МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп’ютерної інженерії

Цимбал Денис Михайлович

Програмний модуль реалізації криптоалгоритму Рабіна / Software module of Rabin’s cryptoalgorithm implementation

спеціальність: 123 – Комп’ютерна інженерія
освітньо-професійна програма – Комп’ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
Цимбал Денис Михайлович

Науковий Керівник
к.т.н., доцент Дубчак Л.О.

ТЕРНОПІЛЬ-2023

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний модуль реалізації криптоалгоритму Рабіна» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 81 сторінок пояснюючої записки, 12 рисунків, 5 таблиць, 4 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою даної кваліфікаційної роботи є розробка програмного модуля реалізації криптоалгоритму Рабіна для захисту конфіденційних даних.

Методи дослідження включають методи структурного синтезу, теорія графів, елементи математичної логіки.

Розроблено програмний засіб реалізації алгоритму захисту інформації в комп'ютерних системах. Така реалізація здійснена засобами С, що дає можливість реалізації даного проекту на програмному рівні і апаратно за допомогою програмованих логічних інтегральних схем, наприклад сімейства Arduino.

Ключові слова: КОМП'ЮТЕРНА СИСТЕМА, КРИПРОАЛГОРИТМ, КРИПТОАЛГОРИТМ РАБІНА, С++.

RESUME

Qualification thesis “Software module of Rabin’s cryptoalgorithm implementation” of the specialty 123 "Computer Engineering" of bachelor education degree contains 81 pages of explanatory notes, 12 figures, 5 tables, 4 appendixes. The volume of graphic material is 2 sheets of A3 format.

The purpose of this qualification work is the development of a software module for the implementation of Rabin's crypto-algorithm for the protection of confidential data.

Research methods include methods of structural synthesis, graph theory, elements of mathematical logic.

A software tool for implementing the information protection algorithm in computer systems has been developed. Such an implementation was carried out by means of C, which makes it possible to implement this project at the software level and hardware with the help of programmable logic integrated circuits, for example, the Arduino family.

Keywords: COMPUTER SYSTEM, CRYPTHOALGORITHM, RABIN, C++.

ЗМІСТ

					КР.КІ.8351311.00.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив		Цимбал Д.М.			ПРОГРАМНИЙ МОДУЛЬ РЕАЛІЗАЦІЇ КРИПТОАЛГОРИТМУ РАБІНА	Літ.	Арк. 7 / Акрушів
Перевір.		Дубчак Л.О.					
Консульт.		Савка Н.Я.				ЗУНУ,ФКІТ, КІ-41	
Н. Контр.		Мельник Г.М.					
Затвердив		Дубчак Л.О.					

ВСТУП

Можна сказати, що різноманітність і складність проблем інформаційної безпеки в суспільстві та в цілому, що виникають внаслідок процесів розвитку інформаційних технологій, зростає з кожним днем.

Сучасне вирішення багатьох проблем захисту інформації неможливо уявити без використання криптографічних методів. Серед багатьох проблем забезпечення інформаційної безпеки, які вирішуються за допомогою криптографічних методів і засобів, є, мабуть, одна з найактуальніших на сьогодні задача забезпечення цілісності та надійності передачі інформації [3].

Сучасні вимоги до інформаційних систем - це завдання стає все більш серйозною проблемою.

Концепцію криптографії з відкритим ключем запропонували Вітфілд Діффі та Мартін Хеллман, Клод Шеннон. Пароль побудований таким чином, що завдання його злому еквівалентно конкретній математичній задачі, яка вимагає обчислення обсягу, недосяжного для сучасних комп'ютерів. У криптографії відкрилися нові напрямки. Ця робота не тільки суттєво змінила криптографію, але й призвела до появи та швидкого розвитку нових напрямків у математиці. Він заклав основу криптографії з відкритим ключем і теорії криптографічних протоколів [7].

Варто зазначити, що криптоалгоритми з відкритим ключем використовують математичний апарат із теорії чисел, що в майбутньому дасть можливість вирішувати задачі у сфері захисту інформації.

Відповідно до криптосистеми існує багато способів реалізації алгоритмів шифрування та дешифрування інформації. Одним з основних критеріїв вибору криптоалгоритму є простота його реалізації, а також відповідний рівень стійкості.

Метою даної кваліфікаційної роботи є розробка програмного модуля реалізації криптоалгоритму Рабіна для захисту конфіденційних даних.

Реалізація даної мети у вигляді програмного модуля дозволяє побудувати політику захисту інформації як малих, так і великих підприємств чи окремого користувача.

1 АНАЛІЗ СИСТЕМ КРИПТОШИФРУВАННЯ

1.1 Захист інформації за допомогою шифрування

Метою теорії інформації та кодування є пояснення сучасних наукових методів або концепцій та інформаційних технологій, математичне моделювання та дослідження, вивчення основних напрямів досліджень теорії інформації та кодування в інформаційних системах, оволодіння методами. кодування та теорія кодування, оптимальне виявлення та пошук, обробка та захист інформації за наявності перешкод, керування інформаційними потоками в інформаційних мережах [6].

Знання – це поняття, яке має різні значення залежно від контексту. Воно походить від латинського слова «знання», яке має кілька значень:

- пояснення; виклад фактів, подій; переклад;
- уявлення, поняття;
- практика, навчання.

Криптографія(секретне письменство) т. спеціальна система модифікацій звичайного письма, розроблена таким чином, що суть написаного баула може зрозуміти лише той, хто знайомий із системою. Іншим таким виразом є криптографія (грец. kryptos — таємний і graphein — писати).-наука про математичні методи надання конфіденційної інформації [10].

Захист інформації–це точно Набір заходів, вжитих для запобігання (зменшення до безпечного рівня) можливості витоку, викрадення, втрати, розповсюдження, знищення, підробки, фальсифікації або блокування інформації. Для правильної побудови системи захисту необхідно визначити:

1) конкретні дії щодо інформації;

2) що саме являє собою автоматизована система;

3) які загрози безпеці автоматизованих систем;

4) заходи проти загроз безпеці;

5) принципи побудови систем захисту інформації.

Ця наука розвинулася для того, щоб дуже надійно передавати важливу інформацію. Сучасна криптографія характеризується використанням відкритих алгоритмів шифрування, що включає використання обчислювальних пристроїв (відповідно до системи шифрування Рабіна). Відомо більше десятка перевірених алгоритмів шифрування, що робить зашифрований текст недоступним для криптографічного аналізу при використанні достатньо довгого ключа і коректній реалізації алгоритму. Twofish, IDEA, RS4 тощо. широко використовуються алгоритми шифрування.

Інформаційна безпека є одним із засобів захисту інформації від несприятливих впливів і пов'язана з технологічними процедурами забезпечення захисту. Конфіденційність інформації – це ситуація, яка фіксується відповідно до важливості інформації та вимагає певного рівня безпеки. Тому це поняття пов’язане з людьми, особами, які відповідають за інформацію та вирішують, яку інформацію можна розкрити, а яку можна приховати від інших людей [5].

Одним із методів захисту конфіденційної інформації є шифрування. Шифрування дає змогу приховати інформацію від ненавмисних людей, навіть якщо вони бачать сам зашифрований текст.

Відмінність кодів від шифрів

Коди та шифри — це різні способи шифрування повідомлення. Код — це спосіб зміни повідомлення шляхом заміни кожного слова іншим словом, яке має інше значення.

З іншого боку, шифр перетворює повідомлення за допомогою свого алгоритму для перетворення даних, що представляють літери та слова в повідомленні. Шифри легше реалізувати та використовувати з комп’ютерами, оскільки алгоритми автоматизовані та легко програмуються.

Типи шифрів

Шифри можна охарактеризувати по-різному, зокрема:

Блокові шифри шифрують блоки даних однакового розміру.

Потокові шифри можна застосовувати до потоків даних, які часто приймаються та надсилаються через мережу.

Шифри можуть залежати від традиційних ключів, які використовуються безпосередньо для ключового зашифрованого тексту, або від криптографії з еліптичною кривою (ECC). Коли ECC використовується зі 160-бітним ключем, він може забезпечити безпеку традиційного шифру, подібного до того, що використовується в криптосистемі RSA (Rivest-Shamir-Adleman) із використанням ключа довжиною 1024 біти.

Сучасні алгоритми шифрування розроблені таким чином, щоб протистояти атакам, навіть якщо зловмисник знає, який шифр використовується. Історично склалося так, що шифри були менш захищені від атак, оскільки їх використовували для шифрування відкритого тексту вручну, і їх було легше проаналізувати та зламати за допомогою потужності комп'ютера.

Симетричні шифри найчастіше використовуються для захисту онлайн-комунікацій. Вони також включені в багато різних мережевих протоколів, які використовуються для обміну даними. Наприклад, Secure Sockets Layer і TLS використовують шифри для шифрування даних прикладного рівня, особливо при використанні з HTTP Secure (HTTPS).

Віртуальні приватні мережі, які з'єднують віддалених співробітників або віддалені філії з корпоративними мережами, використовують протоколи з симетричними ключовими алгоритмами для захисту передачі даних. Симетричні шифри захищають конфіденційність даних у більшості мереж Wi-Fi, онлайн-банкінгу та послуг електронної комерції, а також мобільного телефону.

Деякі протоколи використовують асиметричну криптографію для шифрування та автентифікації кінцевих точок. Вони також використовують його для захисту обміну симетричними ключами для шифрування даних сеансу. Ці протоколи включають наступне:

- TLS
- HTTPS
- Безпечна оболонка
- Відкрите Pretty Good Privacy
- Безпечні/багатоцільові розширення Інтернет-пошти

Хоча криптографія з відкритим ключем вважається більш безпечною, ніж симетричне шифрування, вона також потребує більше обчислень. З міркувань продуктивності протоколи часто покладаються на алгоритми симетричного ключа для шифрування даних сеансу.

Деякі добре відомі історичні шифри включають наступне:

Цезар. Цей шифр приписують Юлію Цезарю, який, як кажуть, використовував його для безпечного спілкування зі своїми генералами. Це простий шифр підстановки, де кожна літера у відкритому тексті зсувається на певну кількість розрядів у алфавіті. Номер зміни, який використовував Цезар, був три. Шифри підстановки часто реалізуються шляхом запису алфавіту відкритого тексту, причому алфавіт шифрованого тексту написаний над літерами відкритого тексту, зміщений на число, з яким погоджуються ті, хто спілкується. Зсув на три ставить літеру D над відкритим текстом A, E над B тощо. Кількість зміщених символів вважається простою формою ключа.

Атбаш. Цей шифр є шифром підстановки, у якому алфавіт відкритого тексту відображається сам на себе, але у зворотному порядку. Іншими словами, буква відкритого тексту A відображається на зашифрованому тексті Z, B відображається на Y, C на X і так далі. Атбаш названо на честь двох перших і двох останніх літер єврейського алфавіту. Вважається, що він використовувався протягом сотень років.

Проста заміна. Цей також використовувався протягом сотень років. Він замінює кожен символ відкритого тексту на інший символ зашифрованого тексту, в результаті чого фактично є ключ із 26 символів. Він відрізняється від шифру Цезаря тим, що алфавіт шифрування повністю переплутаний, а не просто зміщений однакову кількість місць.

Віженер. Цей шифр є формою поліалфавітної підстановки, тобто він заснований на підстановці з використанням кількох алфавітів підстановки. Шифр Віженера використовує серію переплетених шифрів Цезаря, заснованих на буквах ключового слова. Оригінальний текст зашифровано за допомогою так званого квадрата Віженера або таблиці Віженера.

Омофонічна заміна. Цей шифр підстановки використовує кілька різних літер зашифрованого тексту для заміни окремих літер відкритого тексту. Цей тип шифру зазвичай набагато складніше зламати, ніж стандартні шифри заміни.

