МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра загальної математики

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КРИПТОГРАФІЇ ТА КОДУВАННЯ, ЩО ВИКОРИСТОВУЮТЬ АЛГОРИТМИ МОДУЛЯРНОЇ АЛГЕБРИ»**

Виконала: студентка 2 курсу, групи 8.1119
спеціальності 111 математика
(шифр і назва спеціальності)
освітньої програми математика
(назва освітньої програми)
Л.М. Левкун
(ініціали та прізвище)
Керівник завідувач кафедри загальної математики, доцент, к.ф.-м.н. Зіновєєв І.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри фундаментальної математики, доцент, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя
2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра загальної математики

Рівень вищої освіти магістр

Спеціальність 111 математика
(шифр і назва)

Освітня програма математика

ЗАТВЕРДЖУЮ

Завідувач кафедри загальної математики, к.ф.-м.н., доцент
_____ Зіновєєв І.В.
(підпис)

« 22 » 05 2020 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Левкун Людмилі Михайлівні
(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Дослідження та практична реалізація алгоритмів криптографії та кодування, що використовують апарат модулярної алгебри

керівник роботи (проекту) Зіновєєв Ігор Валерійович, к.ф.-м.н., доцент.
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 20 » 05 2020 року № 576-С

2. Строк подання студентом роботи 11.12.2020 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості з криптографії та кодування.
3. Розробка першого, другого та третього розділів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 22.05.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	22.05.2020-29.05.2020	
2.	Збір вихідних даних.	29.05.2020-10.06.2020	
3.	Обробка методичних та теоретичних	10.06.2020-16.07.2020	
	джерел.		
4.	Розробка першого розділу.	16.07.2020-03.08.2020	
5.	Розробка другого та третього розділів.	03.08.2020-29.10.2020	
6.	Оформлення та нормо контроль	29.10.2020-14.12.2020	
	кваліфікаційної роботи.		
7.	Захист кваліфікаційної роботи.	14.12.2020-19.12.2020	

Студент _____ Л.М. Левкун
(підпис) (ініціали та прізвище)

Керівник роботи _____ І.В. Зіновєєв
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г. Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Дослідження та практична реалізація алгоритмів криптографії та кодування, що використовують апарат модулярної алгебри»: 56 с., 15 рис., 25 джерел.

АЛГОРТМ, ВІДКРИТИЙ КЛЮЧ, ЕЛЕКТРОННИЙ ЦИФРОВИЙ ПІДПИС, КОДУВАННЯ, КРИПТОСИСТЕМА, КРИПТОГРАФІЯ, ПОЛЕ, СЕКРЕТНИЙ КЛЮЧ, СХЕМА, ФАКТОРИЗАЦІЯ, ХЕШ-ФУНКЦІЯ, ШИФРУВАННЯ.

Об’єкт дослідження – алгоритми криптографії та кодування.

Мета роботи: ознайомитись з основними поняттями криптографії та кодування, дослідити алгоритми криптографії, кодування та використання апарату модулярної алгебри в цих алгоритмах, навести приклади практичної реалізації конкретних алгоритмів криптографії та кодування.

Методи дослідження: аналіз, синтез.

У кваліфікаційній роботі розглянуто основні поняття криптографії та кодування, досліджено сучасні криптосистеми та способи їх класифікації, розглянуто основні поняття електронного цифрового підпису, циклічного кодування та окремих криптосистем. Розглянуто основні проблеми, які стосуються використання алгоритмів криптографії та кодування. На основі цього матеріалу досліджено окремі алгоритми електронного цифрового підпису, алгоритми циклічного кодування та наведена програмна реалізація цих алгоритмів в системі Maple.

SUMMARY

Master’s Qualification Thesis «Research and the Implementation of the Cryptography and Coding Algorithms, which Use the Modular Algebra Tool»: 56 pages, 15 figures, 25 references.

ALGORITHM, CLEAR KEY, ELECTRONIC DIGITAL SIGNATURE, CODING, MATRIX, CRYPTOSYSTEM, FIELD, PRIVATE KEY, SCHEME, FACTORIZATION, HASH-FUNCTION, ENCRYPTION.

The object of the study: cryptography and coding algorithms.

The aim of the study is learn the basic concepts of cryptography and coding, to investigate algorithms of cryptography, coding and the use of modular algebra in these algorithms, give examples of practical implementation of specific cryptography and encoding algorithms.

The method of research are analysis, synthesis.

The basic concepts of cryptography and coding are considered, modern cryptosystems and methods of their classification are investigated, the basic concepts of electronic digital signature, cyclic coding and separate cryptosystems are considered.in the qualification work. The main problems concerning the use of cryptography and coding algorithms are considered. On the basis of this material separate algorithms of electronic digital signature, algorithms of cyclic coding are investigated and software realization of these algorithms in Maple system is resulted.

ЗМІСТ

ВСТУП

Розвиток і широке застосування інформаційних технологій в промисловості, керуванні, зв’язку, наукових дослідженнях, сфері послуг, комерційної, фінансової та інших сферах людської діяльності є в даний час пріоритетним напрямком науково-технічного прогресу. У міру розвитку і ускладнення засобів, методів і форм автоматизації процесів обробки інформації підвищується залежність суспільства від ступеня безпеки використовуваних ними інформаційних технологій. Розв’язання проблеми забезпечення безпеки електронної інформації здійснюється на сучасному етапі розвитку комп’ютерної технології криптографічними методами: шифрування інформації, контроль цілісності даних, аутентифікація, генерація секретних ключів.

