Одеський національний політехнічний університет
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Єрьоменко Анастасія Іванівна,
студентка групи РЗ-151

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Удосконалення методу модулярного множення в системі залишкових класів для асиметричних криптосистем

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Тимошенко Лідія Миколаївна,
к.е.н., доцент

Одеса – 2020

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення
Рівень вищої освіти другий (магістерський)
Спеціальність 125 – Кібербезпека
Освітня програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КПЗ

_____
д.т.н.,проф. А.А.Кобозєва
____ _____202_р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Єрьоменко Анастасії Іванівні

1.Тема роботи: Удосконалення методу модулярного множення в системі залишкових класів для асиметричних криптосистем,

керівник роботи Тимошенко Лідія Миколаївна, к. е. н., доцент,

затверджені наказом ректора ОНПУ від „____”_____ 20___ р. №_____ .

2.Зміст роботи: *аналіз асиметричних криптосистем; дослідження системи числення залишкових класів; розробка алгоритму побудови трьохмодульної модифікованої форми системи залишкових класів на основі факторизації; обґрунтування вибору інструментальних засобів для реалізації алгоритмів; програмна реалізація виконання операції множення у системі залишкових класів та її модифікованій формі; експериментальне дослідження часових характеристик програмної реалізації виконання операції множення у системі залишкових класів та її модифікованій формі.*

3. Перелік ілюстративного матеріалу: *схема асиметричних криптосистем, ілюстрації, що описують роботу програмного забезпечення, графіки часових характеристик виконання операції множення в трьохмодульній СЗК та МФ СЗК.*

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
Охорона праці	к.т.н., доцент Ярова І.А.	Завдання видав	Завдання прийняв
		05.11.2020	26.11.2020

6. Дата видачі завдання "_____"_____20___ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	Аналіз джерел з теми випускної кваліфікаційної роботи та збір даних	01-09-2020	виконано
2	Удосконалення алгоритму множення для трьохмодульної модифікованої форми системи залишкових класів	01-10-2020	виконано
3	Розробка програмного інтерфейсу	16-11-2020	виконано
4	Підготовка тексту роботи	02-11-2020	виконано
5	Підготовка презентації та доповіді	18-12-2020	виконано
6	Попередній захист	01-12-2020	виконано
7	Нормоконтроль, рецензування	15-12-2020	виконано
8	Занесення роботи в електронний архів	21-12-2020	виконано
9	Допуск до захисту у завідувача кафедри	24-12-2020	виконано

Здобувач вищої освіти _____ Єрьоменко А.І.

Керівник роботи _____ Тимошенко Л.М.

ЗАВДАННЯ

на розробку розділу “Охорона праці та безпека в надзвичайних ситуаціях”

Єрьоменко Анастасії Іванівні, група РЗ-151

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Тема роботи Удосконалення методу модулярного множення в системі залишкових класів для асиметричних криптосистем

Зміст розділу:

1 Аналіз умов праці і вибір заходів і засобів захисту від небезпечних і шкідливих виробничих факторів.
2 Аналіз техногенних небезпек, вибір заходів і засобів забезпечення безпеки у надзвичайних ситуаціях.
3 Розрахунок штучного освітлення

Керівник роботи

_____ (_____)

«____» _____ 2020 р.

Консультант з охорони праці

_____ (_____)

«____» _____ 2020 р.

АНОТАЦІЯ

Дана кваліфікаційна робота на тему “Удосконалення методу модулярного множення в системі залишкових класів для асиметричних криптосистем ” на здобуття освітньо-кваліфікаційного рівня “Магістр” з напряму 125 - «Кібербезпека» містить 7 рисунків, 4 таблиці, 1 додаток, 50 літературних джерел за переліком посилань. Робота виконана на ## сторінках загального тексту і ## сторінках основного тексту.

Метою роботи є підвищення швидкодії модулярного множення для системи залишкових класів шляхом використання модифікованої форми системи залишкових класів та побудови системи модулів для цієї модифікованої форми.

В роботі експериментально досліджено часові затрати під час програмної реалізації операції модулярного множення у трьохмодульній системі залишкових класів та її модифікованій формі.

У результаті виконання кваліфікаційної роботи розроблено алгоритми побудови трьохмодульної модифікованої форми системи залишкових класів.

Результати даної роботи показали, що використання модифікованої форми дає можливість зменшити час обчислення арифметичних операцій за рахунок виключення виконання операції пошуку оберненого елемента за модулем і множення на нього при переведенні в десяткову систему числення. Наведено графічні залежності часових характеристик у вигляді графіків, які підтверджують переваги використання модифікованої форми системи залишкових класів.

АСИМЕТРИЧНІ КРИПТОСИСТЕМИ, СИСТЕМА ЗАЛИШКОВИХ КЛАСІВ, МОДИФІКОВАНА ФОРМА СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ, МОДУЛЯРНЕ МНОЖЕННЯ, БАГАТОРОЗРЯДНІ ЧИСЛА

ANNOTATION

This qualification work on "Improvement of modular multiplication method in the system of residual classes for asymmetric cryptosystems" for the educational qualification level "Master" in the direction of 125 - "Cybersecurity" contains 7 figures, 4 tables, 1 appendix, 50 references according to the list of references. The work is written on ## pages of general text and ## pages of the main text.

The aim of the work is to increase the speed of modular multiplication for the system of residual classes by using a modified form of system of residual classes and construct a system of modules for this modified form.

In work time costs of program implementation of modular multiplication operation in three-modular system of residual classes and its modified form are experimentally investigated.

As a result of the qualification work algorithms for construction of the three-modular modified form of the residual classes system are developed.

Results of this work showed that usage of the modified form enables to reduce time of calculations of arithmetic operations at the expense of exclusion of operation of search of the reverse element modulo and multiplication by it at translation into decimal notation system. Graphic dependences of time characteristics in the form of graphs confirming advantages of using the modified form of the residual classes system are given.

ASYMMETRIC CRYPTOSYSTEMS, RESIDUAL CLASSES SYSTEM, MODIFIED FORM OF RESIDUAL CLASSES SYSTEM, MODULAR MULTIPLICATION, MULTI-DIGIT NUMBERS

ЗМІСТ

ВСТУП

Актуальність роботи. На даний час однією з головних тенденцій у розвитку засобів обчислювальної техніки є створення високопродуктивних обчислювальних пристроїв. Це обумовлено необхідністю вирішення дуже важливих для теорії і практики математики задач, що вимагають обчислень з цілими багаторозрядними числами або величинами, що змінюються в порівняно великих діапазонах.

У зв'язку з цим інтенсивно розвивається прикладний і обчислювальний аспект теорії чисел, який застосовується в інформаційних системах для надійності передачі, зберігання і обробки цифрової інформації. Це призводить до необхідності вирішення значної кількості завдань, де виникають обчислення, при яких цілочисельні змінні можуть набагато перевищувати розрядну сітку існуючих універсальних комп'ютерних засобів, що особливо актуально з розвитком криптографічних методів і засобів захисту інформації.

Арифметичні властивості кожної системи числення перш за все визначаються характером міжрозрядних зв'язків, що виникають в ході виконання відповідних арифметико - логічних операцій. Дослідження показують, що в межах звичайних (десяткової, двійкової) позиційних систем числення стрибкоподібного прискорення виконання арифметичних операцій додавання, віднімання і множення практично досягти неможливо. Таким чином, використання позиційних систем числення призводить до суттєвого зменшення швидкодії, збільшення обчислювальної та часової складності використовуються алгоритмів. Особливо це актуально у випадку підвищення ефективності обробки багаторозрядних чисел.

Таким чином, для вирішення багатьох наукових, технічних і прикладних задач потужності сучасних комп'ютерів може не вистачати. Незважаючи на те, що ресурси наднової обчислювальної техніки, яка функціонує в позиційній системі числення, постійно удосконалюються і збільшуються, вони в принципі не можуть бути безмежними. Це означає, що позиційні системи числення в даний час вичерпують свої можливості для побудови високопродуктивних обчислювальних систем. Тому фундаментальною стратегією теоретичних і практичних досліджень

є використання підходів, заснованих на інтенсивному застосуванні в обчислювальних системах різних форм паралелізму. Даною особливістю володіють непозиційної коди з паралельної структурою. Найбільш перспективною з них є система залишкових класів (СЗК), яка дозволяє реалізувати ідею розпаралелювання на рівні виконання елементарних арифметичних операцій додавання, віднімання і множення. Подання операндів у вигляді залишків від ділення на порівняно невеликі взаємно прості модулі дозволяє уникнути міжрозрядних переносів і набагато зменшити числа, над якими виконуються операції.

Мета роботи. Метою даної роботи є підвищення швидкодії модулярного множення для системи залишкових класів шляхом використання модифікованої форми СЗК та побудови системи модулів для цієї модифікованої форми.

Досягнення поставленої мети включало розв’язання таких взаємопов’язаних завдань:

– аналіз поширених асиметричних криптосистем та операцій опрацювання багаторозрядних чисел в них;

– дослідження системи числення залишкових класів та її модифікованої форми;

– розробка системи модулів модифікованої форми системи залишкових класів на основі факторизації;

– програмна реалізація виконання операції множення у системі залишкових класів та її модифікованій формі;

– експериментальне дослідження часових характеристик програмної реалізації виконання операції множення у системі залишкових класів та її модифікованій формі.

Об'єктом дослідження є процес виконання операції множення в системі залишкових класів.

Предметом дослідження є алгоритми побудови трьохмодульної системи залишкових класів.

Методи дослідження. В основу наукових досліджень покладено методи алгебри і теорії чисел, програмування, комп'ютерного моделювання обчислювальних процесів.

Наукова новизна одержаних результатів визначається наступним:

– запропоновано використання модифікованої форми системи залишкових класів для виконання модулярного множення;

– розроблено систему модулів трьохмодульної модифікованої форми системи залишкових класів на основі факторизації, що дозволило уникнути процедури пошуку оберненого елемента за модулем при використанні системи залишкових класів.

Практична цінність одержаних результатів полягає в тому, що:

- розроблено алгоритм для дослідження часових характеристик системи залишкових класів, що дозволило провести експериментальні дослідження часових характеристик операції множення;

- реалізовано програмне забезпечення для виконання операції множення у системі залишкових класів та її модифікованій формі.

Основні положення і результати роботи опубліковані у фаховому науково-практичному журналі «Сучасна спеціальна техніка» державного науково-дослідного інституту Міністерства внутрішніх справ України [1].

1 АНАЛІЗ СУЧАСНОГО СТАНУ ОПРАЦЮВАННЯ БАГАТОРОЗРЯДНИХ ЧИСЕЛ В АСИМЕТРИЧНИХ КРИПТОСИСТЕМАХ

1.1 Аналіз асиметричних криптосистем

Відомо, що усім симетричним криптосистемам, у яких шифрування та розшифровування відбувається з одним і тим самим ключем, притаманні такі недоліки:

- принциповою є надійність каналу передачі ключа другому учаснику переговорів, тобто ключ повинен передаватися по секретному каналу;
- до служби генерації ключів пред'являються підвищені вимоги, обумовлені тим, що залежність кількості ключів від кількості абонентів є квадратичною.

Для вирішення перерахованих вище проблем симетричного шифрування призначені системи з асиметричним шифруванням або шифруванням з відкритим ключем (рисунок 1.1), що використовують властивості функцій з секретом, розроблених Діффі і Хеллманом [2].