Ці історичні шифри все ще актуальні, оскільки вони використовують різні фундаментальні компоненти сучасних шифрів, такі як підстановка та транспозиція.

1.2 Огляд аналогових програм

Відповідно до криптосистеми існує багато способів реалізації алгоритмів шифрування та дешифрування інформації. Одним з ефективних алгоритмів шифрування є RSA система відкритих ключів [2].

Алгоритм RSA був винайдений у 1977 році Р. Рівестом, А. Шаміром і Л. Адлманом. Свою назву цей спосіб шифрування отримав за першими літерами їхніх прізвищ. Суть методу полягає в тому, що, знаючи відкритий текст M, модуль N і показник степеня e, можна визначити. $M^e \bmod(N)$.Функція піднесення до степеня є односторонньою функцією для обчислення коренів і логарифмів. Система RSA використовує той факт, що пошук добутку великих простих чисел не вимагає тривалих обчислень, тоді як розбір добутку двох таких чисел є обчислювально складним завданням [9].

Щоб згенерувати закритий і відкритий ключі, перший користувач випадковим чином вибирає два великих простих числа P і Q, перемножуючи їх, щоб отримати двоскладовий модуль N. Відкритий ключ вибирається у вигляді N і спеціально вибраної основи потужності e, а ключ у вигляді секрету-Числа P і Q. Кожен, хто знає N, може виконати процедуру шифрування, що складається з піднесення до степеня за модулем N [1].

Але тільки той, хто знає P і Q, може розшифрувати текст. Використовуючи числа P і Q, ми можемо визначити значення функції Ейлера

φ(N), яка представляє кількість натуральних чисел від 1 до N, які є простими з N:

$$\varphi(N) = (P - 1)\cdot(S - 1). \quad (1.1)$$

Знаючи φ (N), користувач може вказати таке число d, що:

$$e \cdot d \equiv 1 \bmod(\varphi(N)). \quad (1.2)$$

Якщо криптограма $M^e \bmod(N)$ Якщо підняти його до ступеня d, то в результаті можна отримати чистий текст M:

$$\left(M^e\right)^d = M^{ed} \equiv 1 \bmod(N) \quad (1.3)$$

Алгоритм RSA зручний для користувача. Наступна інформація містить функції введення-виведення:

M– значення, яке використовується для кодування;

p, q – дані, що шифрують число;

C– значення під час шифрування;

R_1–R_4 – Результат виконання алгоритму RSA.

Відповідно до шифрування та дешифрування інформації за алгоритмом Рабіна була визначена аналогова програма, а саме алгоритм RSA. Ці алгоритми в чомусь схожі, але їх відрізняє ряд особливостей:

- швидкість виконання алгоритму;
- вартість готової продукції;
- споживання енергії;
- програмний інтерфейс;
- ефективна обробка даних.

1.3 Формування вимог та постановка задачі

Сфера цієї розробки - додаток дозволяє користувачеві виконувати складні математичні розрахунки для захисту інформації, такі як шифрування та дешифрування даних за допомогою алгоритму Рабіна.

Велике значення для криптографії має система захисту інформації в суспільстві.

Захист інформації – це набір заходів, вжитих для запобігання (зменшення до безпечного рівня) можливості витоку, викрадення, втрати, розповсюдження, знищення, підробки, фальсифікації або блокування інформації.

Шифрування - процес перетворення відкритої інформації (звичайного тексту) у зашифрований текст.

Розшифрування - процес перетворення зашифрованої інформації в читабельну.

Кодування - заміна логічних (смислових) елементів, наприклад, слів.

Пароль називається парою алгоритмів шифрування-дешифрування.

Дія шифру контролюється як алгоритмами, так і, в усякому разі, ключем.

Ключ - це прихований параметр (в ідеалі, відомий лише обом сторонам) для певного контексту під час передачі повідомлення.

Метою роботи є програмний додаток, який використовується для обчислення складних математичних виразів у сфері захисту інформації для отримання системи шифрування з відкритими ключами. Додаток відповідає принципу шифрування та дешифрування даних за алгоритмом Рабіна.

Середовище розробки «MicrosoftVisioStudio» виробництва Microsoft є найкращим вибором для реалізації проекту. Для розробки цього проекту ви повинні використовувати операційну систему Windows XP або новішу.

Компоненти програми, в якій буде розроблятися алгоритм, повинні містити зрозумілі знаки (символи), нотацію, з якої розраховується і виводиться алгоритм.

Результатом цієї роботи має бути чотири розв’язки цього проекту і лише одна з відповідей правильна.

Робоча область повинна бути розроблена у вигляді графічного діалогу, показаного на рисунку 1.1.

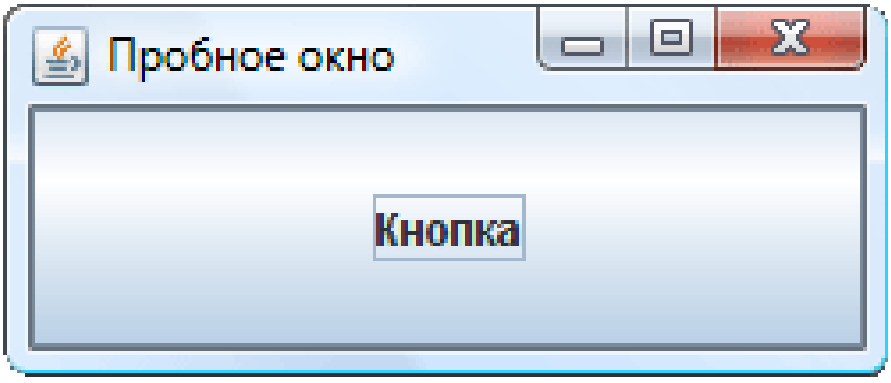

Рисунок 1.1 – Зображення тестової графічної області програмного забезпечення

Програма містить елементи керування у вигляді кнопок для підтвердження операцій шифрування та дешифрування та виходу з програми.

Реалізовано введення та виведення інформації в діалоговому вікні в текстовому форматі.

Програмне забезпечення має:

– бути стійким до некоректних дій користувача (помилки в діях користувача не повинні призводити до збоїв (помилок) у роботі цього програмного забезпечення);

– забезпечити контроль вхідної та вихідної інформації та достовірність введеної користувачем інформації.

Програмне забезпечення розроблено на розподіленій основі операційної системи.

Порядок доступу та захисту програмного забезпечення розроблено на основі:

– запобігати зміні або знищенню інформації;

– запобігання несанкціонованому використанню інформації програмного забезпечення.

Вимоги до програмного інтерфейсу мають бути такими:

– налаштувати колір монтажної області програмного забезпечення (не яскраві кольори) без контрасту для зручного та правильного читання тексту в графічному діалозі;

– заголовний напис містить назву великими літерами відповідним шрифтом;

– елементи робочого простору розташовані відповідно до естетичного вигляду ПЗ.

При розробці даного проекту необхідно скласти кошторис і розрахувати вартість даного продукту, виходячи зі складності завдання та інших аспектів.

Криптографічні шифри працюють із звичайними текстовими блоками. До них пред'являються такі вимоги:

– адекватна криптостійкість;

– простота процедур шифрування та дешифрування;

– прийнятна надійність

Криптостійкість означає час, необхідний для злому пароля з використанням найкращого методу криптоаналізу. Надійність-обмін інформацією, розшифрованою за допомогою деяких криптоаналітичних алгоритмів.

При перетворенні самого пароля слід дотримуватися таких принципів (за К. Шенноном):

– дифузія, тобто зміна будь-якого знака відкритого тексту чи ключа впливає на численні знаки зашифрованого тексту, приховуючи статистичні властивості відкритого тексту;

– плутанина - це використання перетворень між зашифрованим і відкритим текстом, що ускладнює отримання статистичних залежностей.

Для створення системи захисту інформації на основі криптоалгоритму SEAL розроблено дерево рішень (рисунок 1.2).

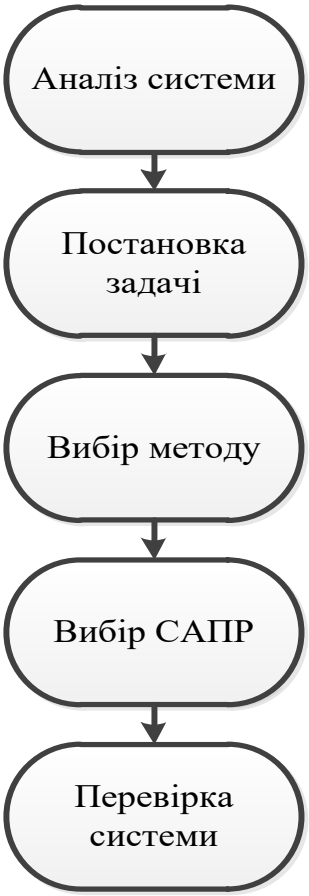

Рисунок 1.2 – Дерево рішень кваліфікаційної роботи

2 МОДЕЛЮВАННЯ АЛГОРИТМІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1Алгоритм Рабіна для шифрування та дешифрування інформації

Криптографічними прикладами захисту інформації є такі спеціальні методи шифрування, кодування або певного перетворення інформації; в результаті його вміст стає недоступним без представлення ключа криптограми та зворотного перетворення. Криптографічний метод захисту, безумовно, є найнадійнішим методом захисту, оскільки сама інформація захищена безпосередньо і до неї немає доступу (наприклад, зашифрований текст неможливо прочитати, навіть якщо носій вкрадено). Існують симетричні та асиметричні алгоритми шифрування [9].

Стандарт шифрування даних (TheDataEncription Standard, DES) був розроблений IVM і затверджений у 1975 році. Він заснований на розборі вихідної інформації на 64-розрядні блоки. Ці блоки інформації підлягають первинній перестановці. Після цього створюється криптограма за допомогою функцій логічного додавання по модулю 2 та інших перестановок. Алгоритм досить складний, але його надійність доведена практикою.

Генерація ключа для схеми шифрування Рабіна виглядає наступним чином:

1) генеруються два великих простих числа p і q - зазвичай однакової довжини;

2) розрахунковий $N = P * Q$;

3) відкритий ключ *A* — *n*, закритий ключ *A* — *P* і *Q*.

Схема шифрування Рабіна шифрує повідомлення *M* для *A*, а потім *A* розшифровує повідомлення.

Кодування інформації, тобто дії користувача *B*:

1) отримати n відкритих ключів від *A*;

2) представити повідомлення *m* як число в діапазоні {0, ..., *n*-1};

3) обчислити $C = M_2 \bmod n$;

4) надсилання зашифрованого повідомлення від *C* до *A*.

Декодування інформації, тобто дії *A*:

1) обчислити квадратний корінь із числа *C*. Нехай це будуть m_1, m_2, m_3, m_4. (якщо НСД(m, n)1, то рівняння С може мати один або два корені);

2) використовуйте кілька корисних інструментів, щоб визначити, який із коренів m_1, m_2, m_3, m_4 є вихідним повідомленням М.

Розглянемо приклад реалізації цього криптоалгоритму.

Генерація ключів. p = 277, q = 331, n = 277 * 331 = 91687.

Кодування. Перед кодуванням 10-бітного повідомлення M = 10011110012 ми присвоюємо останні 6 біт до кінця:

$$M = 10011110011110012 = 40569.$$
$$C = M_2 \bmod n = 405692 \bmod 91687 = 62111$$

Розшифровка. 62111 модуль Обчислюємо квадратний корінь з числа 91687:

$$m_1 = 69654,\ m_2 = 22033,\ m_3 = 40569,\ m_4 = 51118,$$

який у двійковій системі числення має вигляд:

$$m_1 = 10001000000010110, \qquad m_2 = 10101100000100001,$$
$$m_3 = 1001111001111001, \qquad m_4 = 1100011110101110.$$

Початкове повідомлення було $M=m_3$, оскільки останні 6 бітів повторюються лише в одному з коренів.