В книгах [1] та [11] було досліджено криптосистеми, що базуються на модулярній алгебрі. Робота [13] була присвячена збільшенню швидкості шифрування великих об’ємів мультимедійних даних за рахунок поліноміальної системи класів лишків. В [15] було розглянуто проблема криптографічного захисту інформації, проаналізовано різноманітні криптографічні алгоритми.

Проаналізувавши літературу, можна побачити що вчені вже не одне сторіччя вивчають та розв’язують проблеми захисту інформації, але через блискавичний розвиток комп’ютерних технологій ця проблема залишається актуальною.

Метою кваліфікаційної роботи є дослідження окремих алгоритмів криптографії та кодування в яких застосовуються апарат модулярної алгебри. В роботі представлений огляд сучасних криптосистем, огляд алгоритмів криптографії та кодування, а також їх програмна реалізація в СКА Maple.

1 ОСНОВНІ ПОНЯТТЯ КРИПТОГРАФІЇ, КОДУВАННЯ ТА МОДУЛЯРНОЇ АЛГЕБРИ

1.1 Поняття модулярної алгебри. Квадратичні лишки

Модулярна алгебра – це розділ алгебри, в якому закладено всі основні алгебраїчні особливості модулярної теорії. Під задачами криптографії та кодування з використанням модулярної алгебри, що розглядаються в даній роботі будемо розуміти задачі криптографії та кодування над алгебраїчними структурами (кільцями та полями), а також задачі, що використовують апарат алгебри над алгебраїчними структурами (квадратичні лишки за простим та зіставним модулем, розв'язання алгебраїчних рівнянь з квадратичними лишками та ін.).

Розглянемо квадратичні лишки за простим модулем. Якщо рівність за модулем

$$x^2 \equiv a(mod\ p), p \geq 3, (a,p) = 1 \qquad (1.1)$$

має розв'язок, тоді число a називають *квадратичним лишком за модулем* p. В іншому випадку число a називають *квадратичним нелишком за модулем* p. [1]

Дискретним квадратним коренем за модулем p [1] називають вираз вигляду

$$x = \sqrt{a}(mod\ p), x \in \mathbb{Z}_m.$$

Теорема 1.1 Якщо a – це квадратичний лишок за модулем p, то рівність за модулем (1.1) має два розв'язки. [9]

Теорема 1.2 Зведена система лишків за модулем p

$$-\frac{p-1}{2}, \dots, -2, -1, 1, 2, \dots, \frac{p-1}{2}$$

містить $\frac{p-1}{2}$ квадратичних лишків, які рівні за модулем з числами $1^2, 2^2, \dots, \left(\frac{p-1}{2}\right)^2$ і $\frac{p-1}{2}$ квадратичних нелишків.[1]

Теорема 1.3 Якщо a – це квадратичний лишок за модулем p, то $a^{\frac{p-1}{2}} \equiv 1\ (mod\ p)$ або ж $a^{\frac{p-1}{2}} - 1 \equiv 0\ (mod\ p)$. Якщо a – це квадратичний нелишок за модулем p, то $a^{\frac{p-1}{2}} \equiv -1\ (mod\ p)$ або ж $a^{\frac{p-1}{2}} + 1 \equiv 0\ (mod\ p)$.[7]

Нехай $a \in \mathbb{Z}, p$ – непарне просте число, що не є дільником a. Символ Лежандра $\left(\frac{a}{p}\right) = 1,$ якщо a – це квадратичний лишок за модулем p, і $\left(\frac{a}{p}\right) = -1,$ якщо a – це квадратичний нелишок за модулем p. Число a називають чисельником символу Лежандра, а число p – знаменником символу Лежандра. [19] Символ Лежандра має такі властивості:

а) $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}}(mod\ p)$;

б) якщо $a \equiv a_1(mod\ p)$, то$\left(\frac{a}{p}\right) = \left(\frac{a_1}{p}\right)$;

в) $\left(\frac{1}{p}\right) = 1$;

г) $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$;

д) $\left(\frac{abc\dots l}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right) \cdot \left(\frac{c}{p}\right) \cdot \dots \cdot \left(\frac{l}{p}\right)$.

