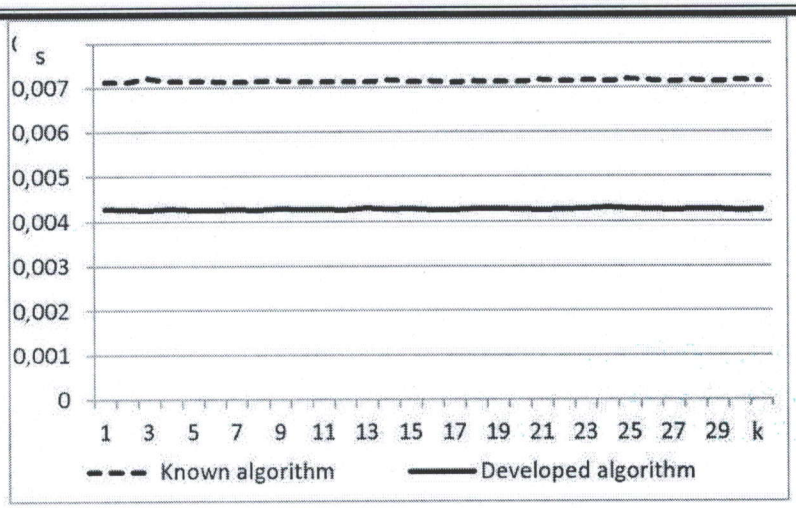

Рисунок 2 - Часові характеристики розробленого та відомого алгоритмів для 1024 – бітних чисел.

3. Алгоритм знаходження залишку багаторозрядного числа Ферма

Для пошуку залишку чисел спеціального виду (Мерсена або Ферма) доцільно використовувати модифікований алгоритм, який буде враховувати властивості чисел відповідної групи. Розглянемо n-розрядне число Ферма $M = 2^n+1$. Представимо його та модуль у двійковій формі: $M_{(2)}$=100000…01, $P = \sum_{i=0}^{k-1} p_i 2^i$, де p_i=0,1, причому вважаємо, що $P>>n$.

Далі потрібно здійснити розклад $M_{(2)}$ у добуток:

$$M_{(2)}=100..00\ldots00\times100..00\ldots00\times\ \ldots\ \times100..00\ldots00. \qquad (6)$$

Кількість бітів кожного множника у розкладі буде дорівнювати n+1.

У (6) кількість однакових множників з розрядністю n+1 буде рівна

$$l = \left\lceil \frac{P}{n} \right\rceil .$$

Для знаходження $2^p \bmod p$ необхідно обчислити залишки кожного з множників рівності (6) шляхом віднімання. У випадку, якщо l – парне, то на наступному етапі залишки $M_{(2)}^2$ групуються попарно і перемножуються, тобто шукаються квадрати $\left(M_{(2)}^2\right)^2$.

В результаті таких операцій отримується l залишків $M_{(2)}^2$ і останній множник у записі (6).

Коли l – непарне, то попарно групуються l-1 залишків $M_{(2)}^2$, які,

аналогічно до попереднього випадку, підносяться до квадрату, і один залишок з останнім множником у формулі (6).

Покрокове виконання запропонованого алгоритму реалізується таким чином:

1. Вхід: n-розрядне число Ферма M та модуль P.
2. Шукається різниця $M_{(2)}^2=2^{n+1}-P$.
3. Шукається ціла частина від ділення $l=P/n$ та $U=2^{p-l\cdot n+1}$.
4. Якщо l - парне, то обчислюється $res=(M_{(2)}^2)^2 \bmod P$ та відбувається побітовий зсув змінної l, тобто $l=l/2$. Якщо l - непарне, тоді $l=l-1$, $U=(U\cdot M_{(2)}^2) \bmod P$.
5. Якщо l – парне, тоді $res=(res\cdot res) \bmod P$ та виконується побітовий зсув змінної l, тобто $l=l/2$. Якщо l – непарне, тоді $l=l-1$, $U=(U\cdot M_{(2)}^2) \bmod P$.
6. Якщо $l>0$, тоді відбувається перехід на крок 5.
7. Відбувається операція модулярного множення та присвоєння $res=(res\cdot U) \bmod P$.

Нехай, наприклад, потрібно обчислити $x=2^{100}+1 \bmod 13$.

Оскільки розрядність модуля дорівнює 4, то число 2^{100} розкладається на 18 п'ятирозрядних добутків та закінчення числа, яке відрізняється від множників. Далі шукається залишок одного з множників за заданим модулем: $16 \bmod 13 = 3$.

Обчислюється проміжний залишок, який отримався в молодших розрядах числа:

$$2^{10}+1 \bmod 13 = 11.$$

Отже, рівність набула такого вигляду:

$$x=2^{100}+1 \bmod 13 = (3\times 19 + 11) \bmod 13.$$

Після цього відбуваються рекурентні ітерації розробленого алгоритму, оскільки другий множник для багаторозрядних чисел може бути досить великим.

Якщо кількість множників непарна, то аналогічно до попереднього число рекурентно перетворюється в набір множників та обчислюється проміжний залишок

$$x= (3\times 18 + 1) \bmod 13.$$

В результаті обчислень рівність набуває такого вигляду:

$$x= (6 \times 9 + 1) \bmod 13.$$

Аналогічні дії приводять до остаточного результату:

$x = (12 \times 4 + 7) \bmod 13$, $x = (11 \times 2 + 7) \bmod 13$, $x = (9 \times 1 + 7) \bmod 13 = 3$.

Слід відмітити, що обчислення залишку 100 розрядного числа відбулось за 5 ітерацій, причому операції виконувались над числами значно меншої розрядності.