Проблема дублювання інформації. Одержувач С-кодованого повідомлення стикається з проблемою пошуку вихідного М-повідомлення серед квадратних коренів з m1, m2, m3, m4. Для цього перед кодуванням можна дублювати певну частину даних (наприклад, останні 64 біти). Тоді з високою ймовірністю останні біти будуть скопійовані в один з коренів mi, що вважається переданим М-повідомленням. Якщо жоден з mi не має

повторюваних даних, тоді повідомлення С вважається помилковим і більше не береться до уваги.

Інший спосіб вдосконалення алгоритму Рабіна наведено нижче.

Зверніть увагу, що алгоритм Рабіна є модифікацією алгоритму RSA. Безпека алгоритму Рабіна полягає в складності знаходження складеного числа за модулем квадратного кореня.

Нам потрібно вибрати два простих числа, p і q, які можна порівняти зі значенням 3 mod 4. Ці прості числа є закритим ключем, а їхній добуток:

$$n = p*q\text{- відкритий ключ,} \quad (2.1)$$

$$P \equiv 3 \text{ (режим 4)} \equiv -1 \text{ (режим 4),} \quad (2.2)$$

$$Q \equiv 3 \text{ (режим 4)} \equiv -1 \text{ (режим 4).} \quad (2.3)$$

Ми припустимо, що E постійне і завжди дорівнює числу 2, тоді зашифрований текст повідомлення M обчислюється як:

$$C \equiv M_2 \pmod{n}. \quad (2,4)$$

Розшифровка криптограми С. Введемо допоміжні значення x і y:

$$X \equiv C_k \pmod{p}; \quad (2,5)$$

$$p \equiv C_l \text{ (режим } q) \quad (2,6)$$

де $4k = p + 1$; $4l = q + 1$.

Для x_2 і y_2 отримуємо:

$$x_2 \equiv C_{2k} \pmod{p} \equiv \left[(M^2)^{\frac{q+1}{4}}\right]^2 \bmod q \equiv \text{Режим } M_2 \quad (2,7)$$

$$y_2 \equiv C_{2k1} \pmod{q} \equiv \left[(M^2)^{\frac{q+1}{4}}\right]^2 \bmod q \equiv \text{Режим } M_2\, q \quad (2,8)$$

Отримуємо чотири системи рівнянь для M_1, M_2, M_3, M_4:

$$\begin{cases} M_1 \equiv x \pmod p \\ M_1 \equiv y \pmod q \end{cases} \quad \begin{cases} M_2 \equiv x \pmod p \\ M_2 \equiv -y \pmod q \end{cases} \quad \begin{cases} M_3 \equiv -x \pmod p \\ M_3 \equiv y \pmod q \end{cases} \quad \begin{cases} M_4 \equiv -x \pmod p \\ M_4 \equiv -y \pmod q \end{cases}$$

Один із чотирьох результатів M_1, M_2, M_3 та M_4 є повідомленням M. Вибрати правильне M легко, якщо повідомлення написано словами. З іншого боку, якщо повідомлення є випадковим бітовим потоком (для генерації ключів цифрового підпису), то визначити, який M є правильним, є важким завданням. Один із способів вирішення цієї проблеми - додати відомий заголовок до повідомлення, виконаного перед шифруванням.

Для розуміння та узагальнення цього завдання візьмемо приклад.

Дано p = 3, q = 11, M = 8.

Ми перевіряємо виконання умов:

$$3 \equiv 3 \text{ (режим 4)} \equiv -1 \text{ (мод 4);} \qquad (2,9)$$

$$11\text{-й} \equiv 3 \text{ (режим 4)} \equiv -1 \text{ (режим 4).} \qquad (2.10)$$

Визначимо $n = 3 \cdot 11 = 33$.

Ми шифруємо повідомлення $C \equiv 8^2 \pmod{33} \equiv 31$.

Покупець знаходить:

$k = (3 + 1) / 4 = 1$, $l = (11 + 1) / 4 = 3$, потім користувач обчислює x і y:

$$X \equiv 311 \text{ (режим 3)} \equiv \text{один;} \qquad (2.11)$$

$$p \equiv 313 \text{ (мод 11)} \equiv 3. \qquad (2.12)$$

Додаємо чотири системи рівнянь і визначаємо M_1, M_2, M_3 і M_4:

$$\begin{cases} M_1 \equiv 1 \pmod 3 \\ M_1 \equiv 3 \pmod{11} \end{cases} \quad \begin{cases} M_2 \equiv 1 \pmod 3 \\ M_2 \equiv -3 \pmod{11} \end{cases} \quad \begin{cases} M_3 \equiv -1 \pmod 3 \\ M_3 \equiv 3 \pmod{11} \end{cases} \quad \begin{cases} M_4 \equiv -3 \pmod 3 \\ M_4 \equiv -3 \pmod{11} \end{cases}$$

$M_1=25$, $M_2=19$, $M_3=14$ і $M_4=8$.

Можна зазначити, що кожне число, починаючи з 252 - 25, 19, 14, 8 - може бути основним повідомленням.

2.2 Структурне моделювання програмного засобу

Розробивши певний алгоритм, створюється діаграми за допомогою програмного забезпечення VisualParadigm for UML - це так званий інструмент для проектування програмного забезпечення будь-якої складності.

Пакет UseCaseModelingBin може містити одну або кілька діаграм варіантів використання, які використовуються більше для концептуального рівня дизайну (рисунок 2.1). Панель інструментів цього пакета містить такі специфічні для діаграм візуальні інструменти: актори, варіанти використання, асоціації тощо.

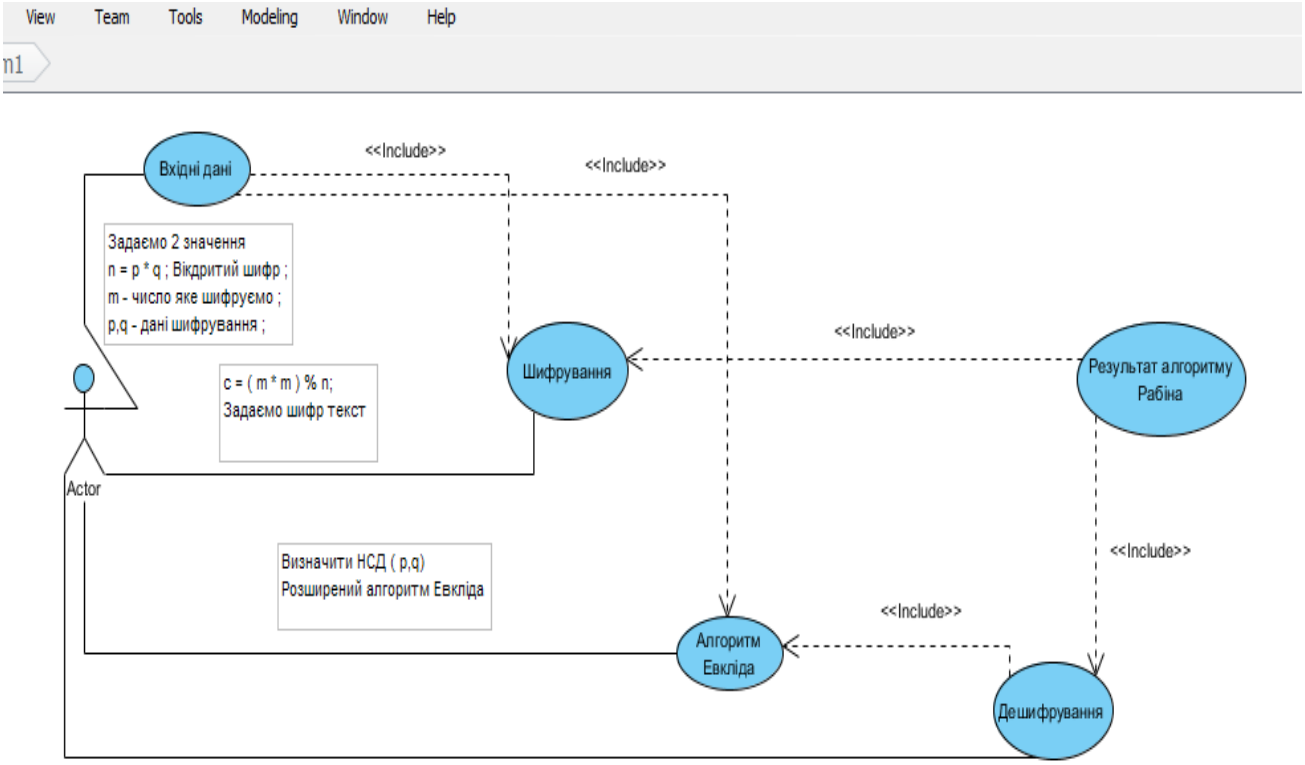

Рисунок 2.1 — Розробка діаграми проекту за допомогою діаграми UML

На рисунку 2.1 показано хід виконання алгоритму Рабіна, тобто послідовність і виконання його кроків для досягнення успішних результатів.

Короткий опис прецедентного сценарію за стандартом RUP такий:

1. Зацікавлені сторони та вимоги прецеденту: ця заява встановлює умови для всіх можливих учасників процесу.

Виконавець повинен правильно та швидко опрацювати дані для кожного кроку поставленого завдання відповідного предмета, тобто «шифрування та дешифрування даних за алгоритмом Рабіна».

2. Передумови (передумови) прецеденту - це завжди список подій, які мають бути виконані перед початком базового сценарію:

– програмна система має бути активною;

– виконання алгоритму Евкліда;

– виконання шифрування та дешифрування;

– розробка 4 алгоритмів варіантів відповідей.

3. Основний успішний сценарій: цей розділ специфікації описує «сценарій успіху», тобто дії, які призводять до успішного завершення подій в основному процесі.

Після проведення всіх відповідних розрахунків підрядник отримує результат (це і буде успішна реалізація даного проекту).

2.3 Діаграми рівнів

Діаграма однорангової послідовності представлена на рисунку 2.2.

Ця структура показує, як саме реалізована система і які дії вона виконує під час роботи.

Прецеденти не можуть бути безпосередньо перетворені у відповідні програмні об'єкти. Для цього слід використовувати певний проміжний рівень опису прецедентів, який дозволяє враховувати деякі особливості наступного

програмного додатку, такі як необхідність інтерфейсу користувача або наявність функцій бізнес-логіки. Одним із таких способів подальшої розробки однорангової моделі є діаграма стабільності.

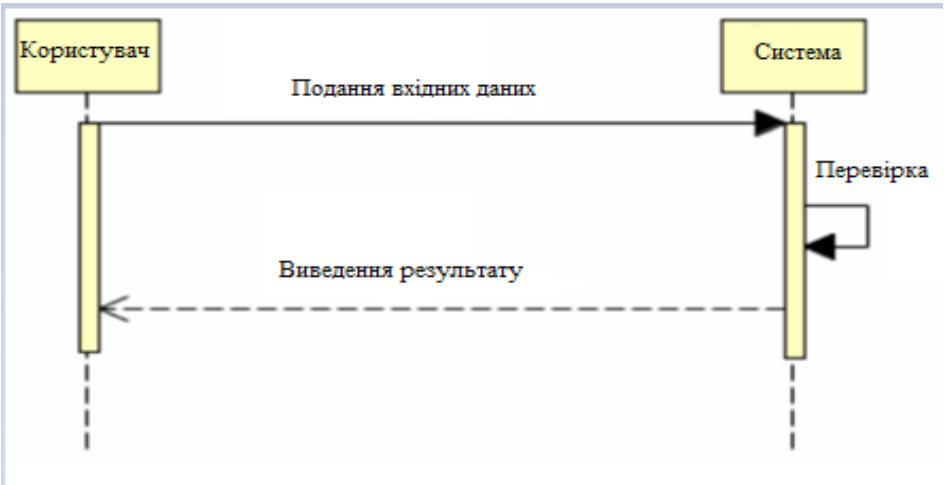

Рисунок 2.2 — Діаграма послідовності для прецеденту цього завдання

Діаграма стабільності розроблена відповідно до шаблону проектування MVC (Model-View-Controller). Тобто система, що розробляється, повинна мати три типи програмних об’єктів:

– модель - це об'єкти, які моделюють дані домену;

– представлення – це об’єкти, які реалізують відображення даних з моделі;

– контролери — це об’єкти, які маніпулюють даними моделі для подальшого перегляду.