Нехай $P > 1 -$ непарне просте число, $P = p_1 p_2 \dots p_r -$ це факторизація числа P на прості множники, $(a, P) = 1$, тоді символом Якобі називають вираз [19]

$$\left(\frac{a}{P}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right) \dots \left(\frac{a}{p_r}\right).$$

Символ Якобі має властивості, аналогічні властивостям символа Лежандра.[19]

Розглянемо квадратичні лишки за зіставним модулем. Дано рівність за модулем

$$x^2 \equiv a(mod\ p^\alpha), p > 3, \alpha > 0, (a, p) = 1. \qquad (1.2)$$

Теорема 1.5 Якщо рівність за модулем (1.1) має розв'язок (a – це квадратичний лишок за модулем p), то рівність за модулем (1.2) має два розв'язки; якщо рівність за модулем (1.1) немає розв'язків (a – це квадратичний нелишок за модулем p), то рівність за модулем (1.2) немає розв'язку. [1]

Теорема 1.6 Якщо рівність за модулем $x^2 \equiv a(mod\ 2^\alpha), \alpha \geq 1, (a, 2) = 1$ має розв'язок, то $a \equiv 1(mod\ 2), \alpha = 1$; $a \equiv 1(mod\ 4), \alpha = 2$; $a \equiv 1(mod\ 8)$, $\alpha \geq 3$.[9]

Зауваження. Наявність розв'язку рівності за модулем $x^2 \equiv a(mod\ m)$ можна встановити за значенням символу Якобі:

$$\left(\frac{a}{m}\right) = \begin{cases} 1, то\ рівність\ за\ модулем\ x^2 \equiv a(mod\ m)\ має\ розв'язок, \\ -1, то\ рівність\ за\ модулем\ x^2 \equiv a(mod\ m)\ немає\ розв'язку. \end{cases}$$

Алгоритм з поліноміальною часовою складністю розв'язання рівності за модулем $x^2 \equiv a(mod\ m)$ невідомий. Існує припущення, що такого алгоритму не існує. Алгоритм перебору при великих m (з довжиною десятиричного запису m порядку 120 цифр) практично не здійсненний. [1]

1.2. Основні поняття кодування та криптографії. Класифікація криптосистем

Кодування – це процес перетворення повідомлення на впорядкований набір символів, елементів, знаків. При кодуванні кожному повідомленню ставиться у відповідність зумовлена кодова комбінація – набір символів

(елементів, знаків) з деякої скінченної кількості їх, яка називається *алфавітом*. [12]

Код – це сукупність кодових комбінацій, які побудовані за одним алгоритмом кодування. Коди поділяються на двійкові та недвійкові. Алфавіт двійкових кодів складаються із символів 0 та 1. Недвійкові (багатопозиційні) коди – це коди, алфавіт яких містить більше двох символів.

За функціональним призначенням коди поділяють на *безнадмірні* (первинні, прості) та *надмірні* (коригувальні). Перша група кодів призначена для економного кодування інформації – стиснення. Друга використовується для виявлення та/або виправлення помилок, що виникають у процесі передачі даних каналом зв'язку із завадами.[12]

Криптографія – це область математики, основним завданням якої є захист інформації від несанкціонованого доступу. Захист досягається шляхом перетворення вихідного набору даних в такий вигляд, з якого отримання вихідної інформації неможливе або важкодосяжне. Тобто для того щоб отримати вихідну інформацію потрібні достатньо складні обчислення, які займуть тривалий період часу. Отримання вихідної інформації досягається шляхом застосування спеціальної послідовності даних та алгоритмів. Під *ключем* розуміється частина системи, яку можна легко змінити, що зберігається в таємному місці і вона визначає одне з можливих перетворень криптотексту.

Криптосистема – сукупність перетворень, що вибираються за допомогою ключа, які трансформують блок інформації, що знаходиться під захистом, в шифрограму і назад. Множина послідовностей символів або цифр, які формують вхідні дані і з якими функціонує система безпосередньо при шифруванні, називається *алфавітом відкритого тексту,* а при дешифрування – *алфавітом криптотексту*.[4] Всі сучасні криптосистеми можна розділити за наступними критеріями:

а) за методом шифрування даних;

б) за методом передачі ключів;

в) за методом зберігання даних.

Криптосистему можна класифікувати за методом шифрування даних в такий спосіб: якщо фрагмент відкритого тексту (окремі символи або послідовність символів) замінюється на деякий його еквівалент в криптотексті, то відповідна криптосистема відноситься до *криптосистем класу заміни*.[25] Якщо букви відкритого тексту при шифруванні лише міняються один з одним місцями, то дана система є прикладом *класу шифрів перестановки*. Якщо ж до початкового тексту застосовується обидва види перетворень, то дана криптосистема відноситься до класу *композиційних шифрів*.

Криптосистеми класифікуються як *симетричні,* якщо при шифруванні, і відправник і одержувач застосовують однаковий криптоключ до вхідних даних. В цьому випадку шифроключ тримається в секреті, так як його наявність дозволяє однозначно декодувати існуючий канал зв’язку. Даний тип шифрування називається синхронним так, як і відправник і одержувач чинять однакові дії для передачі інформації. Прикладами таких криптосистем є блокові шифри, мережі Фейстеля, криптоалгоритм Blowfish, Cobra і багато інших.[25] Їх особливістю є відносна висока швидкість дешифрування, так як на даному етапі немає необхідності у виконанні додаткових дій для перетворення ключа.

У *асиметричних* криптосистемах з асинхронним шифруванням присутні два ключа: відкритий ключ, який передається у відкритому вигляді по каналу зв’язку, і закритий (секретний) ключ, який використовується при шифруванні повідомлень на одній стороні цього каналу зв’язку.[14] Одним з найбільш яскравих прикладів асиметричної криптосистеми є алгоритм RSA. Основною особливістю даної криптосистеми, є досить висока криптостійкість закритого ключа, так як зломщикові знадобиться для початку виявити достатнє велике натуральне просте число, а потім перевірити всі результати в кільці розмірності цього числа. Однак дана криптостійкість вимагає досить великих витрат комп’ютерних ресурсів, таких як процесорний час і пам’ять.