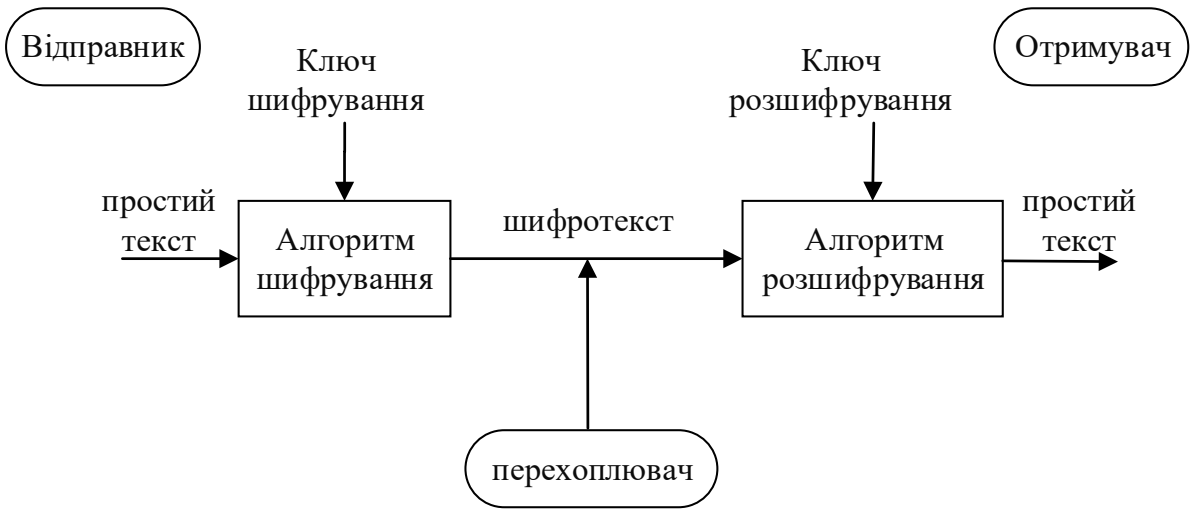

Рисунок 1.1 – Схема асиметричних криптосистем

Ці системи характеризуються наявністю у кожного абонента двох ключів: відкритого і закритого (секретного). При цьому відкритий ключ відкрито передається всім учасникам секретних переговорів. В результаті, вирішуються

проблеми:

- немає потреби в секретній доставці;

- відсутня квадратична залежність кількості ключів від кількості користувачів - для j користувачів потрібно *2j* ключів.

Першим шифром, розробленим на принципах асиметричного шифрування, є шифр RSA. Він названий так за першими літерами прізвищ його винахідників: Рона Райвеста, Аді Шаміра і Леонарда Елдемана - засновників компанії RSA Data Secutity [3]. RSA - не тільки найпопулярніший з асиметричних шифрів, але, мабуть, взагалі найвідоміший.

Математичне обґрунтування RSA таке: пошук дільників дуже великого натурального числа, яке є добутком двох простих чисел – дуже трудомістка процедура. Відповідно, за допомогою відкритого ключа дуже складно обчислити відповідний йому таємний ключ.

Шифр RSA всебічно вивчений і визнаний стійким при достатній довжині ключів. Наприклад, 512 біт для забезпечення стійкості не вистачає, а 1024 біти вважається прийнятним варіантом. З ростом потужності процесорів при даній довжині ключа RSA втрачає стійкість до атаки повним перебором, однак це дозволяє застосувати більш довгі ключі, що в свою чергу підвищить стійкість шифру.

Шифр працює за алгоритмом, який включає в себе генерацію ключів, шифрування та дешифрування.

Для того, щоб згенерувати пари ключів виконуються такі дії [4]:

- вибираються два великі прості числа p і q;
- обчислюється їх добуток $n=p \cdot q$;
- обчислюється функція Ейлера $\varphi(n)=(p-1)(q-1)$;
- вибирається ціле число e таке, що $1<e<\varphi(n)$ та e - взаємно просте з $\varphi(n)$, тобто НСД $(e, \varphi(n))=1$;
- за допомогою розширеного алгоритму Евкліда знаходиться число d таке, що $ed \pmod{\varphi(n)}=1$ або $d=e^{-1} \pmod{\varphi(n)}$.

Для шифрування необхідно знати пару чисел. Перша пара - відкритий ключ, друга - закритий. Знаючи відкритий ключ, можна обчислити значення закритого ключа. Необхідною проміжною дією цього перетворення є знаходження множників p і q, для чого потрібно розкласти n на множники - ця процедура займає дуже багато часу. Саме з величезною обчислювальною складністю пов'язана криптостійкість RSA.

Криптосистема Рабіна стала першою асиметричною криптосистемою, яка ґрунтується на складності знаходження квадратичного лишку. Вона, як і будь-яка асиметрична криптосистема, використовує відкритий і закритий ключі. Відкритий ключ використовується для шифрування повідомлень і може бути опублікований для загального огляду. Закритий ключ необхідний для розшифровки і повинен бути відомий тільки одержувачам зашифрованих повідомлень [5].

Для генерації ключів вибираються два випадкових числа p і q з урахуванням таких вимог:

- числа повинні бути великими;
- числа повинні бути простими;
- повинна виконуватися умова: $p \equiv q \equiv 3 \bmod 4$.

Виконання цих вимог сильно прискорює процедуру знаходження коренів за модулем p і q. Далі обчислюється число $n=p \cdot q$, тоді число n - відкритий ключ; числа p і q - закритий.

Якщо вихідний текст являє собою текстове повідомлення, то визначення правильного тексту не є важким. Однак якщо повідомлення є потоком випадкових бітів (наприклад, для генерації ключів або цифрового підпису), то визначення потрібного тексту стає реальною проблемою. Одним із способів усунути цей недолік є додавання до повідомлення перед шифруванням відомого заголовка або якоїсь мітки [6].

Однак у класичній криптосистемі Рабіна блок відкритого тексту обмежується величиною відкритого ключа ($A<n$). Тому для досить довгих повідомлень потрібно кожен блок шифрувати окремо.

Приблизно такою ж обчислювальною складністю, як і факторизація, володіє операція дискретного логарифмування у скінченному полі, на якій ґрунтується асиметрична криптосистема Ель-Гамаля (Elgamal). Вона включає в себе алгоритм шифрування та алгоритм цифрового підпису. Проте одним з недоліків криптосистеми Ель-Гамаля є подвоєння довжини зашифрованого тексту в порівнянні з початковим текстом. Для схеми імовірнісного шифрування саме повідомлення і ключ не визначають шифротекст однозначно [7].

На даний час криптосистеми з відкритим ключем вважаються найбільш перспективними. До них належить і схема Ель-Гамаля, крипостійкість якої ґрунтується на обчислювальній складності проблеми дискретного логарифмування. Також заслуговують на увагу криптоалгоритми, які використовують арифметику еліптичних кривих, визначених над простими полями Галуа [8].

Однак усі операції у проаналізованих криптоалгоритмах відбуваються в ПСЧ (двійковій або десятковій системі числення), у яких відсутня можливість розпаралелення процесу обчислень.

1.2 Операції опрацювання багаторозрядних чисел

Фундаментальною теоретичною основою сучасних асиметричних криптосистем [10-13] є алгебра і теорія чисел, або, точніше, теорія залишків чи конгруенцій [14, 15]. Цілі числа a та b називаються порівнянними (або конгруентними) за модулем z, якщо різниця чисел $a–b$ націло ділиться на z. Співвідношення між a, b і z запишемо у вигляді:

$$a \equiv b(\mathrm{mod}\ z) \tag{1.1}$$

Запис *mod z* буде означати, що $z \geq 1$, числа a і b – залишки. Запис (1.1) називається конгруенцією.

Відповідно до визначення, запис $a \equiv 0(\text{mod } z)$ означає, що а націло ділиться на *z*. Число а конгруентне числу *b* тоді й тільки тоді, коли *a* й *b* мають однакові залишки при діленні на *z*, тому як визначення конгруенцій можна взяти наступне: цілі числа *a* й *b* називаються конгруентними за модулем *z*, якщо залишки від ділення цих чисел на *z* рівні.

Основні властивості конгруенцій:

- рефлексивність: $a \equiv a(\text{mod } z)$;

- симетричність: якщо $a \equiv b(\text{mod } z)$, то $b \equiv a(\text{mod } z)$;

- транзитивність: якщо $a \equiv b(\text{mod } z)$, $b \equiv c(\text{mod } z)$, то $a \equiv c(\text{mod } z)$;

- якщо $a \equiv b(\text{mod } z)$ і k_0 – довільне ціле число, то $k_0 a \equiv k_0 b(\text{mod } z)$;

- якщо $k_0 a \equiv k_0 b(\text{mod } z)$ й НСД (k_0, z) = 1, то $a \equiv b(\text{mod } z)$;

- якщо $a \equiv b(\text{mod } z)$ й k_0 – довільне натуральне число, то $k_0 a \equiv k_0 b(\text{mod } k_0 z)$;

- якщо $k_0 a \equiv k_0 b(\text{mod } k_0 z)$, де k_0 і z – довільні натуральні числа, то $a \equiv b(\text{mod } z)$;

- якщо $a \equiv b(\text{mod } z)$, $c \equiv d(\text{mod } z)$, то $a + c \equiv b + d(\text{mod } z)$ й $a - c \equiv b - d(\text{mod } z)$;

- якщо $a \equiv b(\text{mod } z)$, $c \equiv d(\text{mod } z)$, то $ac \equiv bd(\text{mod } z)$;

- якщо $a \equiv b(\text{mod } z)$, то при будь-якому цілому k > 0, $a^k \equiv b^k(\text{mod } z)$;

- будь-який доданок лівої або правої частини конгруенції можна перенести із протилежним знаком в іншу частину.

Один з методів виконання арифметичних операцій над даними цілими числами грунтується на простих положеннях теорії чисел. Ідея цього методу полягає в тому, що цілі числа представляються в одній з непозиційних систем – СЗК. А саме: замість операцій над цілими числами оперують із залишками від ділення цих чисел на заздалегідь обрані взаємно прості числа – модулі $p_1, p_2, ..., p_j$. Найчастіше числа $p_1, p_2, ..., p_j$ вибирають із множини простих чисел.

Нехай $A \equiv \alpha_1(\text{mod } p_1)$, $A \equiv \alpha_2(mod\ p_2), \ldots, A \equiv \alpha_j(\text{mod } p_j) \ldots$

Важливо відзначити, що при цьому немає ніякої втрати інформації за умови, що $A < p_1 p_2 ... p_j = P$, тому що завжди, знаючи $(\alpha_1, \alpha_2, ..., \alpha_j)$ можна відновити саме число А. Тому кортеж $(\alpha_1, \alpha_2, ..., \alpha_j)$можна розглядати як один зі способів подання цілого числа А в комп'ютері – модулярне подання або подання в СЗК.

Мультиплікативно оберненим елементом до числа а у модулярній арифметиці є таке число b, що виконується конгруенція [16]:

$$ab \bmod z = 1. \qquad (1.2)$$

Умовою існування мультиплікативно оберненого елемента є рівність 1 найбільшого спільного дільника (НСД) чисел а і b, тобто числа а і b повинні бути взаємно прості. Якщо ця умова не виконується, то мультиплікативно обернений елемент до а не існує.