Основна ідея шаблону MVC полягає в тому, щоб відокремити дані від їх відображення. Таким чином, якщо виникне необхідність змінити модель даних під час процесу розробки, це жодним чином не вплине на відображення (не потрібно вносити зміни до компонентів View).

На рисунку 2.3 наведено діаграму стійкості програмного продукту алгоритму Рабіна, який дозволяє створювати, додавати та опрацьовувати нову інформацію, введену користувачем у новому запиті програми.

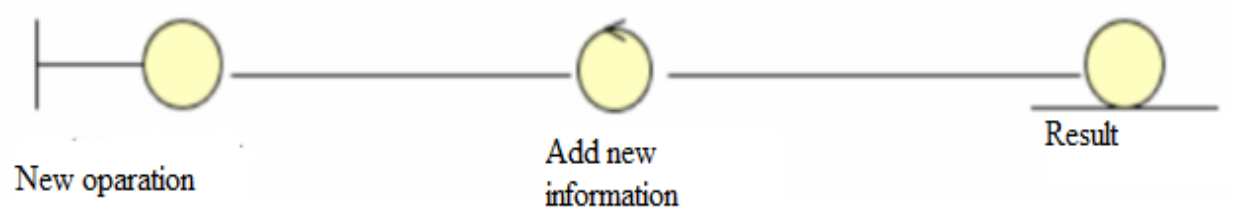

Рисунок 2.3 — Діаграма стабільності для однорангової мережі

Діаграма кінцевого автомата описує процес зміни станів екземпляра певного класу деякої програмної системи. У той же час зміна стану об'єкта може бути викликана зовнішніми діями інших об'єктів. Основна мета цієї діаграми — описати можливі послідовності станів і переходів, які разом характеризують поведінку окремого елемента моделі UML втраченої системи протягом її життєвого циклу. Щоб перейти в режим генерації діаграми кінцевого автомата в середовищі VisualParadigm, потрібно вибрати StateMachineDiagram у вкладці UML.

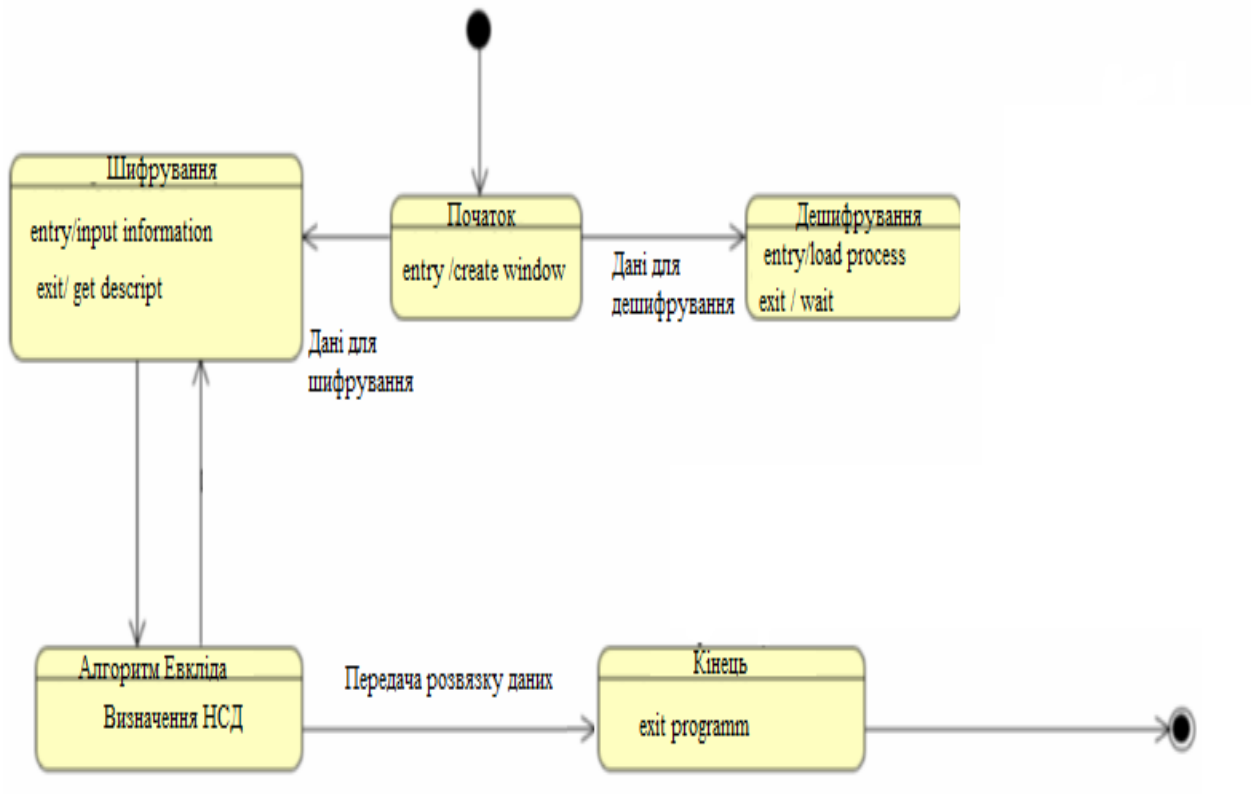

Рисунок 2.4 – Схема кінцевого автомата

Діаграма кінцевого автомата описує процес зміни станів екземпляра певного класу деякої програмної системи. У той же час зміна стану об'єкта може бути викликана зовнішніми діями інших об'єктів. Основна мета цієї діаграми — описати можливі послідовності станів і переходів, які разом характеризують поведінку окремого елемента моделі UML втраченої системи протягом її життєвого циклу. Щоб перейти в режим генерації діаграми кінцевого автомата в середовищі VisualParadigm, потрібно вибрати StateMachineDiagram у вкладці UML.

Реалізація комунікаційної схеми дозволяє тестувати взаємодію різних компонентів із додатком. Основною особливістю комунікаційної схеми є можливість графічного зображення не тільки порядку взаємодії, але й усіх структурних зв’язків між об’єктами програмної системи, що беруть участь у цій взаємодії.

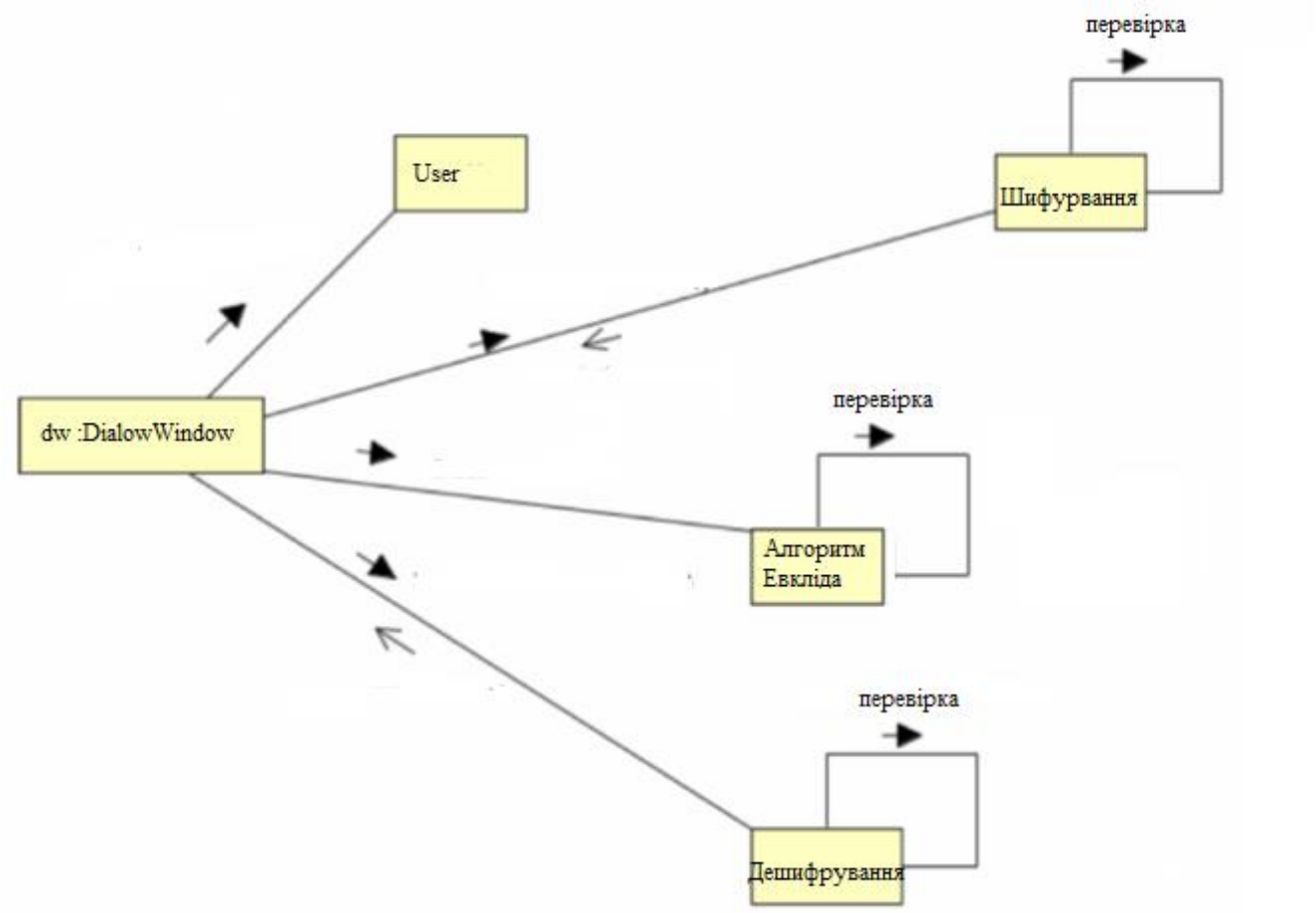

Рисунок 2.5 – Діаграма зв’язку

За цими схемами можна детально вивчити послідовність дій та їх виконання програмно. За своєю природою ця мова об’єктно-орієнтованого програмного забезпечення має дуже обмежену корисність для програмування на основі інших парадигм.

UML – це не метод розробки, іншими словами, конструкції цієї мови не повідомляють вам, що робити спочатку і що робити в останню чергу, і не дають інструкцій для побудови системи, але мова допомагає вам візуалізувати структуру системи . і полегшує співпрацю з іншими розробниками системи.

3 ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАСОБУ

3.1 Опис програмного інтерфейсу інструменту та тестових програм

Програмне забезпечення для кібербезпеки – це програмне забезпечення, яке спрямоване на зупинку складних кібератак. Сучасне програмне забезпечення для кібербезпеки часто містить кілька рівнів інструментів безпеки, включаючи машинне навчання та рішення штучного інтелекту для кібербезпеки, щоб постійно посилювати захист у міру появи нових кіберзагроз. Однак у разі успішної кібератаки програмне забезпечення для кібербезпеки також забезпечує стійкість для захисту електронної пошти, резервного копіювання важливих даних і забезпечення безперервності бізнесу.