Переваги симетричним криптосистем полягають в наступному:[6]

а) симетричні ключі опрацьовують (шифрують і дешифрують) інформацію достатньо швидко. Деякі апаратні реалізації шифрів шифрують

інформацію зі швидкістю сотні мегабайт у секунду. Програмна реалізація може досягати швидкості декілька мегабайт в секунду;

б) ключі симетричних шрифтів порівняно короткі;

в) різноманітні симетричні шифри можна поєднувати разом і отримувати більш швидкі шифри.

До недоліків симетричним криптосистем відносять наступне:

а) учасники зв’язку повинні таємно обмінюватись ключем по деяким каналам, на які можуть нападати хакери;

б) ключ потрібно часто змінювати, майже для кожного сеансу.

Переваги шифрів з відкритим ключем полягають в наступному:[4]

а) тільки частина ключа повинна бути секретною;

б) секретною частиною ключа учасники зв’язку не обмінюються;

в) ключ може залишатися незмінним в продовж великого проміжку часу, іноді роками;

г) шифри з відкритим ключем мають варіанти електронного підпису.

Серед недоліків шифрів з відкритим ключем відносять наступне:

а) шифри з відкритим ключем працюють значно повільніше симетричних шифрів;

б) розміри ключів шифрів з відкритим ключем набагато довші ключів симетричних шифрів;

в) наразі немає доведень щодо криптографічної стійкості шифрів з відкритим ключем.

У більшості криптосистем, таких як аффинна підстановка Цезаря або криптосистема Хілла, ключ передається в початковому вигляді без будь-яких перетворень.[14] Існують ряд криптосистем, які використовують в реалізації свого алгоритму шифрування значний обсяг даних, але при цьому ці дані можуть бути однозначно отримані за допомогою введення в систему криптоключа, розмір якого може відрізнятися від необхідних алгоритму даних в кілька разів. Такі криптосистеми називаються *розширюваними* криптосистемами, так як вони проходять два етапи шифрування або дешифрування повідомлень:

а) етап розширення ключа (на цьому етапі з введеного ключа за допомогою певної послідовності дій отримують вхідні дані алгоритму);

б) процес шифрування або дешифрування (на цьому кроці отримані на попередньому етапі дані застосовуються алгоритмами перетворення для його кодування або декодування).

Прикладом розширювальної криптосистеми є криптосистема Blowfish, розроблена в 1993 році Брюсом Шнайером.[25] На першому етапі криптосистема перетворює вихідний ключ (довжиною до 448 біт) в вісімнадцять 32-бітових підключів і в чотири 32-бітних S-блоки, що містять 256 елементів. Інформаційна ємність, що необхідна для роботи алгоритму обчислюється як:

$$T = (18 + 4 \cdot 256) \cdot 32 = 33344 \text{ біт} = 4168 \text{ б}a\text{йт}.$$

У результаті криптосистема при малому розмірі вхідного ключа (близько 448 біт), розширює його до необхідних даних розміром, що перевищує розмір вхідного ключа трохи більше ніж в 72 рази.[14]

Шифри *заміни* характеризуються тим, що окремі частини символьної множини A замінюються по деякому правилу окремими частинами символьної множини B. При цьому шифрування є взаємно-однозначним, тобто тільки одному символу (або сукупності символів) з множини A відповідає єдиний символ (або група символів) множини B.

Шифром *перестановки* називається шифр, побудований за наступним правилом: блок розміру n, що складається з символів алфавіту відкритого тексту, міняється місцями з іншим блоком тієї ж розмірності. [25]В результаті вихідне повідомлення являє собою ті ж самі блоки, але розставлені в псевдо-довільному порядку. Ключем в таких криптосистемах є матриця (вектор), задана в певному кільці. При шифруванні даних перетворений блок множиться на ключ. При розшифровці зашифровані блоки множаться на матрицю (вектор) зворотної перестановки, яка є зворотною до введеного в систему ключа.

Як було сказано раніше деякі системи застосовують спеціальні криптоалгоритми, що використовують невизначений набір даних при шифруванні, вони стискають ці дані на основі ключа, тим самим знижуючи обсяг переданої інформації. Однак, такий же механізм можна застосувати і до зашифрованих послань з метою економії мережевого трафіку і швидкості розшифровки послань. Також його застосовують з тією метою щоб ускладнити небажане проникнення до переданої інформації. Для цього застосовуються спеціальні криптоалгоритми, звані блоковими шифрами.[14]