Методи пошуку мультиплікативно оберненого елемента можна розділити на дві великі категорії [17]: методи, що не ґрунтуються на методах пошуку НСД, і методи, які є похідними від методів пошуку НСД.

Найбільш поширеними методами, які відносяться до першої групи, є повний перебір всіх можливих варіантів (або брутальна атака) та метод на основі функції Ейлера.

Під брутальною атакою розуміється метод рішення математичних задач, при якому складність повного перебору (брутальної атаки) залежить від кількості всіх можливих варіантів вирішення задачі. Цей метод відноситься до класу методів пошуку рішення задач із вичерпуванням можливих варіантів розв’язку системи, є найпростішим і водночас найзатратнішим. Він характеризується високою обчислювальною складністю, оскільки повний перебір вимагає значних часових затрат.

Функція Ейлера $\varphi(z)$, де z - натуральне число, це цілочисельна функція, яка рівна кількості натуральних чисел, не більших за z і взаємно простих з ним. Функцію Ейлера можна подати у вигляді так званого добутку Ейлера:

$$\varphi(z) = z \prod_{p_0 | z} \left(1 - \frac{1}{p_0}\right), \tag{1.3}$$

де p_0 – просте число.

При використанні теореми Ейлера $a^{\varphi(z)} \bmod z = 1$ отримується: $a^{\varphi(z)-1} \bmod z = a^{-1} \bmod z$. Дана процедура передбачає виконання операції модулярного експоненціювання, що може привести до переповнення розрядної сітки процесора і ускладнює пошук оберненого елемента для багаторозрядних чисел.

Найбільш ефективними та поширеними є методи другої категорії, зокрема пошук мультиплікативного оберненого елемента за модулем на основі розширеного алгоритму Евкліда. Для цього спочатку потрібно записати прямий алгоритм Евкліда, згідно якого для будь–якого $z>a=r_0$, де z і a – цілі числа, виконується така система рівнянь:

$$\begin{aligned}
&z = r_0 \cdot q_1 + r_1,\ \ q_1 = a,\ \ 0 \le r_1 < r_0;\\
&r_0 = r_1 \cdot q_2 + r_2,\ \ 0 \le r_2 < r_1;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&r_{j-3} = r_{j-2} \cdot q_{j-1} + r_{j-1},\ \ 0 \le r_{j-1} < r_{j-2};\\
&r_{j-2} = r_{j-1} \cdot q_j + r_j,\ \ 0 \le r_j < r_{j-1};\\
&r_{j-1} = r_j \cdot q_{j+1} + 0.
\end{aligned} \tag{1.4}$$

Оскільки a і z є взаємно простими, то $r_j=1$. Далі для реалізації розширеного алгоритму Евкліда описану процедуру необхідно повторити в зворотньому порядку:

$$\begin{aligned}
1 = r_j &= r_{j-2} - q_j r_{j-1} = r_{j-2} - q_j\left(r_{j-3} - q_{j-1} r_{j-2}\right) = r_{j-2} - q_j r_{j-3} + q_j q_{j-1} r_{j-2} =\\
&= -q_j r_{j-3} + \left(1 + q_j q_{j-1}\right) r_{j-2} = -q_j r_{j-3} + \left(1 + q_j q_{j-1}\right)\left(r_{j-4} - q_{j-2} r_{j-3}\right) = \ldots
\end{aligned} \tag{1.5}$$

Даний процес продовжується до тих пір, поки не отримається вираз $v \cdot z + t \cdot r_0 = 1$, де величина $b = t \bmod z = a^{-1} \bmod z$ і буде шуканим оберненим елементом. В таблиці 1.1 наведено приклад використання розширеного алгоритму Евкліда.

Таблиця 1.1 - 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.

Методи пошуку оберненого елемента на основі алгоритму Евкліда можна розділити на два класи [16]:

– методи, що ґрунтуються на класичному алгоритмі Евкліда;

– методи, що засновані на бінарному алгоритмі Евкліда.

Методи другого класу є більш ефективними, оскільки не використовують обчислювально-витратних операцій ділення на довільне число. Ці методи використовують лише елементарні операції, такі як додавання, віднімання і ділення на 2, що еквівалентне зсуву на один двійковий розряд праворуч.

Поширеним застосуванням розширеного алгоритму Евкліда є китайська теорема про залишки (КТЗ) - один з найдавніших, але досить важливий на даний

час обчислювальний алгоритм [18].

Ще в першому столітті нашої ери китайський математик Сунь–Цзи придумав цікаву загадку, якою було покладено початок модулярній арифметиці: потрібно було знайти число, яке при діленні на 3 дасть в остачі 2, на 5 – 3, на 7 – 2. Крім того, він показав у частковому випадку еквівалентність розв'язку системи модулярних рівнянь і розв'язку одного модулярного рівняння.

Протягом майже двох тисяч років КТЗ постійно вдосконалювалася та розвивалася. Зокрема, в XIII столітті китайський математик Цань Цзю–шао розв'язав наведену вище задачу. У XVIII столітті німецький математик Л. Ейлер навів загальне формулювання та доведення КТЗ, а К.–Ф. Гаус істотно розвинув його в своїх знаменитих «Арифметичних дослідженнях» [19].

І, нарешті, в середині XX століття чеські учені М. Валах та А. Свобода запропонували використати древню китайську ідею, створивши перші модулярні електронно–обчислювальні машини «Епос» та «Епос–2» [20].

Зазначимо, що сьогодні існує декілька еквівалентних формулювань КТЗ. Найбільш поширене з них таке [9]: якщо натуральні числа $p_1, p_2, \ldots, p_j$ попарно взаємно прості, то для будь–яких цілих $r_1, r_2, \ldots, r_j$, таких що $0 \le r_i < p_i$ існує число A, яке при діленні на p_i дає залишок r_i при всіх $i=1, 2, \ldots, j$; більше того, якщо існує два таких числа A_1 та A_2, то $A_1 \bmod P = A_2 \bmod P$, де $P = \prod_{i=1}^{j} p_i$.

Дана теорема представляється переважно у вигляді такої системи порівнянь:

$$\begin{cases} A \bmod p_1 = r_1 \\ A \bmod p_2 = r_2 \\ \ldots\ldots\ldots\ldots \\ A \bmod p_i = r_i \\ \ldots\ldots\ldots\ldots \\ A \bmod p_j = r_j. \end{cases} \qquad (1.6)$$

Шукане число обчислюється за формулою:

$$A = \left(\sum_{i=1}^{j} M_i m_i r_i \right) \bmod P, \qquad (1.7)$$

де $M_i = \frac{P}{p_i} = p_1 p_2 \ldots p_{i-1} p_{i+1} \ldots p_j$, $m_i = M_i^{-1} \bmod p_i$.

Як було сказано вище, пошук мультиплікативного оберненого елемента, необхідний для реалізації КТЗ, характеризується значною обчислювальною складністю в зв’язку з неможливістю розпаралелення процесу обчислень. Причому всі ці операції повинні виконуватися над дуже великими числами, що може призвести до переповнення розрядної сітки сучасних потужних обчислювальних засобів.

1.3 Постановка задачі

Усяка обчислювальна структура тісно пов'язана із системою числення, у якій вона працює. Під системою числення розуміють сукупність прийомів позначення (запису) чисел, або точніше, спосіб кодування (подання) елементів деякої кінцевої моделі дійсних чисел словами одного або більше алфавітів. За допомогою пошуку зворотного елементу, що називають декодуванням, можна визначити систему числення.

Дослідження показали [21-23], що в рамках звичайної позиційної системи числення (ПСЧ) значного прискорення виконання операцій домогтися неможливо. Однак сьогодні перевага віддається обчислювальним структурам, що володіють здатностями до паралельної обробки інформації [24-26]. Цими особливостями володіють непозиційні коди з паралельною структурою, які дозволяють реалізувати ідею розпаралелювання операцій на рівні виконання елементарних арифметичних дій.

Ця думка зародилася в середині 50–х років минулого століття, коли у дослідженнях в області непозиційних систем числення розглядали подання чисел у

вигляді набору залишків від ділення на обрані натуральні модулі – основи системи. Подібну систему числення стали називати системою залишкових класів (СЗК) або модулярною системою числення (МСЧ).

Пошук мультиплікативного оберненого числа є досить складною і громіздкою задачею, що загальмувало розвиток СЗК. В роботах [27-29] описана досконала форма СЗК, в якій виконується наступна умова:

$$M_i \bmod p_i = 1, \tag{1.8}$$

Це дозволяє уникнути процедури пошуку оберненого елемента і множення на нього в (1.5). У [30-32] вирішена задача та визначені умови для аналітичного знаходження m_i. Однак перші два модулі повинні бути строго визначені і дорівнюють 2 і 3. Крім того, значення p_i швидко збільшуються, що неприпустимо при необхідності використання модулів однакової розрядності [33-34].

Тому метою даної магістерської роботи є підвищення швидкодії модулярного множення для системи залишкових класів шляхом використання модифікованої форми СЗК та побудови системи модулів для цієї модифікованої форми. Для досягнення поставленої мети потрібно вирішити такі взаємопов'язані завдання:

– аналіз асиметричних криптосистем;

– дослідження системи числення залишкових класів;

– розробка алгоритму побудови трьохмодульної модифікованої форми системи залишкових класів на основі факторизації;

– обґрунтування вибору інструментальних засобів для реалізації алгоритмів;

– програмна реалізація виконання операції множення у системі залишкових класів та її модифікованій формі;

– експериментальне дослідження часових характеристик програмної реалізації виконання операції множення у системі залишкових класів та її модифікованій формі.

Отже, в даному розділі проаналізовані асиметричні криптосистеми, а також розглянуто операції опрацювання багаторозрядних чисел. У результаті з’ясовано, що усі операції у проаналізованих асиметричних криптосистемах відбуваються в двійковій або десятковій системі числення, у яких відсутня можливість розпаралелення процесу обчислень. Причому всі ці операції повинні виконуватися над дуже великими числами, що може призвести до переповнення розрядної сітки сучасних потужних обчислювальних засобів. Тому запропоновано використання непозиційних систем числення, зокрема, системи залишкових класів.

2 МЕТОД МОДУЛЯРНОГО МНОЖЕННЯ ДЛЯ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

2.1 Використання модулярного множення в системі числення залишкових класів

Модульне множення – операція, яка найчастіше зустрічається в алгоритмах криптографії з відкритим ключем. Якщо виконувати модульне множення безпосередньо, спочатку перемножуючи цілі числа, а потім обчислюючи залишок від ділення, то складність алгоритму буде визначатися процедурою знаходження залишку. Розглянемо методи модульного множення, які не потребують ділення [35].

Найбільш просто виконується множення по модулю чисел p виду $p = 2^n \pm 1$. Для цього потрібно обчислити цілочисельний добуток співмножників і представити його у вигляді 2^nA+B, де B − молодші n біт добутку. Добуток по модулю p дорівнюватиме $B \pm A \ (mod\ p)$. Узагальненням цього методу є множення по модулю чисел $p = 2n \pm c$ для малого c. У цьому випадку, якщо $ab=2na + b$, де $b < 2n$, то $ab \equiv B \pm cA \ (mod\ p)$.