4. Порівняння часових складностей розроблених та відомих алгоритмів

Часова складність розробленого алгоритму пошуку залишку довільного багаторозрядного числа становить $O2(n) = \frac{n}{2} \cdot \log_2 n$.

При реалізації алгоритму для пошуку залишку чисел Ферма потрібно $\log_2 l$ кроків, на кожному з яких відбувається перевірка на парність кількості залишків. Така процедура призводить до виконання на кожному кроці двох операцій модулярного множення, які можна виконати векторно-модульним методом [11] з часовою складністю $O(2 \log_2 n)$.

На рисунку 3 представлено графічні залежності обчислювальних складностей відомого та запропонованих методів.

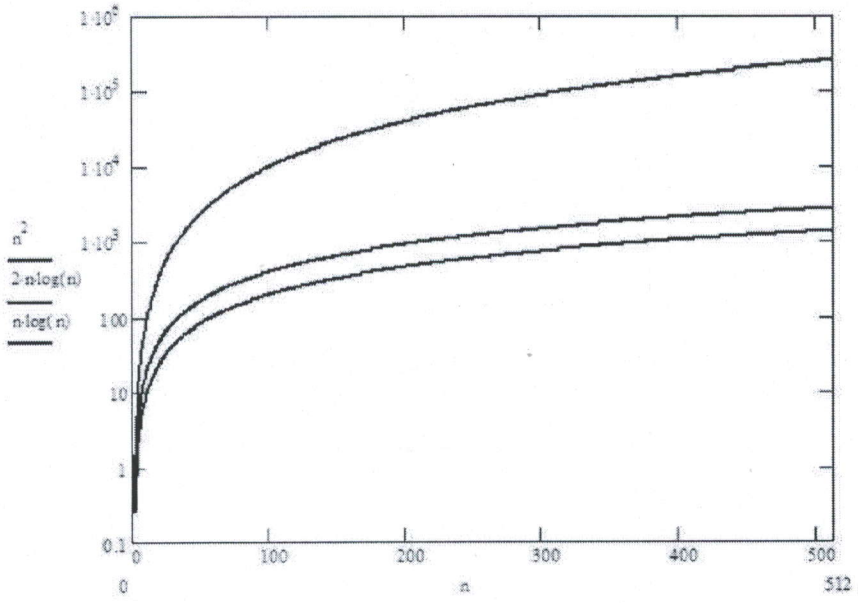

Рисунок 3 - Графічні залежності обчислювальних складностей відомих та запропонованого методу

Чисельний експеримент та оцінка часових складностей відомих і розроблених методів пошуку залишків багаторозрядних чисел, які використовуються при виконанні модульних операцій в асиметричних криптоалгоритмах, переведенні чисел з десяткової системи числення в

систему числення залишкових класів, показує, що при виконанні модульних операцій слід використовувати запропоновані методи.

Висновки.

На основі проведених експериментальних та аналітичних досліджень встановлено, що розроблені методи пошуку залишку багаторозрядних чисел характеризуються нижчою часовою складністю в порівнянні з існуючими. Впровадження запропонованих підходів до виконання операції обчислення залишку в системах захисту та опрацювання інформаційних потоків призведе до зростання їх ефективності.

Перелік джерел.

1. S. Lang, Algebra. 3rd ed. New York: Springer-Verlag; 2002.
2. I.Z. Yakymenko, M.M. Kasianchuk, S.V. Ivasiev, A.M. Melnyk, Ya.M. Nykolaichuk, "Realization of RSA cryptographic algorithm based on vector-module method of modular exponention", Proceedings of the XIV–th International Conference "Modern Problems of Radio Engineering, Telecommunications and Computer Science" (TCSET–2018).–L'viv–Slavske.– 2018, p.550-554.
3. M. Kasianchuk, I. Yakymenko, I. Pazdriy, A. Melnyk, S. Ivasiev, "Rabin's modified method of encryption using various forms of system of residual classes", The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017), 21-25 February, 2017, Polyana-Svalyava, p.222-224.
4. A.E. Okeyinka, Computational Speeds Analysis of RSA and ElGamal "Algorithms on Text Data", Proceedings of the World Congress on Engineering and Computer Science (WCECS 2015) – San Francisco, USA –V. I. – October 21-23, 2015, p.237-242.
5. L. Washington, "Elliptic Curves Number Theory and Cryptography", Series Discrete Mathematics and Its Applications, Chapman & Hall/CRC, 2008, 524 p.
6. A. Omondi, B. Premkumar, "Residue Number System: Theory and Implementation". Imperial College Press, 2007, 296 p.
7. P.V. Ananda Mohan, "Residue number systems: algorithms and architectures", Springer Science+Business Media, NewYork, LLC, 2002, 378 p.
8. M. Kasianchuk, I. Yakymenko, I. Pazdriy and O. Zastavnyy "Algorithms of findings of perfect shape modules of remaining classes system" , XIII International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2015)", Polyana-Svalyava (Zakarpattya), Ukraine, 2015, p.168-171.
9. Ya. M. Nykolaychuk, M.M. Kasianchuk, I.Z. Yakymenko "Theoretical Foundations of the Modified Perfect Form of Residue Number System", Cybernetics and Systems Analysis, 2016, V.52(2), p. 219-223.
10. M.N. Kasianchuk, Ya.N. Nykolaychuk, I.Z. Yakymenko "Theory and Methods of Constructing of Modules System of the Perfect Modified Form of the System of Residual Classes", Journal of Automation and Information Sciences, 2016, Vol.48, №8, p.56-63.
11. T. Rajba, A. Klos-Witkowska, S. Ivasiev, I. Yakymenko, M. Kasianchuk, "Research of Time Characteristics of Search Methods of Inverse Element by the Module", Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS–2017) – Bucharest, Romania. – V.1. – September, 2017, p.82–85.