Блоковий шифр – це один з видів симетричних криптосистем, що оперує або групами біт фіксованої довжини, або текстовими повідомленнями також фіксованої довжини. Характерна довжина блоку шифрування лежить в діапазоні 64-256 біт. Якщо довжина вихідного повідомлення, що міститься в блоці менша ніж розмірність блоку, то вона доповнюється будь-якою інформацією до необхідної довжини. Ця інформація називається «забруднювачем» або непотрібною інформацією [6]. Блокові шифри є важливими компонентами багатьох сучасних криптосистем і криптографічних протоколів, які використовуються для шифрування пакетів передачі по мережі. До переваг блокових шифрів відносять схожість алгоритмів шифрування і дешифрування: вони, в більшості випадків, відрізняються тільки порядком дій (при дешифрування послідовність дій прямо протилежна послідовності дій при шифруванні), а також унікальності шифрованих блоків. Це спрощує створення пристроїв шифрування і створення алгоритмів. Крім можливостей шифрувальних машин блокові шифри можуть бути використані для різних інших цілей: генераторів псевдовипадкових чисел, поточних шифрів і хеш-функцій. На даний момент існують 3 найбільш популярних криптоалгоритмів на основі яких будуються інші блокові криптосистеми – це ітеративні блокові шифри, SP мережі, мережі Фейстеля.[25]

У 1971 Хорст Фейстель запатентував два пристрої, які реалізують різні алгоритми шифрування, названі потім загальною назвою «Lucifer». Один з пристроїв використовував конструкцію, згодом названу «мережею Фейстеля». Вона в майбутньому стала основою для багатьох сучасних криптоалгоритмів

таких як Lucifer, Blowfish, KASUMI, RC2, RC5, RC6, MARS, TEA і багато інших.[14]

Мережею Фейстеля називають певну багаторазову структуру, яка називається блоком Фейстеля. [25]При переході від одного блоку до іншго змінюється ключ відповідно до деякого правила. Алгоритми шифрування, які створюються за допомогою ключа, повинні бути досить простими в реалізації і повинні залежати тільки від вхідного ключа. Таким чином, всі етапи шифрування можна представити у вигляді функції такого вигляду:

$$E(b) = (k_1, k_2, \dots, k_n, f(k_i, b)),$$

де b – блок, що підлягає шифруванню, k_i – ключ с порядковим номером i, а f – функція, що застосовується до шифроблоку, використовуючи певний ключ k_i.

Отже, значення, що отримують на виході системи залежать тільки від порядку слідування ключів. При дешифруванні на вхід алгоритму передаються всі ті ж ключі і зашифрований блок, однак при цьому порядок слідування ключів змінюється на протилежний, тим самим гарантуючи однозначність декодування інформації.

Тепер розглянемо докладніше сам алгоритм.[14] Як було сказано раніше на вході до алгоритму міститься блок фіксованої довжини, причому довжина цього блоку повинна бути парним числом. Далі обраний блок ділитися на два рівних по довжині підблока лівий (L_O) і правий (R_O). Після цього лівий блок шифрується за допомогою функції $f\ (L_O, k_O)$, де k_O – це раундовий ключ. До отриманого результату застосовується будь-яка іволютивна операція, наприклад, бітове складання по модулю 2 XOR. Отриманий результат ставиться на місце того підблоку, який не був задіяний в шифруванні, а значення невикористаного підблока ставитися на місце шифруїмого. Таким чином, обмін даними відбувається хрест-навхрест. Після чого операція повторюється $N - 1$, де N – кількість раундів мережі Фейстеля.[25] При цьому між переходами від одного раунду до іншого ключі можуть змінюватися по деякому правилу

(зазвичай вони йдуть в прямому порядку). Приклад роботи алгоритму представлений на рисунку 1.1.

У сучасному світі нас оточують певні набори даних, які або несуть дуже важливу для нас інформацію, або часто використовуються в різних інформаційних структурах, такі як бази даних, автоматизовані інформаційні системи, операційні системи і т.д. Одними з найбільш яскравих прикладів таких наборів є: PIN коди, номери банківських карт, номери мобільних телефонів, паролі електронної пошти та облікових записів і так далі.

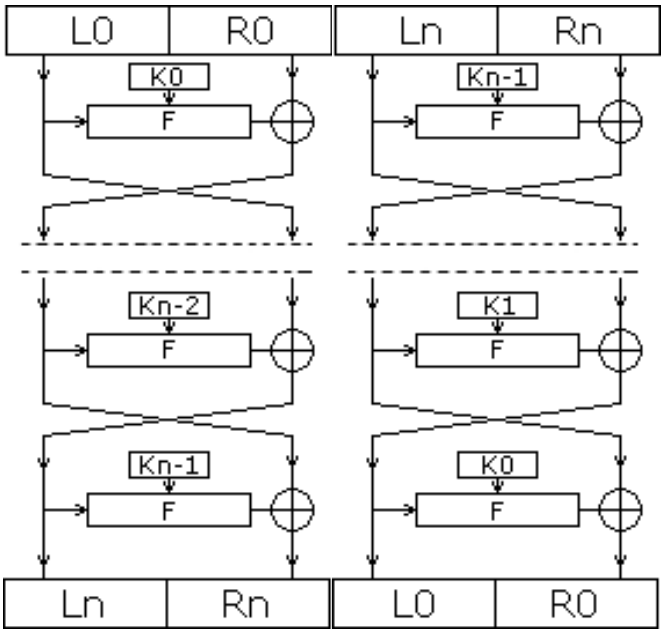

Рисунок 1.1 – Шифрування (зліва) і дешифрування (справа) в мережах Фейстеля

Зазвичай людині для складання таких комбінацій не потрібно великих зусиль: досить лише не повторювати числа, на певних позиціях. Але іноді компаніям потрібні створювати безліч таких комбінацій, наприклад, такими компаніями є банки (в разі кредитних карт) і телекомунікаційні компанії (в разі генерації телефонних номерів). Так як людині дуже складно обробити і сформувати такі обсяги, то компанії змушені вдаватися за допомогою до комп’ютерних технологій і обчислювальних структур. Але тут виникає наступна проблема: комп’ютери засновані на строгих алгоритмах, результати яких за вибором нового елемента генерують послідовності, які можуть бути