Для побудови СЗК потрібно вибрати додатні взаємно прості числа $p_1, p_2, ..., p_n$, які називаються основами чи модулями системи [30]. Їх добуток $P = \prod_{i=1}^{n} p_i$ характеризує величину діапазону системи, в якому містяться отримані результати. СЗК є така непозиційна система числення, у якій будь-яке ціле невід'ємне число N представляється у вигляді сукупності залишків від ділення даного числа на обрані модулі системи, тобто $N = (\alpha_1, \alpha_2, ..., \alpha_n)$, де $\alpha_i = A - \left[\frac{A}{p_i} \right] \cdot p_i, \left(i = \overline{1,n}\right).$

Можливість такого представлення десяткових чисел визначається теоремою про ділення із остачею [31]. Можна помітити, що будь-яка остача α_i отримується незалежно від усіх інших і містить в собі інформацію про все число.

Встановити взаємно-однозначну відповідність між цілими числами із діапазону [0, P) та їх залишками по взаємно простих відповідних модулях дає можливість китайська теорема про залишки.

Можливість використання СЗК у обчислювальних алгоритмах обумовлюється тим, що правила множення, додавання, піднесення до цілого додатного ступеня будь–яких цілих додатних чисел є повністю ідентичними відповідним операціям, що виконуються із системою залишків, тобто виконуються покомпонентно [32].

Нехай операнди A і B, та результати операцій додавання і множення $A+B$ та $A*B$ відповідно представлені залишками α_i, β_i, γ_i, δ_i по модулях $p_i, \left(i=\overline{1,n}\right)$. При цьому результати і обидва числа перебувають в діапазоні [0, P). Тоді:

$$\gamma_i = \alpha_i + \beta_i (mod\ p_i); \tag{2.1}$$

$$\delta_i = \alpha_i \beta_i (mod\ p_i). \tag{2.2}$$

Справедливість таких правил при виконанні арифметичних дій в СЗК випливає безпосередньо з властивостей конгруенцій.

Отже, виконання цих арифметичних операцій в модулярному коді незалежно відбувається по кожному із його модулів. Це і вказує на можливий паралелізм даної системи. Така обставина строго визначає порозрядне виконання операцій. Ця властивість позбавляє від необхідності «переносити» або «позичати» одиницю від старшого розряду, що і приводить до виникнення кодів із паралельною структурою. Тобто дозволяє розпаралелити виконання алгоритмів при виконанні арифметичних операцій.

Звичайно, ця система теж не позбавлена недоліків. До них можна віднести неможливість для візуального порівняння різних чисел [25], відсутність ознаки виходу результатів поза межі діапазону, а також обмеженість дії цієї системи сферою тільки цілих додатних чисел. А основний недолік СЗК - це є відсутність операції ділення [26]. Але СЗК успішно можна використовувати в

вузькоспеціалізованих обчислювальних машинах при виконанні операцій віднімання, додавання і множення, зокрема, в задачах криптографії [27], кодування інформації. Треба відмітити, що така система ефективна особливо при обчисленнях із великими числами.

Метод відновлення деякого десяткового числа за його залишками був знайдений ще у Китаї більш, ніж дві тисячі років тому. Теоретичною основою для цього методу є китайська теорема про залишки:

$$N = \left(\sum_{i=1}^{n} b_i B_i \right) mod\ P, \tag{2.3}$$

де b_i – залишки,

$B_i = M_i m_i$ – базисні числа системи,

$M_i = \frac{P}{p_i}$, m_i – ваги базисів, які шукаються з виразу $(M_i m_i) mod\ p_i = 1$.

При цьому виконуватись повинна така умова: $\left(\sum_{i=1}^{n} B_i \right) mod\ P = 1$.

Потрібно відзначити, що пошук модулів $m_i = M_i^{-1} mod\ p_i$ становить велику обчислювальну складність.

Найбільш поширеними методами пошуку оберненого елемента на даний час $m_i = M_i^{-1} \bmod p_i$ є такі [36]:

- перебором всіх можливих чисел;
- на основі функції Ейлера;
- за допомогою розширеного алгоритму Евкліда.

Найзатратнішим та водночас найпростішим методом для пошуку оберненого елементу за модулем є його знаходження за допомогою перебору всіх можливих варіантів елементу.

Характеризується даний метод значною обчислювальною складністю, так як

великих часових затрат вимагає повний перебір.

Функція Ейлера $\varphi(n)$ визначається як кількість натуральних чисел, що є меншими за n і одночасно взаємно прості з n. Використовується теорема Ейлера, згідно якої $a^{\varphi(n)} \bmod n \equiv 1$, причому числа a і n повинні бути взаємно простими.

Звідси отримується, що $a^{\varphi(n)-1} \bmod n = a^{-1} \bmod n$. Однак дана процедура, передусім, передбачає виконання операції модулярного експоненціювання [37], тобто піднесення до степеня за модулем, що може приводити до переповнення розрядної сітки процесора і тому ускладнює пошук оберненого елемента для багаторозрядних чисел.

Найбільш поширеним методом для пошуку мультиплікативного оберненого елемента по модулю є метод за допомогою розширеного алгоритму Евкліда. Для цього спочатку потрібно записати хід прямого алгоритму Евкліда для знаходження найбільшого спільного дільника двох чисел. Згідно нього, для будь–якого $n>a=r_0$, де n і a є цілими числами, повинна виконуватися система рівнянь:

$$\begin{aligned}
&n = r_0 \cdot q_1 + r_1, \;\; q_1 = a, \;\; 0 \le r_1 < r_0;\\
&r_0 = r_1 \cdot q_2 + r_2, \;\; 0 \le r_2 < r_1;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&r_{n-3} = r_{n-2} \cdot q_{n-1} + r_{n-1}, \;\; 0 \le r_{n-1} < r_{n-2};\\
&r_{n-2} = r_{n-1} \cdot q_n + r_n, \;\; 0 \le r_n < r_{n-1};\\
&r_{n-1} = r_n \cdot q_{n+1} + 0.
\end{aligned} \qquad (2.4)$$

Оскільки числа a і n вибрано взаємно простими, то тоді $r_n=1$. У розширеному алгоритмі Евкліда передбачається зворотній порядок від проходження прямого алгоритму:

$$\begin{aligned}
&1 = r_n = r_{n-2} - q_n r_{n-1} = r_{n-2} - q_n\left(r_{n-3} - q_{n-1} r_{n-2}\right) = r_{n-2} - q_n r_{n-3} + q_n q_{n-1} r_{n-2} = -q_n r_{n-3} +\\
&+ \left(1 + q_n q_{n-1}\right) r_{n-2} = -q_n r_{n-3} + \left(1 + q_n q_{n-1}\right)\left(r_{n-4} - q_{n-2} r_{n-3}\right) = \ldots
\end{aligned} \qquad (2.5)$$

Продовжується описана процедура до тих пір, поки не буде отримано деякий вираз $v \cdot n + t \cdot r_0 = 1$, де величина $b = t \bmod n = a^{-1} \bmod n$ якраз і буде шуканим мультиплікативним оберненим елементом.

Даний метод містить велику кількість ділень із остачею, множень та підстановок, хоч і володіє він мінімальною часовою складністю, якщо порівнювати із іншими двома методами: перебором можливих варіантів та за допомогою функції Ейлера.

2.2 Модифікована форма системи залишкових класів

Зворотне перетворення з СЗК у десяткову систему числення ґрунтується на використанні китайської теореми про залишки [38, 39]:

$$N = \left(\sum_{i=1}^{n} b_i B_i \right) mod\ P, \tag{2.6}$$

де $P = \prod_{i=1}^{n} p_i$,

$B_i = M_i m_i$,

$M_i = \frac{P}{p_i}$,

m_i – ваги базисів, які знаходять з виразу $(M_i m_i) mod\ p_i = 1$.

У роботі [1] описана модифікована форма СЗК, у якій підбір модулів такий, що

$$M_i \bmod p_i = 1, \tag{2.7}$$

тобто m_i =1. В [11], [12] розвинуто дану теорію, однак не вказано методу побудови всіх можливих варіанті в наборів модулів МФ СЗК при заданій їх

кількості.

Запишемо вираз (2.7) у вигляді системи:

$$\begin{cases} M_1 mod\ p_1 = 1 \\ \dots \\ M_n mod\ p_n = 1. \end{cases} \quad (2.8)$$

Домноживши кожне рівняння на відповідний модуль, отримаємо:

$$\begin{cases} P\ mod\ p_1^2 = 1 \\ \dots \\ P\ mod\ p_n^2 = 1. \end{cases} \quad (2.9)$$

Розв’язуючи (2.9) стандартними методами теорії чисел згідно китайської теореми про залишки, матимемо:

$$P = \left(\sum_{i=1}^{n} p_i M_i^2 m_i^2) \ mod\ M \right), \quad (2.10)$$

де $M = \prod_{i=1}^{n} p_i^2 = P^2$.

Врахувавши, що у модифікованій формі СЗК $m_i = 1$, та скоротивши модуль, ліву та праву частину (2.10) на їх дільник $P = \prod_{i=1}^{n} p_i$, запишемо (2.10) таким чином:

$$\left(\sum_{i=1}^{n} M_i \right) \bmod P = 1. \quad (2.11)$$

Вираз (2.11) еквівалентний рівності:

(2.12)

$$\sum_{i=1}^{n} M_i = kP + 1,$$

де *k = 1, 2, 3, ...* .

Поділивши ліву та праву частини (2.12) на $P = p_1 \times p_2 \times p_3$, отримаємо остаточний вираз для пошуку набору модулів у МФ СЗК:

$$\sum_{i=1}^{n} \frac{1}{p_i} = k + \frac{1}{\prod_{i=1}^{n} p_i}. \qquad (2.13)$$

Елементарною підстановкою можна переконатися, що єдино можливою системою з трьох модулів МФ СЗК є 2, 3, 5, оскільки при збільшенні будь-якого p_i ліва частина (2.13) стає меншою 1 [40].

2.3 Побудова системи модулів модифікованої форми в системі залишкових класів на основі факторизації

Обмежимо наші розрахунки трьома модулями [41, 42] і запишемо вираз (1.8) у вигляді системи:

$$\begin{cases} p_2 p_3 mod\ p_1 = \pm 1 \\ p_1 p_3 mod\ p_2 = \pm 1 \\ p_1 p_2 mod p_3 = \pm 1 \end{cases} \qquad (2.14)$$

Домножимо кожне рівняння (2.14) на відповідний модуль. Маємо:

$$\begin{cases} p_1 p_2 p_3 mod\ p_1^2 = \pm 1 \\ p_1 p_2 p_3 mod\ p_2^2 = \pm 1 \\ p_1 p_2 p_3 mod\ p_3^2 = \pm 1 \end{cases} \qquad (2.15)$$

Розв’язуємо (2.15) стандартними методами з теорії чисел на основі китайської теореми про залишки і отримаємо:

$$P = \left(\sum_{i=1}^{3} p_i M_i^2 \, m_i^2 \right) mod \, M, \tag{2.16}$$

де $M = \prod_{i=1}^{3} p_i^2 = P^2$.

Далі треба врахувати, що в МФ СЗК $m_i = \pm 1$, тоді скоротити модуль, ліву і праву частину (2.16) на спільний дільник $P = \prod_{i=1}^{3} p_i$. Звідси (2.16) запишеться таким чином:

$$\left(\sum_{i=1}^{3} M_i \right) mod \, P = \pm 1. \tag{2.17}$$

Вираз (2.17) буде тотожний такій рівності:

$$\sum_{i=1}^{n} M_i = kP \pm 1, \tag{2.18}$$

де $k = \pm 1, \pm 2, \pm 3, \ldots$.