Інструменти та програмне забезпечення для кібербезпеки мають важливе значення для захисту організацій від зростаючого зростання потенційно руйнівних злочинних атак, які загрожують безперервності бізнесу та витоку даних, а також від потенційних платежів програм-вимагачів. Хоча продуктивність бізнесу, простої та прямі грошові втрати є серйозними ускладненнями успішної кібератаки, найбільшим впливом на прибутки компанії часто є репутаційна шкода та подальші наслідки, які з цього випливають.

Інструменти та програмне забезпечення кібербезпеки використовують інтегрований набір технологій, елементів керування та процесів для виявлення потенційних ризиків для безпеки мережі та додатків і запобігання їх зараженню системами організації. Інструменти та програмне забезпечення для кібербезпеки також забезпечують безперервність роботи та захист даних у разі успішної кібератаки. Інструменти та програмне забезпечення кібербезпеки також забезпечують навчання кінцевих користувачів виявленню та уникненню потенційних кібератак.

Зіштовхнувшись зі все більш руйнівними кіберзагрозами, багато організацій шукають програмні рішення для кібербезпеки, які можуть допомогти досягти кіберстійкості.

Традиційно програмне забезпечення для кібербезпеки було зосереджено на припиненні атак і запобіганні кіберзлочинності. Але оскільки ландшафт кіберризиків продовжує розширюватися та ускладнюватися, кмітливі компанії шукають програмні рішення, які забезпечують стійкість у разі успішних атак.

Безперервність бізнесу та доступність даних є ключовими факторами кібервідмовостійкості. Оскільки так багато провідних загроз кібербезпеці походять від електронної пошти, більшість компаній, які працюють над кіберстійкістю, потребують програмного забезпечення кібербезпеки, яке може не тільки захистити електронну пошту, але й забезпечити її безперервність і доступність під час і після атаки.

3.2 Кібербезпека розроблено програмного модуля

Для компаній, які шукають найефективніші рішення кібербезпеки, Mimecast пропонує набір хмарного програмного забезпечення кібербезпеки, яке може забезпечити високоефективну стратегію кібервідмовостійкості.

Деякі з основних функцій програмного забезпечення для кібербезпеки включають:

Розширені служби безпеки, які використовують багаторівневі механізми виявлення та найновішу систему аналізу загроз, щоб захистити електронну пошту та користувачів від цілеспрямованих атак і блокувати спам, віруси, зловмисне програмне забезпечення та витік даних.

Безперервність роботи, щоб користувачі могли продовжувати використовувати електронну пошту під час атак і збоїв. Угода про рівень обслуговування на 100% гарантує, що користувачі завжди мають доступ до необхідних даних електронної пошти.

Багатоцільове архівування. Зберігання та реплікація даних електронної пошти, файлів і миттєвих розмов у хмарі допомагає забезпечити доступність і цілісність даних до, під час і після атаки.

Інструменти керування для спрощення резервного копіювання та відновлення, а також відповідності та електронного виявлення.

Навчання кінцевих користувачів та інструменти для підвищення обізнаності про кібербезпеку та надання допомоги співробітникам у створенні потужної лінії захисту від кібератак.

Пам’ятаючи про повний спектр обов’язків сучасних команд безпеки, ось 13 найкращих програм та інструментів для кібербезпеки:

– SiteLock
– SolarWinds Security Event Manager
– Heimdal Security
– Wireshark
– Nagios
– Nessus Professional
– Acunetix
– Snort
– Teramind
– AxCrypt
– Bitdefender Total Security
– TotalAV Cyber Security
– Norton LifeLock

Щоб розпочати моделювання програмного продукту, ви повинні бути знайомі з програмним забезпеченням та його середовищем.

Проаналізувавши основні особливості мови програмування C# та дослідивши структуру та принципи створення програм цією мовою, покажемо найбільш очевидні переваги досліджуваної мови програмування.

Перш за все, слід зазначити, що мова програмування C# претендує на те, щоб бути дійсно об’єктно-орієнтованим (і кожна мовна сутність претендує на те, щоб бути об’єктом).

Крім того, мова програмування C# була розроблена для практичної реалізації компонентно-орієнтованого підходу до програмування, що призводить до меншої залежності кінцевого програмного коду від архітектури

машини, більшої гнучкості переносимості та простоти повторного використання (фрагментів) програм.

При виконанні проекту необхідно застосовувати шифрування і дешифрування за алгоритмом Рабіна, про який йшлося в попередньому розділі.

На рисунку 3.1 зображено графічний інтерфейс даної криптографічної програми Rabin, де можна побачити основні положення та принципи розташування та основні функції вбудованих ключів використання.

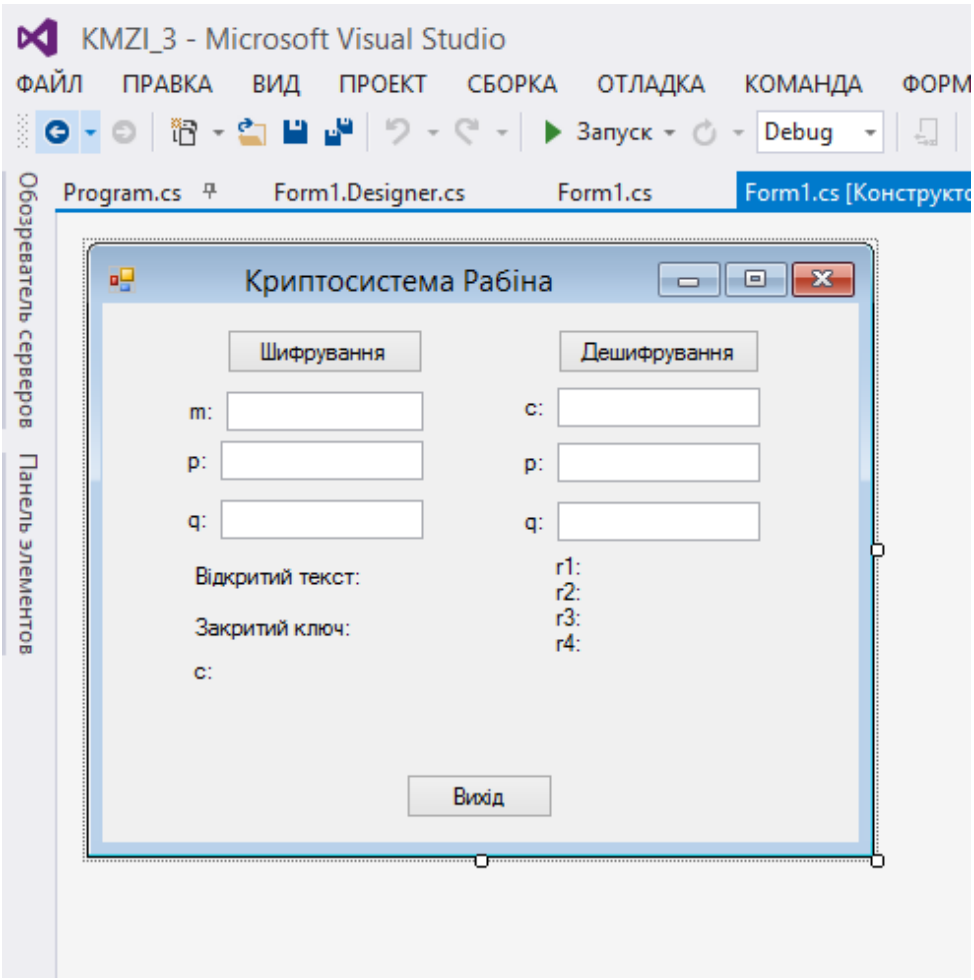

Рисунок 3.1 – Інтерфейс програми

Програмний інтерфейс повинен містити такі елементи керування програмою:

1. Шифрування, визначене формулами та їхніми вхідними даними.

M — це значення, яке ми використовуємо для кодування.

P,Q – дані, що шифрують число.

C - значення M під час шифрування.

R1–R4 є результатом алгоритму Рабіна.

Перше повідомлення m (текст) шифрується відкритим ключем - числом n за формулою:

$$C = режим\ м^2 \qquad (3.1)$$

Завдяки використанню множення за модулем швидкість шифрування системи Рабіна вища за швидкість шифрування методу RSA, навіть якщо в останньому випадку для експоненти вибрано мале значення.

Нехай вихідний текст буде m = 20. Тоді зашифрований текст буде таким: $c = m^2 \bmod n = 20^2 \bmod 77 = 400 \bmod 77 = 15$.

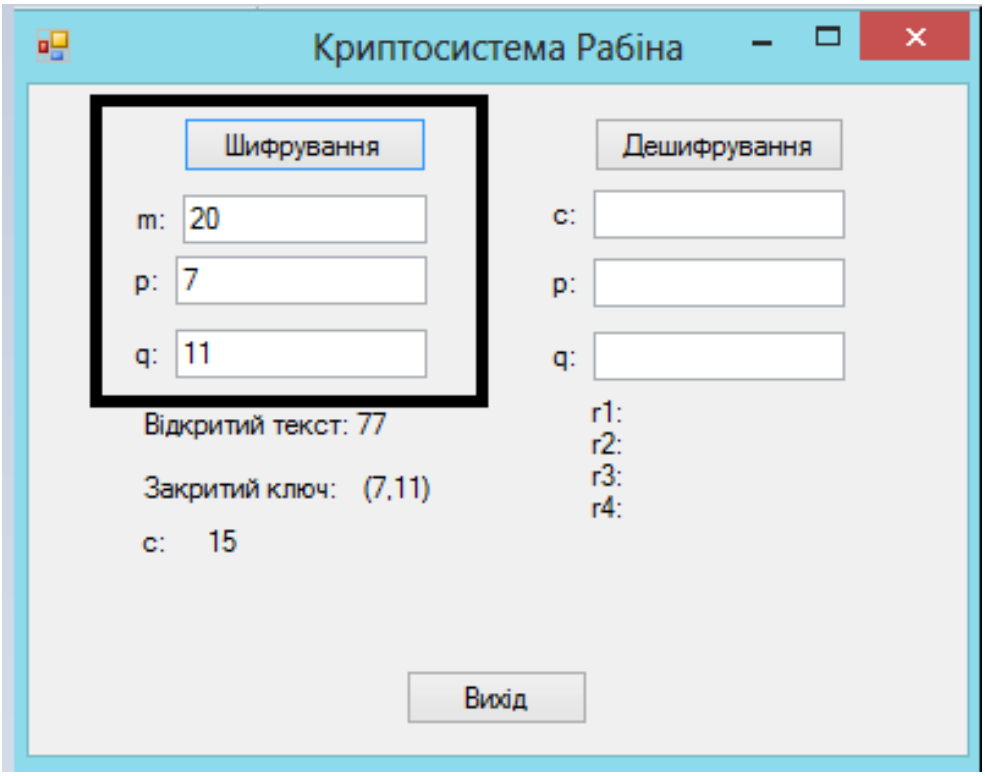

Рисунок 3.2- Шифрування даних

2. Розшифровка відбувається наступним чином.

Для розшифровки повідомлення потрібен спеціальний ключ - цифри P і Q. Процес дешифрування виглядає наступним чином: спочатку за допомогою

алгоритму Евкліда, числа знаходяться з рівняння $y_p y_q$; також, використовуючиКитайська теорема про залишкиОбчисліть чотири числа.

Одне з цих чисел є фактичним відкритим текстом m.

У результаті декодування виходить один із коренів з вихідним текстом m (рисунок 3.4).