спрогнозовані, тому що використовуються псевдовипадкові перетворення. Ці дії ми можемо проводити не тільки з числами, а й в будь-якому алфавіті будь-якої розмірності, наприклад, над алфавітом українських букв, якщо задати однозначне відповідно кожному символу певного числа. Таким чином, формування комбінацій випадкових ланцюжків символів зводитися до шифрування повідомлень в довільному алфавіті.

1.3 Криптосистема RSA

Криптосистема RSA – це одна з широко використовуваних криптосистем з відкритим ключем. Її криптографічна стійкість заснована на складній практичній реалізації проблеми факторизації великих цілих чисел.

Криптосистема RSA забезпечує такі механізми захисту як шифрування і цифровий підпис (аутентифікація – встановлення авторства). Криптосистема була розроблена в 1977 році і названа на честь її розробників Ronald Rivest, Adi Shamir і Leonard Adleman. [2]

Алгоритм RSA працює наступним чином: нехай деякий адресат V відправляє інформацію адресату W. Адресат V подає вихідний текст t у вигляді натурального числа m, шифрує повідомлення m, отримує шифротекст c, відправляє адресату W. Адресат W отримує шифротекст c, дешифрує c, отримує число m та вихідний текст t.

Створення ключів.[1] Кожний адресат обчислює свій відкритий ключ і відповідний йому секретний ключ. Адресат повинен виконати наступні дії:

а) обрати два відмінних простих числа p і q приблизно одного розміру;

б) обчислити $n = pq$ і функцію Ейлера $\varphi(n) = (p-1)(q-1)$;

в) обрати випадкове число $e, 1 < e < \varphi$, таке, що НОД $(e, \varphi) = 1$;

г) за допомогою розширеного алгоритму Евкліда знайти такі цілі a, x, що $ca + \varphi x = 1$. Тоді $ea \equiv 1(mod\ \varphi)$. Нехай довільне $k \in \mathbb{Z}$. Знайшовши суму $ea \equiv 1(mod\ \varphi)$ і $ek\varphi \equiv 0(mod\ \varphi)$, отримаємо $e(a + k\varphi) \equiv 1(mod\ \varphi)$. Якщо

$a \notin (1, \varphi)$, то знайти таке ціле k, що, $(a + k\varphi) \in (1, \varphi)$, і в якості a узяти $a + k\varphi$;

д) відкритий ключ адресата W– це (n, e). Секретний ключ для – це a.

Шифрування. Адресат V пише текст адресату W. Адресат V повинен виконати наступні дії:

а) отримати відкритий ключ (n, e) адресата W;

б) за допомогою будь-якого методу M, адресат V подає свій текст t як повідомлення у вигляді натурального числа m із сегменту $[0, n - 1]$;

в) обчислити шифротекст $c = m^e (mod\ n)$;

г) відправити шифротекст c адресату W.

Дешифрування. Щоб вилучити текст t із шифротексту c, адресат W повинен зробити наступне:

а) взяти свій секретний ключ a і обчислити повідомлення $m = c^a (mod\ n)$;

б) обчислити текст t адресата V за допомогою методу M.

Зауваження. На практиці для криптографічної стійкості модуль n задається двійковим числом з 1024 і більше двійковими розрядами.[25]

1.4 Криптосистема Ель-Гамаля

Схема Ель-Гамаля (Elgamal) – криптосистема з відкритим ключем, що базується на складності обчислення дискретних логарифмів в кінцевому полі. Криптосистема включає в себе алгоритми шифрування та алгоритми цифрового підпису.[2] Схема Ель-Гамаля лежить в основі стандартів електронного цифрового підпису в США та багатьох країн Європи.

Криптографічна стійкість криптосистеми Ель-Гамаля забезпечується складністю практичні реалізації проблеми знаходження дискретного логарифма в мультиплікативній групі при великих простих числах p.

Схема була запропонована Тахером Ель-Гамалем в 1984 році. Ель-Гамель розробив один з варіантів алгоритму Діффі-Хеллмана. Він удосконалив систему

Діффі-Хелмана і отримав два алгоритми, які використовувалися для шифрування і для забезпечення аутентифікації.

Алгоритм **обчислення ключів** полягає в наступному[14]:

а) потрібно обрати випадкове просте число p;

б) обрати довільне ціле число g, що є первісним коренем по модулю p;

в) обрати випадкове ціле число $x, 1 < x < p$;

г) обчислити $y = g^x \ (mod\ p)$;

д) відкритим ключем є трійка (p, g, y), а секретним ключем – x.