Якщо поділити ліву і праву частини (2.18) на $P = p_1 p_2 p_3$, то можна отримати остаточний вираз для пошуку набору модулів у МФ СЗК:

$$\sum_{i=1}^{3} \frac{1}{p_i} = k \pm \frac{1}{\prod_{i=1}^{n} p_i}. \tag{2.19}$$

На відміну від СЗК, де всі модулі повинні бути додатні і $k>0$, у МФ СЗК модулі мають мати різні знаки [43]. Для спрощення задачі можна прийняти $k=0$, що відповідає найбільшому діапазону обчислень для заданої кількості модулів. Тому остання рівність матиме такий вигляд:

$$\frac{1}{p_1}+\frac{1}{p_2}+\frac{1}{p_3}=\pm\frac{1}{p_1p_2p_3}. \tag{2.20}$$

Крім того, на відміну від ДФ СЗК, де найменші модулі мають строго визначені значення ($p_1=2$, $p_2=3$), у МФ СЗК найменші модулі можуть бути будь-які. Домноживши (2.20) на P, можна отримати:

$$p_1p_2+p_2p_3+p_1p_3=\pm1. \tag{2.21}$$

Представимо (2.21) у такому вигляді:

$$p_2p_3+p_1(p_2+p_3)=\pm1. \tag{2.22}$$

Далі потрібно ввести позначення:

$$p_{2,3}=a,b-p_1. \tag{2.23}$$

Після підстановки (2.23) у (2.22) і відповідних математичних перетворень можна отримати умову, яка має виконуватися при визначенні набору із трьох модулів МФ СЗК:

$$\pm1+p_1^2=ab \tag{2.24}$$

Отже, ліва частина (2.24) має бути факторизована [44, 45], звідки визначаються параметри a і b. Тому вираз (2.24) визначає умову для побудови МФ

СЗК із трьох модулів.

Нехай для прикладу $p_1=7$. Тоді з (2.23) та (2.24) можна отримати: $p_{2,3} = a, b - 7i \pm ab = 1 + 49 = \begin{cases} 50 = 2 \cdot 5 \cdot 5 \\ 48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \end{cases}$. Всі можливі варіанти із трьох модулів для МФ СЗК при $p_1=7$ наведені в таблиці 2.1.

Таблиця 2.1 - Можливі варіанти систем із трьох модулів для МФ СЗК при $p_1=7$ (в дужках – розрядність у двійковій системі числення).

№	p_1	ab	a	b	p_2	p_3	P
1	7 (3)	48	1	48	-6 (3)	41 (6)	1722 (11)
2			-1	-48	-8 (4)	-55 (6)	3080 (12)
3			2	24	-5 (3)	17 (5)	595 (10)
4			-2	-24	-9 (4)	-31 (5)	1953 (11)
5			3	16	-4 (3)	9 (4)	252 (9)
6			-3	-16	-10 (4)	-23 (5)	1610 (11)
7			4	12	-3 (2)	5 (3)	105 (7)
8			-4	-12	-11 (4)	-19 (5)	1463 (11)
9			6	8	-1 (1)	1 (1)	7 (3)
10			-6	-8	-13 (4)	-15 (4)	1365 (11)
11		50	1	50	-6 (3)	43 (6)	1806 (11)
12			-1	-50	-8 (4)	-57 (6)	3192 (12)
13			2	25	-5 (3)	18 (5)	630 (10)
14			-2	-25	-9 (4)	-32 (6)	2016 (11)
15			5	10	-2 (2)	3 (2)	42 (6)
16			-5	-10	-12 (4)	-17 (5)	1428 (11)

Для визначення характеру графіка залежності знайдених модулів їх треба перенумерувати у порядку зростання їх абсолютної величини p_3 (таблиця 2.1).

На рисунку 2.1 показано характер зміни для значень модулів p_3 і p_4 залежно від номера модуля на основі таблиці 2.2.

Таблиця 2.2 - Впорядкування модулів

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_2	1	2	3	4	5	5	6	6	8	8	9	9	10	11	12	13
p_3	1	3	5	9	17	18	41	43	55	57	31	32	23	19	17	15

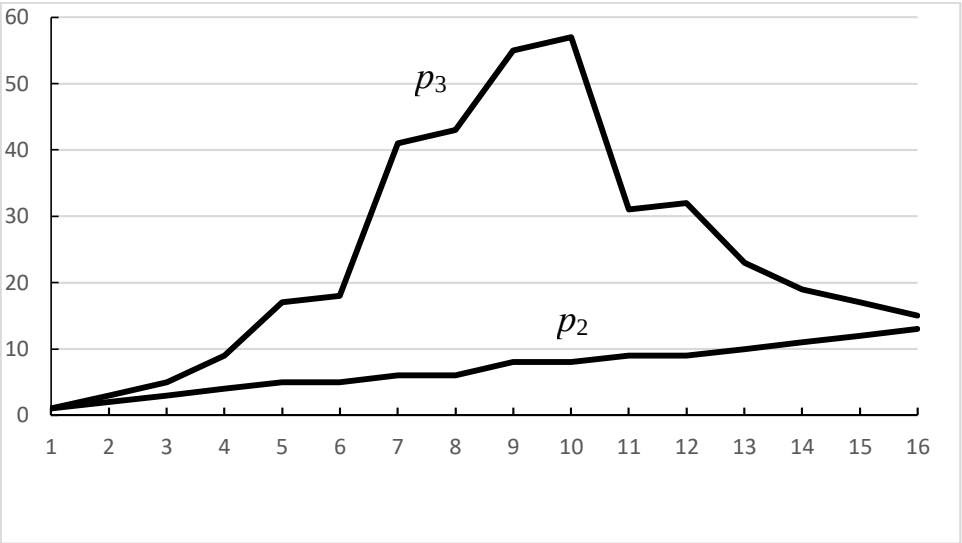

Рисунок 2.1 - Характер зміни значень модулів p_2 та p_3 в залежності від номера модуля згідно таблиці 2.1

Потрібно зазначити, що найбільший діапазон обчислень буде у тому випадку, якщо кожен наступний модуль від добутку абсолютних величин попередніх модулів більший на одиницю. Із таблиці 2.1 слідує, що при використанні цих модулів МФ СЗК розрядність чисел, над якими виконуються усі арифметичні операції, зменшується у 2-3 рази.

У даному розділі розглянуто згідно поставленого завдання модулярне множення в системі числення залишкових класів та модифіковану форму системи залишкових класів. Запропоновано метод побудови модулів модифікованої форми системи числення залишкових класів на основі факторизації, що дозволяє уникнути виконання операцій знаходження мультиплікативного оберненого елемента за модулем та множення на нього при переведенні чисел із системи залишкових класів у десяткову систему числення.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МНОЖЕННЯ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ ТА ЇЇ МОДИФІКОВАНІЙ ФОРМИ

3.1 Обґрунтування вибору інструментальних засобів

Мова Python є однією із найпростіших у вивченні та використанні серед найпоширеніших мов програмування. Програмний код мовою Python читати і писати легко. Все це завдяки тому, що Python – це мова, яка дозволяє вмістити застосування в меншу кількість рядків, ніж при використанні інших мов, зокрема C ++ або Java.

Python - мультиплатформенна мова: зазвичай одна й та сама програма мовою Python запускатися може у різних операційних системах (Windows, UNIX, Linux, BSD та Mac OS). Для цього просто треба скопіювати файли програми на комп'ютер; причому не потрібно навіть виконувати "збірку" чи компіляцію програми.

Однією із переваг мови Python є наявність повної стандартної бібліотеки, яка дозволяє задовольнити вимоги користувачів, зокрема, завантажити файл із Інтернету, розпакувати архів чи створити за допомогою кількох рядків програмного коду веб-сервер.

Існують також тисячі додаткових бібліотек від сторонніх виробників, що забезпечують складніші і потужніші можливості, наприклад, бібліотека для організації мережних взаємодій Twisted, для виконання обчислювальних завдань Numpy чи пакет моделювання Simpy. При цьому більшість сторонніх бібліотек можна знайти в Інтернеті.

Саме ці обставини і спонукали зупинити вибір на мові програмування Python. Python може використовуватися для програмування у об'єктно-орієнтованому, процедурному і, меншою мірою, в функціональному стилі програмування, хоч загалом Python – це є об'єктно-орієнтована мова програмування.

Історія створення та розвитку мови пов'язана із життєвими і службовими проблемами її автора - Гвідо ван Россума. Назва виникла не від виду плазунів, а на честь британського комедійного популярного телешоу 1970-х років "Літаючий цирк Монті Пайтона". Але назву мови найчастіше асоціюють із змією – піктограми

файлів у KDE чи у Microsoft Windows та емблема на сайті Python.org (до виходу версії 2.5) зображають зміїні голови. Розвиток мови відбувається згідно з чітко регламентованим процесом створення, обговорення, відбору та реалізації документів PEP (англ. Python Enhancement Proposal) – пропозицій щодо розвитку Python.

Мови програмування часто оцінюють за рівнем: об'єктно-орієнтовані, процедурні, логічні, функціональні. Рівень мови показує, наскільки вона є близькою до природного для людини запису. Процедурні мови – найнижчого рівня, функціональні – значно вищого. Логічні принципово належати могли б до найвищого рівня, та через високу складність теорії, яка лежить у їх основі, досить повільно розробляються.

При використанні об'єктів можливе збільшення продуктивності програмістів в діапазоні 20–30 %. Змінні і функції у об'єктно-орієнтованому програмуванні групуються в класи. Досягається завдяки цьому вищий рівень для структуризації програми. Об'єктно-орієнтований спосіб написання програм не є особливий і самостійний, тому що базується на процедурній моделі програмування.

Об'єкти не збільшують продуктивність настільки істотно, як сценарна чи скриптова технологія, а переваг, забезпечуваних ними, можна досягнути і за допомогою мов сценаріїв (скриптів). Сильна типізація у більшості об'єктно-орієнтованих мов робить модулі спеціалізованими, ускладнюючи їх повторне використання. Інша проблема цих мов – їх акцент на успадкуванні.

Реалізації для класів прив'язуються одна до одної, тому жоден клас без іншого зрозуміти не можна. Мови сценаріїв реалізують принцип повторного використання. Модель, яку застосовують вони при створенні програм, враховує те, що необхідні компоненти вже є в системі, їх можна "склеювати" за допомогою мови сценаріїв. Такий "поділ праці" забезпечує природну схему, яка полегшує повторне використання коду.

Однак об'єктно-орієнтоване програмування принаймні має дві корисні властивості. Перша – це інкапсуляція: об'єкти поєднують дані і код способом, який приховує деталі реалізації. Друга – це спадкування інтерфейсу, коли класи

забезпечують ті ж самі методи та атрибути, навіть коли вони по-різному реалізовані. Переваг об'єктів у мовах програмування систем можна досягнути й у мовах сценаріїв. При цьому зберігається властива для мов сценаріїв відсутність в об'єктів типу.