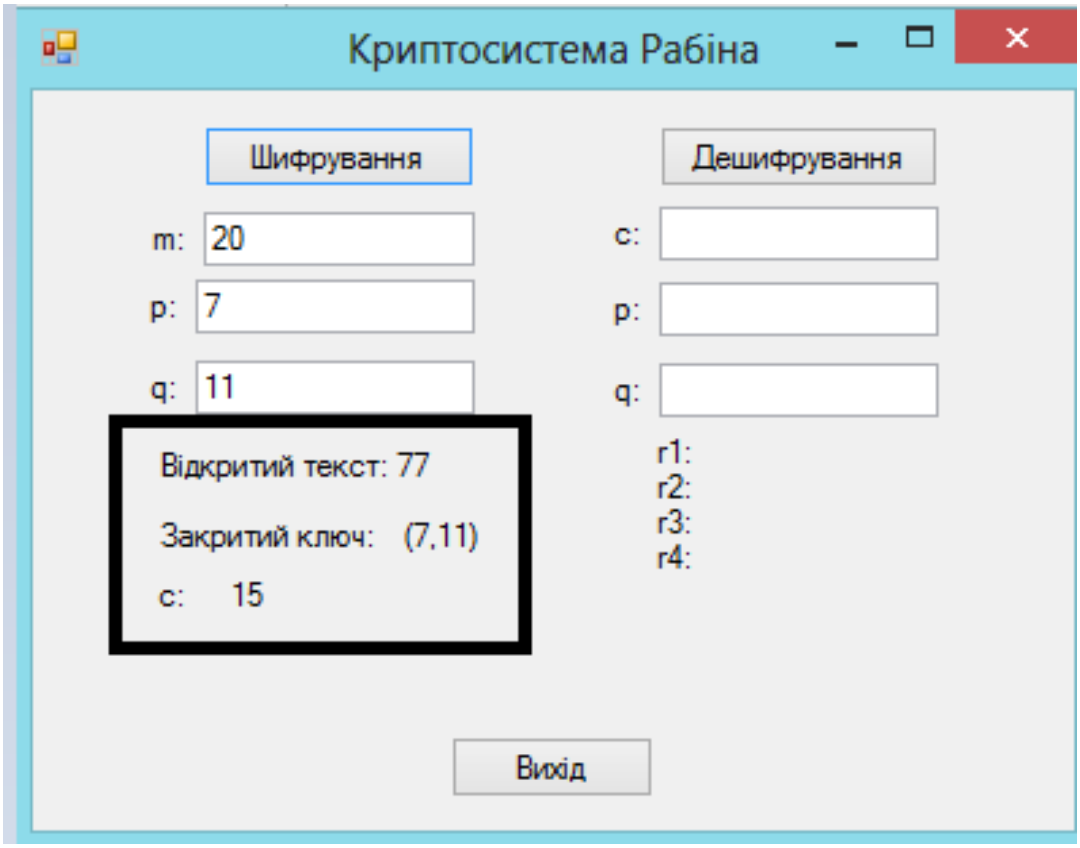

Рисунок 3.3 – Зображення закритого та відкритого програмного перемикача

3. Відкритий ключ знаходиться за формулою N=p*q; (рисунок 3.3)

4. Ми знаходимо закритий ключ за допомогою шифрування CmodP (виконання цієї програми показано на рисунку 3.3)

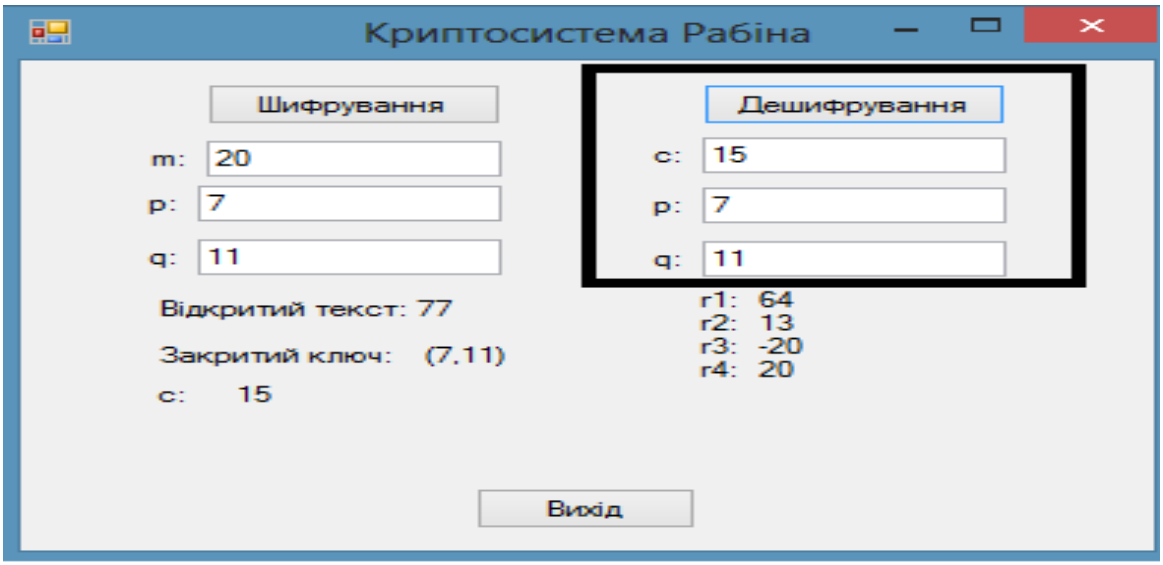

Рисунок 3.4 - Результат шифрування

5. Результат виконання програми записується в смуги r1-r4, зображені на рисунку 3.5.

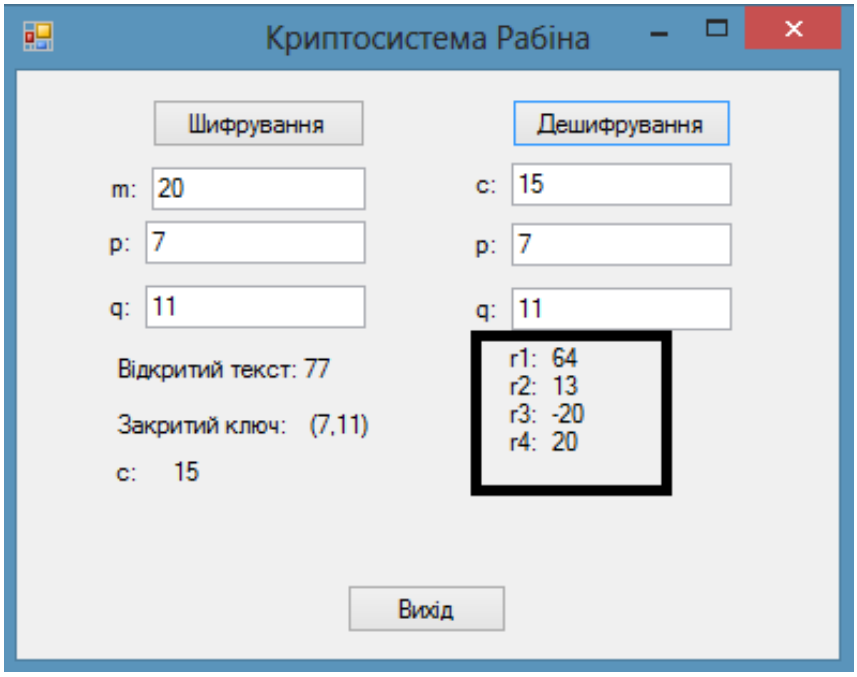

Рисунок 3.5 – Результат виконання програми

6. Вийти з програми (за допомогою клавіші «вихід»);

Після запуску програми ми отримаємо графічний інтерфейс із зображенням запису шифрування та дешифрування, що дозволяє нам працювати та отримувати коректний результат.

Підсумовуючи, хотілося б зазначити, що декодування тексту, крім правильного, призводить до ще трьох неправильних результатів. Це головний недолік криптосистеми Рабіна і один з факторів, що заважає їй знайти широке практичне застосування.

Якщо вихідним текстом є текстове повідомлення, правильний текст визначити неважко. Однак якщо повідомлення є довільним бітовим потоком (наприклад, для генерації ключа або цифрового підпису), то визначення необхідного тексту стає справжньою проблемою. Один із способів вирішення цієї проблеми — додати відомий заголовок або деякі теги до повідомлення перед його шифруванням.

3.3 Порівняння системи шифрування Рабіна з програмою RSA

Алгоритм Рабіна схожий на кодування RSA, але замість того, щоб звести повідомлення до ступеня e, шифрування використовує процес оновлення блоку повідомлення до кадру; це позитивно впливає на швидкість алгоритму без шкоди для криптостійкості.

Для декодування використовується китайська теорема про залишки з двома показниками степеня за модулем. Тут ефективність порівнюється з RSA.

Вибір потрібного тексту з чотирьох текстів спричиняє додаткові витрати на обчислення. І це стало причиною того, що система шифрування Рабіна не отримала широкого практичного застосування. Найбільшою перевагою системи шифрування Рабіна є те, що випадковий текст може бути повністю відновлений лише із зашифрованого тексту. Умови, за яких дешифратор здатний ефективно розкладати відкритий ключ на фактори n. Доведено, що криптосистема Рабіна є

стійкою до атаки типу «все або нічого» на вибраний відкритий текст, лише якщо завдання цілочисельної факторизації складне.

Надійність за принципом «все або нічого» полягає в тому, що надавши текст, зашифрований за допомогою певного алгоритму, зловмисник повинен відновити вихідний блок тексту, розмір якого зазвичай визначається параметром безпеки криптосистеми. З огляду на відкритий і зашифрований текст, зловмисник повинен відновити весь блок секретного ключа. При цьому зловмисник або досягає повного успіху, або не досягає нічого. Слово «нічого» означає, що зловмисник не мав конфіденційної інформації ні до, ні після невдалої атаки.

Система шифрування Рабіна абсолютно вразлива до атак на основі вибраного зашифрованого тексту. Як правило, зловмисник використовує всі доступні йому можливості. Він зв'язується зі зламаним користувачем, надсилає йому зашифрований текст для подальшої розшифровки та запитує повернення оригінального тексту.

Наприклад, додавання надмірності, повторення останніх 64 бітів може зробити корінь унікальним. У цьому випадку алгоритм дешифрування генерує єдиний корінь, уже відомий зловмисникові. Процес додатково вразливий, оскільки в кодуванні використовуються лише квадратні залишки. У прикладі з n = 77 використано лише 23 із 76 можливих випадків.

Що стосується безпеки алгоритму RSA, то він побудований за принципом комплексності розкладання цілих чисел на множники. Алгоритм використовує дваключі— відкритий (public) і приватний (private), відкритий і відповідний закритий ключі разом утворюють пари ключів (key pair). Відкритий ключ не потрібно зберігати в секреті, він використовується для шифрування даних. Якщо повідомлення зашифровано відкритим ключем, його можна розшифрувати лише відповідним закритим ключем. Важливими факторами для порівняння двох систем шифрування шифрування та дешифрування є:

- споживання енергії;
- швидкість виконання транзакцій;
- загальна вартість виробу;

– охорона безпеки;

– використання пам'яті.

Підводячи підсумок цього порівняння двох криптосистем, я хотів би зазначити, що криптосистема Рабіна є більш надійною та ефективною відповідно до наведених вище критеріїв оцінки.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ

В цьому розділі бакалаврської роботи (БР) проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання програмного забезпечення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (К) включають:

$$K = K_1, + K_2 \tag{4.1}$$

де K_1 - витрати на розробку програмних засобів, грн;

K_2 - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп’ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($B_{ОП}$);
- витрати на відрахування у спеціальні державні фонди ($Вф$);
- витрати на покупні вироби ($Пв$);
- витрати на придбання спецобладнання для проведення експериментальних робіт ($Об$);
- накладні витрати ($Н$);
- інші витрати ($Ів$).