Шифрування проходить наступним чином:

а) обрати сесійний ключ – випадкове ціле число k таке, що. $1 < k < p - 1$;

б) обчислюються числа $a = g^k \ (mod\ p)$ і $b = y^k M \ (mod\ p)$;

в) пара чисел (a, b) є шифротекстом.

1.5 Криптосистема Рабіна

Бажана властивість схеми шифрування – це наявність доведення, що атака схеми така ж важка, як і складність розв’язання факторизації цілих чисел чи проблема обчислення дискретного логарифма. Схема Рабіна була першою схемою з доведеною складністю атаки, яка дорівнює складності факторизації.[14]

Обчислення ключів полягає в наступному. [2]

а) обрати два відмінних випадкових простих числа p і q приблизно одного розміру;

б) обчислити $n = pq$;

в) відкритий ключ – це n. Секретний ключ – це (p, q).

Шифрування виконується за наступним алгоритмом:

а) узяти відкритий ключ n;

б) за допомогою метода M, подати свій текст t у вигляді натурального числа m із сегмента $[0, n - 1]$;

в) обчислити $c = m^2 (mod\ n)$;

г) відправити свій шифротекст c.

Дешифрування відбувається наступним чином:

а) розв'язати рівняння $c = m^2 (mod\ n)$ і знайти чотири квадратних кореня m_1, m_2, m_3, m_4 по модулю n (число c має один або два кореня із c по модулю n, якщо НОД$(m, n) \neq 1$, що можливо з дуже маленькою ймовірністю);

б) відправлений текст m – це одне з чисел m_1, m_2, m_3, m_4.

1.6 Хеш-функція

Зміст будь-якого файлу або його частини – це деякий текст t, який складається зі символів клавіатури комп'ютера і подається в комп'ютері як послідовність нулів і одиниць, тобто як *бінарний текст* m (бінарне слово в алфавіті $\{0,1\}$).[1] Всякий бінарний текст можна розглядати як запис m_2 в двійковій системі числення натурального числа m, яке, в свою чергу, можна подати у вигляді запису m_h в системі числення за будь-якою іншою основою h.

Між множинами всіх комп'ютерних текстів, між множинами їх бінарних слів в алфавіті $\{0,1\}$ у комп'ютері існує взаємна-однозначна відповідність. Будь-яке представлення інформації в комп'ютері будемо називати текстом. Позначення тексту, який складається із символів клавіатури комп'ютера – це t. Бінарне подання тексту t позначається m_2, а натуральне h–ричне число позначається m_h.[25]

Хеш-функція $w = h(x)$ – це швидко обчислювальна (за часом) функція, яка відображає усілякий вихідний бінарний текст x довільної довжини в бінарне число $w = h(x)$, що має назву *хеш-значення* w (чи просто хеш), який виступає в ролі компактного представника (паспорта) вхідного слова x. Хеш-функція $w = h(x)$ має наступні властивості:[2]

а) її значення w «легко» обчислюється для будь-якого тексту x, але практично неможливо визначити вихідний текст $x = h^{-1}(w)$ по його хеш-значенню w;

б) практично неможливо знайти два різноманітних тексти x і y, для яких $h(x) = h(y)$.

За допомогою хеш-функції тексту x ставиться у відповідність унікальне число $w = h(x)$. Значення хеш-функції – це велике число, яке виходить за границі величин цілих чисел, що допустимі в більшості алгоритмічних мовах програмування. Mathcad, наприклад, допускає цілі (10-ричні) числа довжини не більше 15 цифр. Для роботи з більшістю цілими числами з довжиною десятиричного запису в 100 і більше цифр, доводиться писати спеціальний програмний процесор.[4] Тому у роботі в практичній реалізації значення хеш-функції буде задаватися штучно, для прикладу, невеликим числом.

1.7 Висновки за розділом 1

У розділі 1 досліджено основні поняття кодування та криптографії, пояснено термін модулярної алгебри, хеш-функції. Розглянуто класифікацію криптосистем. Криптосистеми Рабіна, Ель-Гамаля та RSA розглянуто детальніше, так як вони будуть необхідними для подальшого дослідження алгоритмів криптографії та кодування у другому розділі.

2 АЛГОРИТМИ КРИПТОГРАФІЇ ТА КОДУВАННЯ

Одним із важливих інструментів криптографії є алгоритми електронно-цифрових підписів. Вони забезпечують цілісність електронного документа, аутентифікацію автора документу. Наразі електронний цифровий підпис застосовується для забезпечення безпечних банківських транзикацій, в державних закупівлях, електронному документообіг, податковій звітності, в мережевих протоколах та іншому. Серед алгоритмів кодування розглянемо циклічне кодування. Циклічне кодування використовують в сучасній технології передачі данних, застосовують для оцінки достовірності інформації. [18]

2.1 Поняття електронного цифрового підпису

Електронний цифровий підпис (ЕЦП) – це реквізит електронного документу, який був отриманий в результаті криптографічного перетворення інформації з використанням закритого ключа підпису. [8] Він дозволяє встановити відсутність спотворення інформації в електронному документі з моменту формування підпису і перевірити належність підпису власнику сертифіката ключа підпису. Електронний підпис призначений для ідентифікації людини, яка підписала електронний документ і є повноцінною заміною або аналогом власноручного підпису в випадках, які передбаченні законодавством України. [11]