Python – це є універсальна інтерпретована, високорівнева об'єктно-орієнтована мова програмування сценаріїв з динамічною семантикою. Розвинені вбудовані структури даних в поєднанні із динамічною типізацією і динамічним зв'язуванням роблять дуже привабливою її для швидкої розробки застосувань, та для використання в якості скриптової чи мови, яка "склеює" разом усі наявні компоненти. Мова Python проста, має легкий синтаксис, високу читабельність забезпечує і тому зменшує загальну вартість при експлуатації програм.

Python підтримує модулі і пакети, що сприяють мобільності програм та повторному використанню коду. Інтерпретатор Python і розширена стандартна бібліотека доступна як у двійковому форматі, так і у вихідному коді, причому безкоштовно для всіх основних платформ. Його поширювати можна вільно та навіть вбудовувати у власні застосування.

Запуск інтерпретатора здійснюється звичайно просто командою Python.exe чи вказуванням повного шляху до інтерпретатора. Щоб вийти із Python, скористатися треба комбінацією клавіш CTRL+D – Unix; CTRL+Z чи CTRL+Break – Dos тощо. Якщо не допомогло це, то набрати можна у відповідь на запрошення інтерпретатора (>>>) такі рядки: >>> import sys або sys.exit().

Інтерпретатор працює в двох режимах: інтерактивному і власне інтерпретатора. Вхід у інтерактивний режим здійснюється введенням Python без параметрів, параметр file викликає інтерпретацію або виконання зазначеного файла. У інтерактивному режимі Python про себе пише інформацію і про систему, потім своє запрошення виводить (>>>). Необхідно ввести рядок коду Python з клавіатури. Інтерпретувати введений рядок можна клавішею Enter. Коментарі у Python позначаються символом # і продовжуються до кінця рядка:

```
>>> a = "Це рядок" # - це коментар;
>>> b = "# це вже НЕ коментар"
```

Незважаючи на зручності для інтерактивного режиму роботи при написанні програм на мові Python, зазвичай потрібно зберігати вихідний програмний код і для подальшого використання. В такому випадку підготовлюються файли, що передаються потім інтерпретатору на виконання. По відношенню до інтерпретованих мов програмування дуже часто вихідний код називають скриптом. Файли з кодом на Python зазвичай мають розширення .py.

Отже, програма (чи скрипт) на Python – це команди, які записані у текстовому файлі. Програмі параметри можна передати через командний рядок, розділяючи їх пробілами. Передаються вони у список sys.argv[i] для подальшої обробки в програмі. За необхідності вказати треба повні шляхи до файлів:

Python.exe скрипту параметри.

На рисунку 3.1 представлено головне вікно програми.

Існує для Python багато різних графічних середовищ програмування. Ефективність роботи програміста залежить істотно від правильного вибору такого середовища.

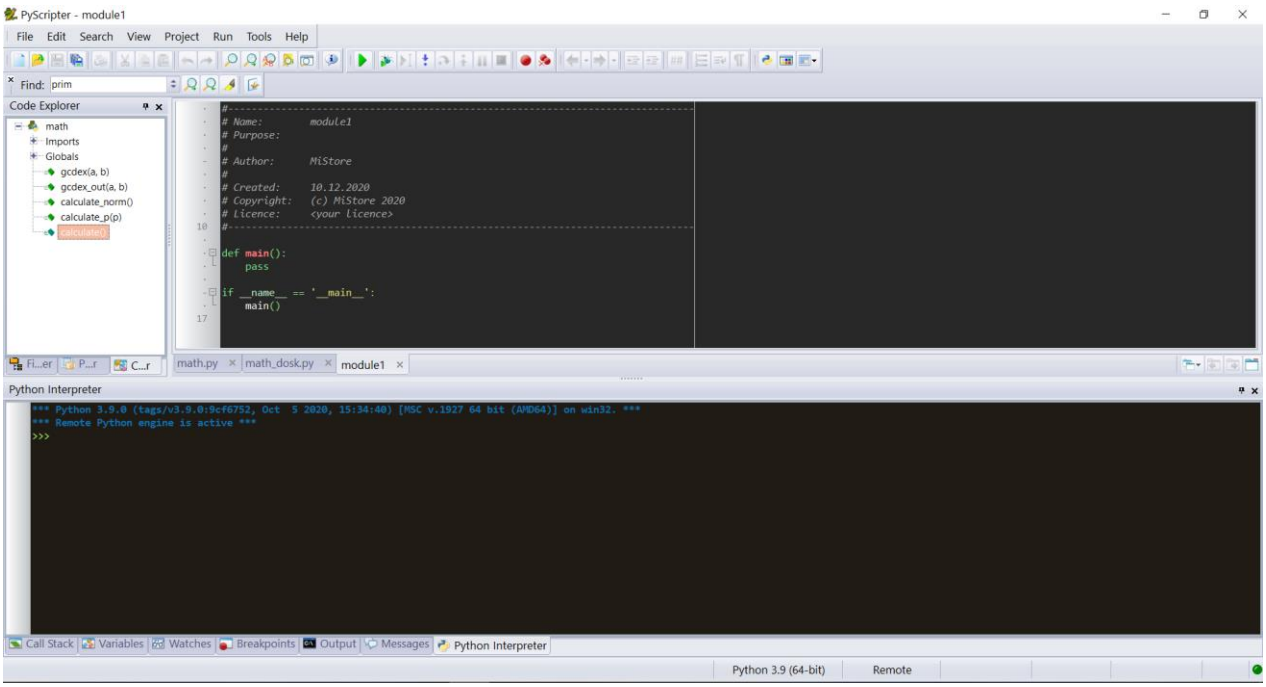

Рисунок 3.1 – Головне вікно програми Python

Числа в Python не відрізняються від звичайних чисел. Вони підтримують

набір математичних операцій, які представлені в таблиці 3.1.

Також потрібно відзначити, що числа в мові Python, на відміну від багатьох інших мов, підтримують довгу арифметику (проте, це вимагає більше пам'яті).

Таблиця 3.1 – Математичні операції в Python

Операція	Позначення операції
Додавання	x+y
Віднімання	x-y
Множення	x*y
Ділення	x/y
Ціла частина від ділення	x//y
Залишок від ділення	x%y
Зміна знаку числа	-x
Модуль числа	abs(x)
Піднесення до степеня	x**y
Піднесення до степеня за модулем	pow(x, y [z])

Отже, Python - це мова програмування з досить ясним і легко читаним кодом. Це пов'язано з тим, що в ньому зведені до мінімуму допоміжні елементи (дужки, крапки з комою), а для поділу синтаксичних конструкцій використовуються відступи від початку рядка. Тому ця мова була вибрана для реалізації звичайної СЗК та МФ СЗК.

3.2 Експериментальне дослідження часових характеристик програмної реалізації множення в системі залишкових класів та її модифікованій формі

Для визначення часових характеристик програмної реалізації операції множення в СЗК і МФ СЗК [46] була обрана високорівнева мова програмування загального призначення Python.

Результати розміщуються у файл з розширенням .csv, ім’я якого записане в останньому рядку головного вікна і включає в себя усі вхідні параметри.

На рис. 3.2 представлені часові характеристики виконання операції множення $N=p\cdot q$ в трьохмодульній СЗК при фіксованому множнику $p=65536$ з двома різними системами модулів (перший випадок - модулі мало відрізняються один від одного: $p_1=1625=\left[\sqrt[3]{65536^2}\right]$, $p_2=1626$, $p_3=1627$ - пунктирні лінії, другий випадок - модулі відрізняються сильно: $p_1=163$, $p_2=1627$, $p_3=16381$ - суцільні лінії) [43, 44].

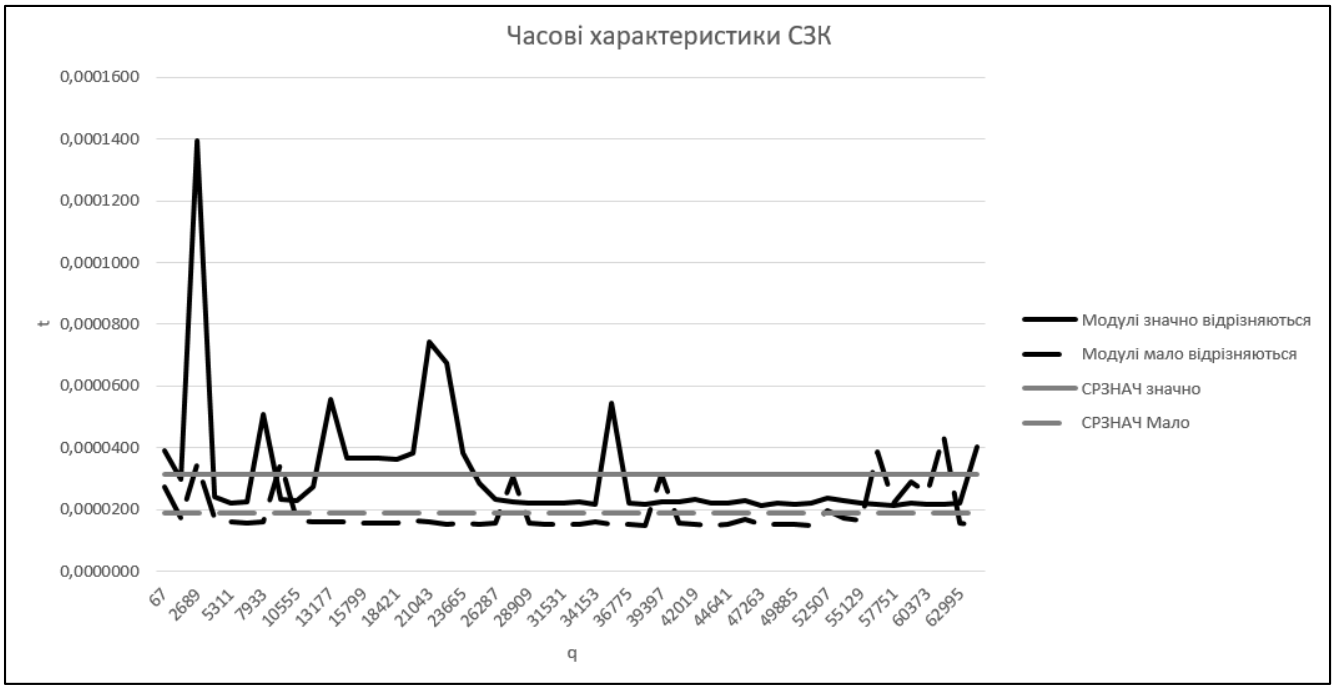

Рисунок 3.2 – Часові характеристики виконання операції множення в трьохмодульній СЗК

Другий множник q змінювався від значення 67 до p з кроком 1311. Останній визначав кількість отриманих обчислень, яке дорівнювало 50. Добуток модулів обох систем перевищує 2^{32}.

Як видно з рисунку 3.2, графік першого випадку носить осцилюючий характер. Середній час виконання операції множення дорівнює 0,009259 мс. У другому випадку, крім початкових значень q, час виконання множення не зазнає

суттєвих коливань. Середній час складає 0,006066 мс, що в 1,53 рази менше, ніж в попередньому випадку. Тому для підвищення швидкодії в СЗК попарно взаємно прості модулі необхідно вибирати так, щоб вони якомога менше відрізнялися один від одного.