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП

обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці програмного забезпечення задіяні наступні спеціалісти - розробники, а саме – керівник проекту, студент-дипломник.

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Погодинна заробітна плата, грн
Керівний дипломної роботи	124 грн/год
Студент	17 грн/год

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} \qquad (4.2)$$

де n_{ij} – чисельність розробників i-ої спеціальності j-го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i-ої спеціальності j-го тарифного розряду, год;

C_{ij} – годинна ставка працівника i-ої спеціальності j-го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_{i}}, \qquad (4.3)$$

де C_{ij} – основна місячна заробітна плата розробника i-ої спеціальності j-го тарифного розряду, грн;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ - місячний фонд робочого часу працівника *i*-ої спеціальності *j*-го тарифного розряду, год (приймаємо 168 год).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавця	Час розробки, год.	Погодинна заробітна плата, грн	Витрати на оплату праці, грн
1	Керівник дипломної роботи	16 годин	124 грн/год	1 984 грн
2	Студент	234 години	17 грн/год	3978 грн
РАЗОМ витрати на розробку				5 962 грн

Відрахування на соціальні заходи:

1) Заробіток керівний дипломної роботи становить 1 984 грн.

ЄСВ (єдиний соціальний внесок) становить 22%

$$1\ 984\ \text{грн} \times 22\% = 436{,}48 \qquad (4.4)$$

Розрахуємо ПДФО (податок на прибуток або на доходи 18%):

$$1\ 984\ \text{грн} \times 18\% = 357{,}12 \qquad (4.5)$$

Також з зарплати має бути утриманий ВЗ (військовий збір 1,5%):

$$1\ 984\ \text{грн} \times 1{,}5\% = 29{,}76 \qquad (4.6)$$

Працівнику має бути перераховано за місяць:

$$1\ 984\ \text{грн} - 436{,}48 - 357{,}12 - 29{,}76 = 1\ 160{,}64 \qquad (4.7)$$

2) Заробіток студент становить 3978 грн.

До розрахунку загального місячного (річного) оподатковуваного доходу платника податку не включається, зокрема, сума стипендій України, призначених законом, постановами Верховної Ради України, указами Президента України

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

Найменування	Виробник (модель)	Одиниці вимірювання	Кількість	Ціна за одиницю, грн	Сума, грн
Контролер	Настінний контролер дистанційного керування Allen & Heath PL-14	Шт.	1	6 786 грн	6 786 грн
Разом					6 786 грн

Якщо для розробки КС використовується електрообладнання, то необхідно розрахувати витрати на електроенергію за формою, наведеною в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп’ютерної техніки

Назва устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнанн, год	Ціна електроенергії, кВт год грн	Сума, грн
Ноутбук	0,17	0,8	234	1,44 грн	57,2
Разом витрати на електроенергію					57,2

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1{,}5 \cdot 5962 = 8943\,(грн.) \quad (4.8)$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 5962 \cdot 0{,}1 = 596{,}2\ (грн) \quad (4.9)$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_Ф + B_{ПВ} + H + I \quad (4.10)$$

$$K_1 = 1847{,}6 + 378{,}76 + 187 + 2771{,}4 + 184{,}76 = 5369{,}86 \quad (4.11)$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від} \quad (4.12)$$

де $S_{м.г.}$ - вартість однієї машино-години роботи ПК, грн/год.;

$t_{від}$ - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 40 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому:

$$K_2 = 5{,}2 \cdot 80 = 416\ грн. \quad (4.13)$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 2.5).

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн
Витрати на оплату праці	1847,6
Відрахування у спеціальні державні фонди	378,76
Витрати на куповані вироби	187
Накладні витрати	2771,4
Інші витрати	184,76
Витрати на відлагодження і дослідну експлуатацію програмного продукту	416
Разом	5785,52

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Річні поточні витрати на експлуатацію програмного забезпечення визначаються за формулою:

$$Вепк = Ве + Ва + Врем + Вдк + Ві , \quad (4.14)$$

де Ва – річні відрахування на амортизацію,

Ве – річні витрати на електроенергію для ПК, Врем – річні витрати на ремонт ПК,

Вдк – річні витрати на додаткові комплектуючі ПК,

Ві – інші витрати.

Отже,

$$Вепк = 1911 + 30912 + 3091,2 + 1030,4 + 2576 = 39520,6 \text{ грн.} \quad (4.15)$$

Суму річних амортизаційних відрахувань визначаємо за такою формулою:

$$Ва = Цпк * На , \quad (4.16)$$

де Цпк – балансова вартість ПК,

На– норма амортизаційних відрахувань (дорівнює 15% у квартал).

Отже,

$$Ва = 51520 * 0,6 = 30912 \text{ грн}, \quad (4.17)$$

де Цпк = 51520 грн ,

На = 4 * 15% = 60% = 0,6 (дорівнює 15% у квартал).

Балансову вартість ПК розраховуємо за формулою:

$$Цпк = Ц р *(1 + Кун), \quad (4.18)$$

де Цр – ринкова вартість ПК,

Кун – коефіцієнт, що враховує витрати на установку й налагодження ПК (приймається рівним 12%).

Отже,

$$Цпк = 46000 *(1 + 0,12) = 51520 \text{ грн}, \quad (4.19)$$

де Цр = 46000 грн (Ноутбук ASUS ZenBook Duo 14 UX482EG-HY419W (90NB0S51-M003H0) Celestial Blue),

$$Кун = 12\% = 0,12.$$

Витрати на електроенергію, що споживає ПК, визначаємо за формулою:

$$Ве = Рпк * Фпк * Це * Рів , \quad (4.20)$$

де Рпк – паспортна потужність ПК,

Фпк – річний фонд корисного часу роботи ПК,

Це – вартість 1 кВт/год електроенергії,

Рів – коефіцієнт інтенсивного використання ПК (0,7 - 1).

Отже,

$$Ве = 0,8 * 1843 * 1,44 * 0,9 = 1911 \text{ грн}, \quad (4.21)$$

де Рпк = 0,8,

Фпк= 1843 год,

Це = 1,44 кВт·год

Рів = 0,9

Таким чином, розрахункове значення витрат на електроенергію, що споживає ПК, складає:

- витрати на поточний і профілактичний ремонт (приймаються рівними 6% від вартості ПК):

$$Врем = Цпк * 0,06 \quad (4.22)$$

Отже,

$$Врем = 51520 * 0,06 = 3091,2 \text{ грн} \quad (4.23)$$

- витрати на додаткові комплектуючі – витрати необхідні для забезпечення експлуатації ПК (приймаються рівними 2% від вартості ПК):

$$Вдк = Цпк * 0,02 \quad (4.24)$$

Отже,

$$Врем = 51520 * 0,02 = 1030,4 \text{ грн} \quad (4.25)$$

- інші витрати, тобто непрямі витрати пов'язані з експлуатацією ПК (приймаються рівними 5-10% від вартості ПК):

$$Вдк = Цпк * 0,05 \quad (4.26)$$

Отже,

$$Врем = 51520 * 0,05 = 2576 \text{ грн} \quad (4.27)$$

4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання - це витрати на придбання і експлуатацію програмного продукту за весь строк його служби:

$$Ц_{С(П)} = Ц_{П} + B_{(E)NPV} \quad (4.28)$$

де $Ц_{п}$ - ціна придбання програмного продукту, грн.

$$Ц_{П} = К(1 + \frac{П_{Р}}{100}) + К_{0} + К_{к} \quad (4.29)$$

де $К$ - кошторисна вартість;

$П_{р}$ - рентабельність;

$К_о$ - витрати на прив'язку та освоєння програного забезпечення на конкретному об'єкті, грн;

$К_к$ - витрати на доукомплектування технічних засобів на об'єкті, грн.

$$Ц_Д = 5785{,}52 \cdot (1 + 0{,}3) = 7521{,}2 \text{ (грн.).} \quad (4.30)$$

Вартість витрат на експлуатацію програмного забезпечення (за весь час його експлуатації), грн:

$$B_{enpv} = \sum_{t=0}^{T} \frac{B_{e\Pi}}{(1+R)^t} \quad (4.31)$$

де B_{en} - річні експлуатаційні витрати, грн;

T - термін служби програмного забезпечення, років;

R - річна ставка проценту банку.

$$B_{e\text{пр}v} = \sum_{t=1}^{5} \frac{4299{,}12}{(1+0{,}08)^t} = 17200{,}15 \text{ (грн)} \quad (4.32)$$

$$B_{e\text{пр}v} = \sum_{t=1}^{5} \frac{6448{,}68}{(1+0{,}08)^t} = 25800{,}2 \text{ (грн)} \quad (4.33)$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

$$Ц_{сп} = 7521{,}2 + 17200{,}15 = 24721{,}35 \text{ (грн)} \quad (4.34)$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{са} = 6500{,}0 + 25800{,}2 = 32300{,}2 \text{ (грн)} \quad (4.35)$$

4.4 Визначення показників економічної ефективності

За міжнародним стандартам для оцінки ефективності розробки ПЗ застосовують такі показники:

- внутрішня норма дохідності;
- чистий приведений дохід;
- рентабельність;
- термін окупності.

Показник внутрішньої дохідності характеризує величину чистого прибутку (чистого валового доходу), що припадє на одиницю інвестиційних вкладень у кожному часовому інтервалі життєвого циклу проекту.

Розрахунок цього показника виконується за такою формулою:

$$\sum_{i=0}^{T}\frac{Di}{(1+q)^i}-\sum_{i=0}^{T}\frac{Ki}{(1+q)^i}=0, \qquad (4.36)$$

де Di - дохід (прибуток) у і-му періоді;

Ki - інвестиційні вкладення в і-му періоді з урахуванням інфляційних процесів;

і - періоди виконання і впровадження проекту;

Т - загальний період (тривалість) життєвого циклу проекту;

q - показник внутрішньої норми дохідності.

Показник інвестиційних вкладень з урахуванням інфляційних процесів обчислюємо за формулою:

$$Ki = \varphi i * Ri, \qquad (4.37)$$

де φі - коефіцієнт інфляції на поточний період;

Ri - інвестиційні платежі в і-му періоді (капітальні вкладення).

Отже,

$$Ki = 1{,}076 * 50000 = 75320грн, \quad (4.38)$$

де φі = 107,6% (коефіцієнт інфляції поданий в таблиці на 2022 рік в сфері іт)

Ri = 70000 грн

Дохід від розробки ПЗ у і-му періоді розраховуємо за формулою:

$$Di = Ji(Bi-Ci), \quad (4.39)$$

де Bi - ціна продажу програмного продукту в і-му періоді;

Ci - собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

Ji - кількість ПЗ.

Отже,

$$Di = 1(80260-64208)= 16052грн, \quad (4.40)$$

де Bi = 80260 грн

Ci = 64208 грн

Ji = 1.

Вартість продажу розробленого продукту розраховують за формулою:

$$Bi = Взаг * (1 + p/100) \quad (4.41)$$

де р - середній рівень рентабельності на поточний період.

Отже,

$$Bi = 64208 * (1 + 25/100)= 80260 грн \quad (4.42)$$

де р = 25%

У практиці середнього бізнесу для визначення ефективності проектних рішень широко використовується показник рентабельності інвестицій. Економічний зміст – характеризує частку чистого приведеного доходу, що припадає на одиницю дисконтованих в період життєвого циклу проекту інвестиційних вкладень.

$$p = \frac{\sum_{i=0}^{T} \frac{D_i}{(1+q_n)^i}}{\sum_{i=0}^{T} \frac{K_i}{(1+q_n)^i}} - 1 > 0. \qquad (4.43)$$

У ринкових умовах при ціновій політиці, що змінюється, показник терміну окупності є одним з головних для підприємств. Він визначається на основі величини капітальних витрат по періодах розробки програмного продукту та величини фактичних чи прогнозних доходів:

$$\sum_{i=0}^{T} Ki = \sum_{i=0}^{T} Di, \qquad (4.44)$$

де Т - термін окупності,

Di - дохід (прибуток) у поточному періоді,

Ki - капітальні витрати у поточному періоді.