Для дослідження МФ СЗК система модулів зі значною різницею між ними ($p_1=651$, $p_2=691$, $p_3=11246$) вибиралася за формулою (2.23), яку можна записати таким чином:

$$p_3 = p_1 + \frac{p_1^2 \pm 1}{p_2 - p_1}. \qquad (3.1)$$

При побудові трьохмодульної МФ СЗК за формулою (3.1) не може бути вибрана система модулів однаковою розрядності. Найменша різниця між модулями буде за такої умови:

$$p_{2,3} = 2p_1 \pm 1. \qquad (3.2)$$

Виходячи із цього, були вибрані такі модулі: $p_1=1025$, $p_2=2049$, $p_3=2051$. Знову ж добуток модулів в обох випадках перевищує 2^{32}.

Вхідні параметри були ті ж самі, що і для звичайної СЗК. Обчислення проводилися згідно такого виразу для МФ СЗК:

$$A = \left(-b_1P_1 + b_2P_2 + b_3P_3\right) \bmod P, \qquad (3.3)$$

де b_i – залишки.

Отримані результати представлені на рис. 3.3. Суцільна лінія показує час виконання множення і середній час для 50 значень p при $p_1=651$, $p_2=691$, $p_3=11246$, пунктирна - відповідно для $p_1=1025$, $p_2=2049$, $p_3=2051$ [33].

Видно, що в обох випадках при малих значеннях q амплітуда коливань велика, при збільшенні q вона зменшується за винятком невеликого відрізка в

другій половині діапазону змін значення q. Середній час для системи модулів $p_1=651$, $p_2=691$, $p_3=11246$ складає 0,002293 мс, а для $p_1=1025$, $p_2=2049$, $p_3=2051$ - 0,002169 мс, що в 1,057 рази менше, ніж в попередньому випадку. Порівняння рисунків 3.2 та 3.3 показує суттєве підвищення швидкодії за рахунок використання МФ СЗК.

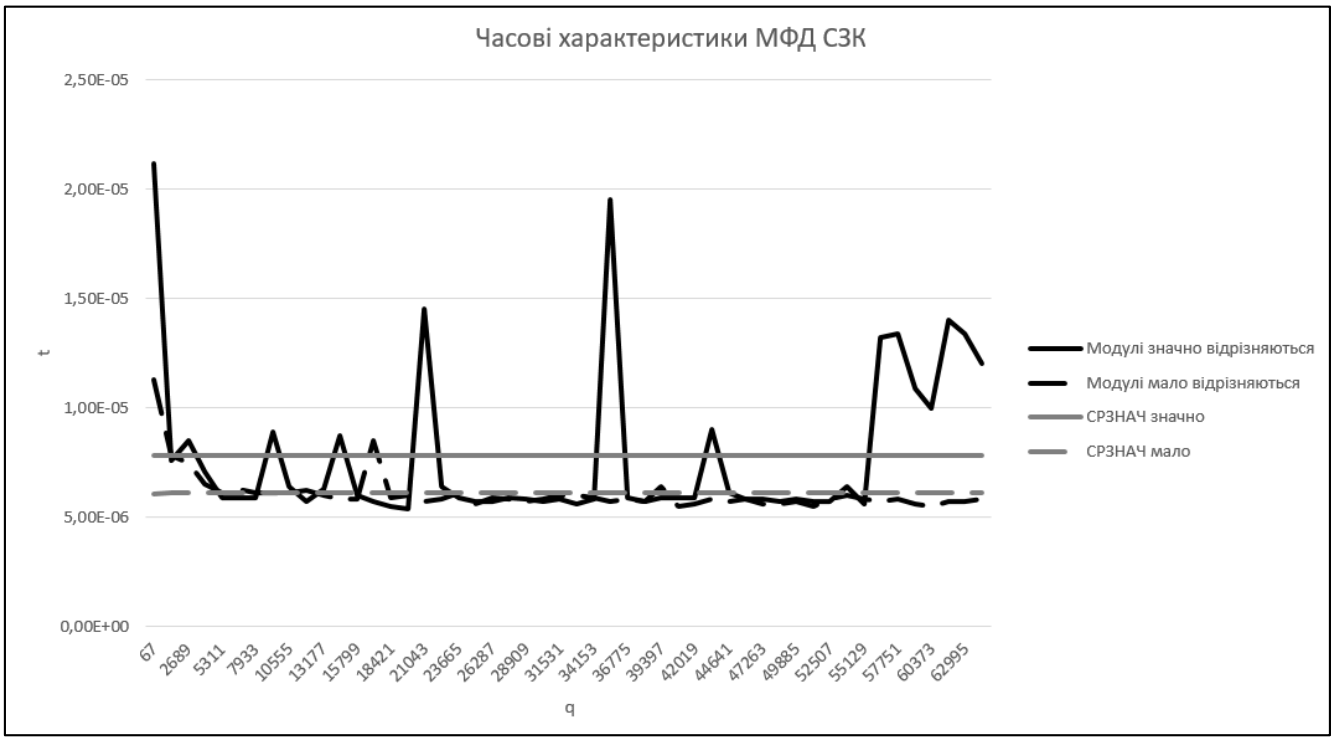

Рисунок 3.3. - Часові характеристики виконання операції множення в трьохмодульній МФ СЗК

Для виключення випадкових впливів на роботу комп'ютера усі обчислення повторювалися 100 разів.

3.2 Опис роботи програмного продукту

Для підрахунку часу виконання операцій можна використати одну зі сторонніх бібліотек, наприклад: timeit.

Робота програми складається з таких етапів:

- знаходження добутку модулів;
- факторизація отриманого добутку;
- перебір всіх можливих комбінацій отриманих чисел;
- обчислення модулів для чисел, що відповідають поставленій умові модифікованій форми системи залишкових класів;
- визначення часових характеристик виконання операції множення.

При реалізації програми потрібно вирішити наступні задачі:

- написати ефективний алгоритм для повного перебору всіх можливих добутків pq;
- розробити універсальну архітектуру, яка буде дозволяти підтримувати окремі реалізації обчислення невідомих модулів при заданій кількості модулів. Для забезпечення універсальності потрібно розробити відповідні інтерфейси;
- розробити клас, що забезпечить відображення часових характеристик виконання операцій множення у системі залишкових класів та модифікованої форми системи залишкових класів.

Розроблена програма являє собою виконуваний файл math.py.

При запуску з’являється вікно, зображене на рисунку 3.4.

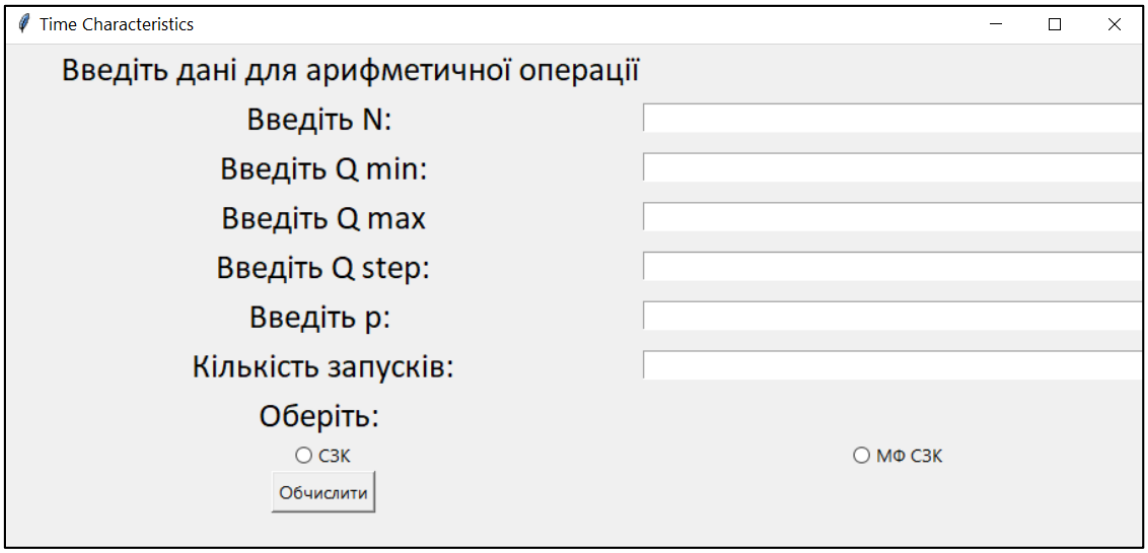

Рисунок 3.4 - Головне вікно програми

Воно містить поля для вводу користувацьких даних для виконання операції

множення, а саме:

- фіксований множник N;
- значення q_{min};
- значення q_{max};
- значення кроку q_{step};
- значення трьохмодульної системи p;
- кількість запусків розрахунку.

Далі користувач обирає систему, у якій буде проводитись розрахунок:

- системі залишкових класів;
- модифікованій формі системи залишкових класів.

Обравши потрібний шлях виконання сценарію розрахунку програми, користувач натискає на кнопку «Обчислити», після чого з’являється повідомлення, що результат розрахунку збережено у файл.

Після натискання на кнопку «Обчислити» результати розміщуються у файл з розширенням .csv, ім’я якого записане в останньому рядку головного вікна і включає в себя усі вхідні параметри. Приклад отриманого файла з часом та результатами множення двох чисел наведений на рисунку 3.5.

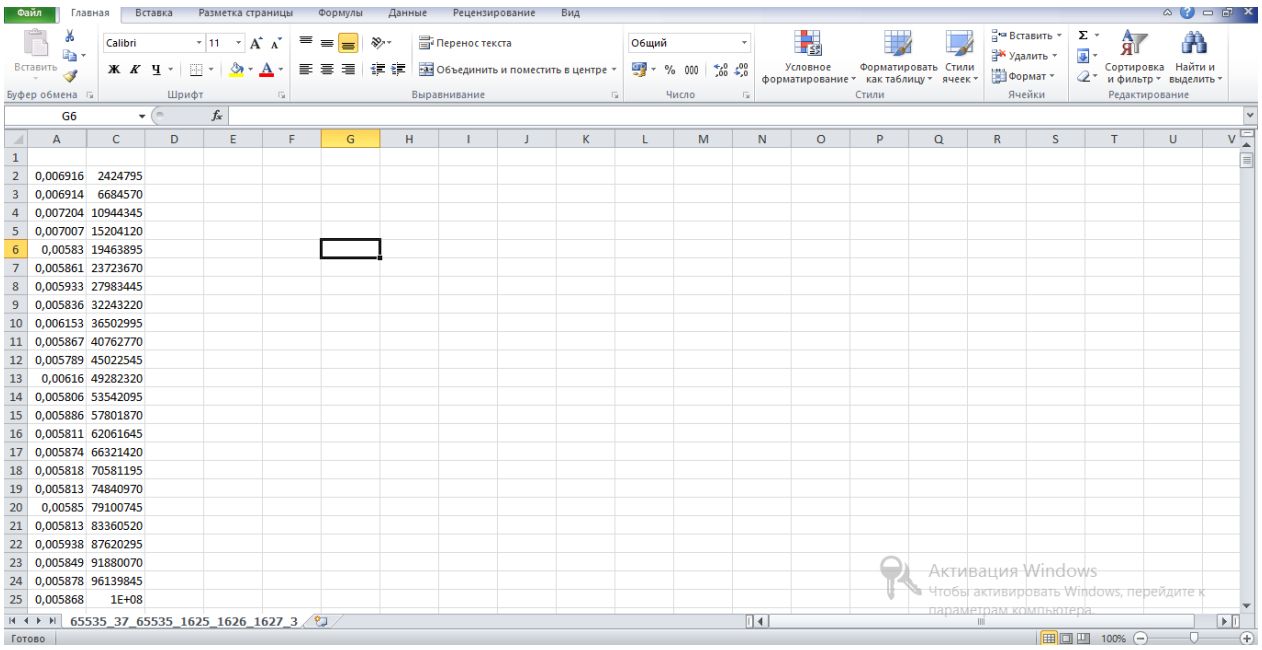

Рисунок 3.5 - Вивід результатів роботи програми

Отже в даному розділі програмно реалізовано множення для системи залишкових класів. В процесі було розроблено алгоритм дослідження експериментально досліджено часові характеристики програмної реалізації множення для системи залишкових класів та її модифікованої форми. Результати показали, що використання модифікованої форми дає можливість зменшити час обчислення арифметичних операцій. Наведено залежності часових характеристик у вигляді графіків, які підтверджують переваги використання модифікованої форми системи залишкових класів.