Економічна ефективніст полягає у відношенні результату від розробленого програмного продьукту до затрачених ресурсів:

$$E = Di/Взаг \qquad (4.45)$$

Отже,

$$E = 16052 / 64208 = 0,15 \qquad (4.46)$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = 1 / E \quad (4.47)$$

Отже,

$$T = 1/0,15 = 6,6 \text{ років} \quad (4.48)$$

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Враховуючи основні економічні показники, що стосуються розробки програмного продукту, можна зробити висновок, щодо доцільності запропонованої розробки. Якщо отримано суттєвий економічний ефект від розробки програмного продукту, а термін окупності капітальних вкладень не більший 10 років, то така розробка є економічно вигідною та конкурентоздатною на ринку подібних ІТ продуктів.

ВИСНОВКИ

В результаті розробки кваліфікаційної роботи щодо шифрування та дешифрування інформації на основі алгоритму Рабіна, реалізованого за допомогою програмного середовища C#, можна зробити наступні висновки:

1) на основі статистичних даних проведено порівняльну характеристику сучасних криптосистем, що дало змогу виявити переваги та недоліки кожної системи та врахувати їх при розробці даного ресурсу;

2) проведено аналітичні підходи до розробки криптосистеми Рабіна, оцінено сучасні засоби розробки продукту та визначено, що найбільш підходящою мовою для розробки цього модуля перевірки знань є мова програмування C#;

3) на основі основних етапів тесту розроблено алгоритм Рабіна, алгоритм перевірки знань та створено UML діаграми тестового алгоритму. Перевагою розробленого алгоритму є його простота в розумінні та зручність використання;

4) на основі знання основних компонентів криптосистем шифрування та дешифрування, науки криптографії, розроблено програмні модулі програмного продукту.

Розроблений програмний засіб апробований на конференції (додаток Б) та планується до використання (додаток В).

СПИСОК ЛІТЕРАТУРИ

1. Вибір транспортного засобузахист інформації в інформаційній системі / А. В. Баглай, А. О. Новицький, М. В. Захарова. [Електронний ресурс]. – 2007. – Режим доступу: http://www.rusnauka.com/11._NPRT_2007/Informatica/22338.doc.

2. Вень Бо Мао. Сучасна теорія і практика криптографії.– К.:2005 рік.–768 стор.

3. Вербицький О. В. Вступ до криптології / З Першин А. – Львів.: Науково-технічна література, 1998. – 248с.

4. Гайкович В. Безпека електронних банківських систем. - М .: Об'єднана Європа, 1994. - 564 с.

5. Герасименко В. А. Основи захисту комерційної інформації та інтелектуальної власності в підприємницькій діяльності / З. П. Павлов Д. В., Шиверський А. Н. - М.: - 1991р.

6. Діффі У., Хеллман М. Захист і стійкість до неприязні. / РІВЕНЬ. - 1979. - Том 67, Випуск 3. - С. 71-109.

7. Два слова про архівістів. – [електронне зварювання].– 2008 – Режим доступу: http://biblos.org.ua/TOOLS/archS.php

8. Жалдак М.І., Рамський Ю.С. Інформатика. – К.: «Вища школа», 1991

9. Закон України «Про захист інформації в автоматизованих системах» // ВВР -1994. - Номер 32..

10. Інформатика: Комп’ютерні технології. Комп’ютерні технології / За ред. О. І. Пушкаря – К.: Видавничий центр «Академія», 2001. – 696 с.

11. Мафтик С. Механізми захисту в комп'ютерних мережах. - М .: Світ, 1993. - 216 с.

12. Мессі Ж. Л. Введення в сучасну криптографію. / РІВЕНЬ. - 1988. - Том 76, Вип.5. - С.24-42.

13. Національний стандарт України. Інформаційні технології. Криптографічний захист інформації., - К.:–2002.

14. Богуш В.М. Моніторинг систем інформаційної безпеки: навч. посібник [для студ. вищ. навч. закл.] / В.М. Богуш, А. М. Кудін – К.: ДУІКТ, 2006. – 414 с.

15. Менаске Д. Производительность Web-служб. Анализ, оценка и планирование / Менаске Д., Виргилио А. ; пер. с англ. – СПб. : ДиаСофтЮп", 2012. – 480 с.

16. Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с. 4. Терейковський І. Нейронні мережі в засобах захисту комп'ютерної інформації / І. Терейковський. – К. : ПоліграфКонсалтинг. – 2007. – 209 с.

17. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа / А. Ю.Щеглов. – СПб. : Наука и техника, 2012. – 384 с.

18. Харин Ю.С. Математические основы криптологии: учеб.пособ. / Ю.С.Харин, В.И.Берник, Г.В.Матвеев. – Мн.: БГУ, 1999. – 319 с.

19. Зима В. Безопасность глобальных сетевых технологий / В.Зима, А.Молдован, Н.Молдован. – СПб.: БХВ-Петербург, 2003. – 368 с.

20. Щербаков А. Прикладная криптография / А. Щербаков, А.Домашев. – М., 2003. – 404 с.

21. Пономаренко Н.Н. Защита информации в телекоммуникационных системах: учеб. пособ. / Н. Н. Пономаренко. – Х.: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк.авиац.ин-т», 2015. – 40 с.

22. Алферов А.П. Основы криптографии: учеб. пособ. / А.П.Алферов, А.Ю.Зубов, А.С.Кузьмин, А.В.Черемушкин. – М.: Гелиос АРВ, 2002 – 480 с.

23. Басалова Г.В. Основы криптографии / Г.В. Басалова. – М.: Национальный Открытый Университет "ИНТУИТ", 2016. – 283 с. URL: https://www.twirpx.com/file/1907188/

24. Богуш В.М., Юдін О.К. Інформаційна безпека держави / - К.: МК-Прес, 2005.

25. Буряк В.Я. Основи права України / Буряк В.Я., Грищук В.К., Грищук О.В. та ін. / За ред. Ортинського В.Л. - Львів: Оріяна-Нова, 2005.

26. Правове забезпечення інформаційної діяльності в Україні / За загальною редакцією Шемшученка Ю.С. та Чижа І.С. - К.: ТОВ «Видавництво «Юридична думка», 2006.

27. Основи інформаційної бечзпеки. Посібник. /В.А.Лужецький, О.П.Войнович., А.Д.Кожухівський, Л.І.Северин, І.Б.Трегубенко – Черкаси, ЧДТУ, 2008р. – 243 с. – ISBN 978-966-402-035-7

28. Основи інформаційної безпеки. Методичні вказівки до виконання курсової роботи для студентів напряму „Інформаційна безпека” /Укл. І.Б.Трегубенко, О.В. Коваль – Черкаси, ЧДТУ, 2008р. – 44 с.

29. Антонюк А. О. Основи захисту інформації в автоматизованих системах: Навч. посіб. – К: Видавничий дім «КМ Академія», 2003.– 243 с.

30. Головань С.М. Нормативне забезпечення інформаційної безпеки / С.М. Головань, О.С. Петров, В.О. Хорошко, Д.В. Чирков, Л.М. Щербак / За ред. проф. В.О. Хорошка. – К.: ДУІКТ, 2008. – 533 с.

31. Закон України “Про основи національної безпеки України”// Урядовий кур’єр, 30 липня 2003 р. 6. Постанова Верховної Ради України від 16 січня 1997 року N 3/97-ВР “Про затвердження Концепції національної безпеки України”

32. Постанова Кабінету Міністрів України “Про затвердження Концепції технічного захисту інформації в Україні” від 08.10.1997 р.

33. Хорошко В.О. Основи інформаційної безпеки / В.О. Хорошко, В.С. Чередниченко, М.Є. Шелест /За ред. проф. В.О. Хорошка. – К.: ДУІКТ, 2008. – 186 с.

34. Рибальський О.В. Інформаційна безпека правоохоронних органів. Курс лекцій / О.В. Рибальський, В.Г. Хахановський, В.В. Шорошев, О.І.Грищенко, С.В. Сторожев, М.В. Кобець. – К.: НАВСУ, 2003. – 160 с.

35. Иофе В.К. Справочник по акустике / В.К. Иофе, В.Г. Корольков, М.А. Сапожков / Под ред. М.А. Сапожкова. – М.: Связь, 1979. – 312 с.

36. Гоноровский И.С. Радиотехнические цепи и сигналы, ч. 1 / И.С. Гоноровский. – М.: Соврадио, 1967. – 439 с.

37. Каторин Ю.Ф. Большая энциклопедия промышленного шпионажа / Ю.Ф.Каторин, Е.В. Куренков, А.В. Лысов, А.Н. Остапенко. – СПб.: ООО «Издательство Полигон», 2000. – 896 с.

38. Юдін О.К. Інформаційна безпека. Нормативно-правове забезпечення / О.К. Юдін // Підручник. — К. : НАУ, 2016. — 620 с.

39. Нормативно-правове забезпечення інформаційної безпеки: Збірник нормативно-правових документів / Уклад. О.Г. Корченко, Ю.О. Дрейс. — Житомир : ЖВІ НАУ, 2018. — 280 с.

40. Юдін О.К. Захист інформації в мережах передачі даних / О.К. Юдін, О.Г. Корченко, Г.Ф. Конахович // Підручник — К. : Вид-во DIRECTLINE, 2019. — 714 с.

41. Семкин С. Н., Семкин А. Н. Основи інформаційної безпеки об'єктів обробки інформації: Науч.-практ. посібник. Орел: 2018г. –300 с.

42. Завгородний В.И. Комплексная защита информации в компьютерных системах: Учебное пособие. - М: Логос; 2015г. –264с.

43. Азаров Д. Особливості механізму вчинення злочинів у сфері комп'ютерної інформації // Юридична Україна. – 2004. – № 7 (19). – С. 64 – 68

44. Біленчук П.Д., Романюк Б.В., Цимбалюк В.С. Комп'ютерна злочинність. Навч. посіб. – К.: Атака, 2002. – 240 с.

45. Колесник В.А. Розслідування комп'ютерних злочинів. Наук.-метод. посіб. – К.: Вид-во НА СБУ, 2003. – 124 с.

46. Єфіменко А. А. Порівняльний аналіз алгоритму симетричного блокового перетворення «Калина» (ДСТУ 7624:2014) з іншими міжнародними стандартами шифрування даних / А.А. Єфіменко, Є. М. Байлюк, О. А. Покотило //Збірник наукових праць ЖВІ. Випуск 15, С. 156-162.

47. Коркішко Т. Алгоритми та процесори симетричного блокового шифрування / Т. Коркішко, А. Мельник, В. Мельник. – Львів. БаК, 2003. – 168 с.

48. Остапов С. Е. Основи криптографії: навчальний посібник / С. Е. Остапов, Л. О. Валь. – Чернівці: Книги–XXI, 2008. – 188 с.

49. Гапак О. М. Визначення довжини періоду генераторів псевдовипадкових послідовностей на основі регістрів зсуву зі зворотнім зв’язком та перенесення / О.М. Гапак // Моделювання та інформаційні технології. – 2014. – Випуск 73. – С. 92 – 97.

50. NIST SP 800-22rev1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications//National Institute of Standards and Technology Special Publication 800-22rev1a, 2010, -131 p.