ВИСНОВКИ

1. Проаналізовано сучасний стан опрацювання багаторозрядних чисел в асиметричних криптосистемах. В результаті з’ясовано, що усі операції у проаналізованих асиметричних криптосистемах відбуваються в двійковій або десятковій системі числення, у яких відсутня можливість розпаралелення процесу обчислень.

2. Розроблено алгоритми побудови трьохмодульної модифікованої форми системи залишкових класів на основі факторизації, що дозволило уникнути процедури пошуку оберненого елемента за модулем при використанні системи залишкових класів.

3. Побудовано програмну реалізацію виконання операції множення у системі залишкових класів та її модифікованій формі .

4. Експериментально досліджено часові характеристики програмної реалізації виконання операції множення у трьохмодульної системі залишкових класів та її модифікованій формі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Касянчук М., Тимошенко Л., Єрьоменко А.

2. Николайчук Я.М. Теоретичні основи побудови і структура спецпроцесорів в базисі Крестенсона / Я.М.Николайчук, О.І.Волинський, С.В.Кулина // Вісник Хмельницького національного університету.- 2007.- №3, Т.1.- С.85-90.

3. Жураковський Ю.П. Теорія інформації та кодування / Ю.П. Жураковський, В.П. Полторак. – К.: Вища школа, 2001. – 255 с.

4. Цымбал В.П. Теория информации и кодирования / В.П. Цымбал. - К.: Вища школа, 1990. - 263 с.

5. Вербицкий О.В. Вступление в криптологию / О.В.Вербицкий. – Львов: Научно-техническая литература, 1998. – 248 с.

6. Бабак В.П., Хандецький В.С., Шрюфер Е.К. Обробка сигналів: Підручник, Либідь, 1996. С.392

7. Сергиенко А.Б. Цифровая обработка сигналов, СПб.: Питер, 2003, 604с.

8. Бородин О.И. Теория чисел / О.И.Бородин. – К.: Высшая школа, 1970. – 275 с.

9. Мельник А. О. Архітектура комп’ютера / А. О.Мельник. - Луцьк: Волинська обласна типографія, 2008. - 470 с.

10. Задірака В.К. Комп’ютерна арифметика багаторозрядних чисел. / В.К.Задірака, О.С. Олексюк. – К.: 2003. – 264 с.

11. Фергюссон Н. Практическая криптография / Н.Фергюссон, Б.Шнайер. – М.: Издательский дом «Вильямс», 2005. – 424 с.

12. Задірака В.К. Комп’ютерна криптологія / В.К.Задірака, О.С. Олексюк. – Тернопіль, Київ, 2002. – 504 с.

13. Рабинович З.Л. Типовые операции в вычислительных машинах / З.Л.Рабинович, В.А.Раманаускас. – К.: Техніка, 1980. –264 с.

14. Торгашев В.А. Система остаточных классов и надежность ЦВМ / В.А.Торгашев. –М.: Советское радио, 1973. - 117 с.

15. Виноградов И.М. Основы теории чисел/ И.М.Виноградов. – Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2003. – 176 с.

16. 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.

17. Касянчук М., Карпінський М., Казмірчук С. Методологія опрацювання багаторозрядних чисел в асиметричних криптосистемах/ Захист інформації. 2019. Т.21, №2. С. 65-73.

18. Касянчук М.М., Николайчук Я.М., Якименко І.З. Теорія алгоритмів перетворень китайської теореми про залишки в матрично розмежованому базисі Радемахера–Крестенсона. Вісник Національного університету «Львівська політехніка» «Комп'ютерні системи та мережі». 2010. №688. С. 118–124

19. 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. – P. 9-18.

20. Kasyanchuk M. Matrix Algorithms of Processing of the Information Flow in Computer Systems Based on Theoretical and Numerical Krestenson's Basis / M.Kasyanchuk, I.Yakymenko, Ya. Nykolaychuk // Proceedings of the X–th International Conference "Modern Problems of Radio Engineering, Telecommunications and Computer Science" (TCSET–2010).–L'viv–Slavske.– 2010. – P.241.

21. 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. - P. 265-273.

22. 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. – P. 649-654.

23. Яциковська У.О. Касянчук М.М., Трембач Р.Б. Удосконалена система захисту комп'ютерної мережі на підставі асиметричного шифрування. Вісник Східноукраїнського національного університету імені Володимира Даля. 2009. №6(136). Ч.1. С. 57–60.

24. Yakymenko I., Kasyanchuk M., Volynskyi O. Fundamental application-oriented tasks in Krestenson base, Methods of effective protection of information flows: collective monograph, By edited V.Zadiraka, Ya.Nykolaichuk, Ternopil: Terno-graf, 2014. P. 149-185. Ch.6

25. Касянчук М.М., Якименко І.З., Дубчак Л.О., Рендзеняк Н.А., Мандебура Н.М. Модифікований метод шифрування Рабіна з використанням різних форм системи залишкових класів. Вісник Хмельницького національного університету. Технічні науки. 2017. №1(245). С. 127-131.

26. Kasianchuk M. Yakymenko I., Pazdriy I., Melnyk A., Ivasiev S. 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): Proceedings of the XIV International Conference. Polyana-Svalyava. 2017. P.222-224 (Scopus).

27. Николайчук Я.Н. Теоретические основы модифицированной совершенной формы системы остаточных классов / Я.Н.Николайчук, М.Н.Касянчук, И.З.Якименко // Кибернетика и системный анализ. - №2. – 2016. – С. 51 – 55.

28. Касянчук М.М. Теорія та математичні закономірності досконалої форми системи залишкових класів. Питання оптимізації обчислень (ПОО–XXXV): Праці Міжнародного симпозіуму. Київ–Кацивелі. 2009. Т.1. С. 306–310

29. Kasianchuk M.N., Nykolaychuk Ya.N., Yakymenko I.Z. 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

30. Касянчук М.М. Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів на основі розв’язку квадратного рівняння. Інформатика та математичні методи в моделюванні. 2016. Т.6, №1. С. 19–25.

31. Касянчук М.М. Аналітичний пошук модулів досконалої форми системи залишкових класів та їх використання в китайській теоремі про залишки / М.М.Касянчук, І.З.Якименко, І.Р.Паздрій, Я.М.Николайчук // Вісник Хмельницького національного університету: Технічні науки. – 2015. - №1. – С. 170-176.

32. Николайчук Я.Н. Теория и методы построения системы модулей модифицированной совершенной формы системы остаточных классов / Я.Н.Николайчук, М.Н.Касянчук, И.З.Якименко // Международный научно-технический журнал «Проблемы управления и информатики». – 2016. - №4. – С. 109-115.

33. Nykolaychuk Ya., Ivas’ev S., Yakymenko I., Kasianchuk M. Test of verification of multidigit numbers on simplicity on the basis of method of vector and modular multiplication. Modern Problems of Radio Engineering, Telecommunica-tions and Computer Science (TCSET–2016): Proceedings of the XIII–th International Conference. L’viv–Slavske. 2016. P.534-536 (Scopus)

34. Nykolaychuk Ya.M., Kasianchuk M.M., Yakymenko I.Z. Theoretical Foundations of the Modified Perfect form of Residue Number System. Cybernetics and Systems Analysis. 2016. Vol. 52, №2. P. 219-223 (Scopus)

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

36. 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. – P. 299–301.

37. Kasianchuk M. Rabin’s modified method of encryption using various forms of system of residual classes / M.Kasianchuk, I.Yakymenko, I.Pazdriy, A. Melnyk, S.Ivasiev // Proceedings of XIV International Conference “The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017)”, 21-25 February, 2017, Polyana-Svalyava. – P.222-224.

38. Касянчук М. М. Досконала форма системи залишкових класів: методи побудови та застосування (Монографія). Тернопіль: Економічна думка (ТНЕУ), 2019. 224 с.

39. Касянчук М.М. Концепція теоретичних положень досконалої форми перетворення Крестенсона та його практичне застосування. Оптико-електронні інформаційно-енергетичні технології. 2010. №2 (20). С. 43-47

40. Касянчук М.М., Якименко І.З., Паздрій І.Р., Николайчук Я.М. Аналітичний пошук модулів досконалої форми системи залишкових класів та їх застосування в китайській теоремі про залишки. Вісник Хмельницького національного університету. Технічні науки. 2015. №1(221). С. 170-176.

41. 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. – P.168-171.

42. 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, Ya.N. Nykolaychuk, I.Z. Yakymenko // Journal of Automation and Information Sciences. – 2016. - Vol.48, №8. - P.56-63.

43. M. Karpiński, S. Ivasiev, I. Yakymenko, M. Kasianchuk, T. Gancarczyk. “Advanced method of factorization of multi-bit numbers based on Fermat's theorem in the system of residual classes”. Proc. of 16th International Conference on Control, Automation and Systems (ICCAS–2016), Gyeongju, Korea, V.1, October, 2016, P.1484–1486.

44. Касянчук М.Н. Построение модифицированной совершенной формы системы остаточных классов с использованием факторизации / М.Н.Касянчук // Радиоэлектроника, информатика, управление. – 2017. – Vol.42, №3. – P.53-59.

45. Николайчук Я.М., Ивасьев С.В., Якименко И.З., Касянчук М.Н. Метод факторизации многоразрядных чисел на основе свойств квадратичности вычетов в системе остаточных классов. Вестник Брестского государственного технического университета. Физика, математика, информатика. 2015. № 5(95). С. 45–45.

46. Касянчук М.М. Дослідження часових харакеристик апаратної реалізації методів пошуку оберненого елемента за модулем / М.М.Касянчук, І.З.Якименко, С.В.Івасьєв, Н.М.Мандебура, В.М.Неміш // Вісник Хмельницько-го національ-ного університе-тету. Технічні науки. – №6 (250). – 2017.– С. 113-120.

47. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. [Електронний доступ] – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page?id_doc=71028

48. ДСП 173-96 Державні санітарні правила планування та забудови населених пунктів [Електроний доступ] – Режим доступу: http://online.budstandart.com/ru/catalog/doc-page?id_doc=46705

49. ДСТУ-Н Б А.3.2-1:2007 Система стандартів безпеки праці в будівництві. [Електронний доступ] – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=6264

50. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний доступ] – Режим доступу: https://zakon.rada.gov.ua/rada/show/v0007282-98#Text