Використання електронного підпису дозволяє здійснити наступне:

а) контроль цілісності документа, який був підписаний ЕЦП: при будь-якій випадковій і невипадковій зміні документа – підпис стане недійсним, так як його створюють на основі вихідного стану документа;

б) захист від зміни або підробки документу: гарантія виявлення підробки при контролі цілісності – робить підроблення недоцільним в більшості випадків;

в) неможливість відмови від авторства: так як створити коректний підпис можливо тільки знаючи закритий ключ, а він, в свою чергу, відомий тільки власнику, то він(власник) не може відмовити від свого підпису під документом.[11]

У законодавстві України вирізняють наступні види електронних підписів:

а) простий електронний підпис. Він створюється за допомогою паролів, кодів та інших інструментів. Такі засоби дозволяють ідентифікувати автора підписаного документа. Недоліком такого підпису є відсутність можливості перевірити документ на предмет наявності змін з моменту підписання. Прикладом простого електронного підпису є комбінація пароля і логіна. Прості підписи можуть використовуватись громадянами для відправки повідомлень органам влади;

б) посилений некваліфікований електронний підпис. Такий підпис створюється за допомогою криптографічних засобів. Він дозволяє визначити не тільки автора, а також перевірити документ на наявність змін. Посилені некваліфіковані і прості підписи можуть замінити підписаний паперовий документ у випадках, які передбаченні законом або за згодою сторін. Посилений підпис може розглядатись як аналог документа з печаткою;

в) посилений кваліфікований електронний підпис. Такий підпис замінює паперові документи у всіх випадках, окрім тих, коли закон вимагає наявність виключно документа на папері. Посилений підпис повинен обов’язково мати сертифікат акредитованого засвідчуючого центру. За допомогою таких підписів ви можете організувати юридично значущий електронний документообіг з партнерськими компаніями, органами державної влади та ін.[15]

Існує декілька схем побудови цифрового підпису: на основі алгоритмів асиметричного шифрування, на основі алгоритмів симетричного шифрування. Найбільш поширеними є схеми зі застосуванням алгоритмів асиметричного шифрування. Відмінність цих схем в тому, що коли цифровий підпис базується на схемі симетричного шифрування, то у системі має бути третя особа, що користується довірою обох сторін. Авторизація документа полягає в

шифруванні його секретним ключем і передачі його третій особі. Створення цифрового підпису на основі алгоритмів асиметричного шифрування полягає в наступному: підписання проводиться зі застосуванням закритого ключа, а перевірка підпису – зі застосуванням відкритого ключа. Забезпечення складності обчислення легітимного підпису без знання закритого ключа спирається на обчислювання: завдання дискретного логарифмування та завдання факторизації.[6]

2.2 Алгоритм електронного цифрового підпису RSA

Основою криптографічної стійкості цифрового підпису RSA є складність проблеми факторізації цілих чисел. Так як шифрування – це взаємно однозначне відображення, то цифровий підпис може бути отриманий за допомогою шифрування та дешифрування.[6]

Розглянемо електронний цифровий підпис RSA з використання хеш-функції $\rho = h(x)$, яка зіставляє будь-якому тексту x унікальне натуральне число $\rho = h(x)$.[11] Нехай деякий учасник V хоче підписати свій документ, тоді учасник W може перевірити підпис учасник а V під його документом t. Розглянемо алгоритми створення ключів, створення підпису та перевірки підпису.[1]

Створення ключів. На цьому етапі кожний учасник обчислює свій відкритий ключ і відповідний йому секретний ключ. Учасник повинен виконати наступні дії:

а) обрати два різних простих числа n і m приблизно однакового розміру;

б) обчислити $p = nm$ і функцію Ейлера $\varphi(p) = (n-1)(m-1)$;

в) обрати випадкове число $k, 1 < k < \varphi$, таке, що НОД(k, φ)=1;

г) за допомогою розширеного алгоритму Евкліда знайти ціле число q, $1 < q < \varphi$, для якого $kq = 1\ (mod\ \varphi)$;

д) відкритий ключ учасника $V-$ це (p,k). Секретний ключ для учасника V – це q.

Створення підпису. Учасник V підписує документ t, користуючись своїм секретним ключем. Учасник V повинен зробити наступні дії:[2]

а) подати текст документа t у вигляді натурального числа r за допомогою будь-якого методу R;

б) обчислити значення хеш-функції $h = h(t)$;

в) обчислити $s = h^q(mod\ p)$;

г) відправити текст документа t з підписом s учаснику W.

Перевірка підпису. Щоб здійснити перевірку підпису s учасника V під його текстом t, учасник W повинен виконати наступні дії:[4]

а) отримати відкритий ключ (p,k) учасника V;

б) за допомогою метода R подати текст t у вигляді цілого числа r з відрізка $[0,p-1]$;

в) обчислити $h = h(t) = h(r)$;

г) обчислити $h_1 = s^k(mod\ p)$;

д) якщо $h = h_1$, то прийняти підпис. Якщо $h \neq h_1$, то відмовити у підписанні.

2.3 Алгоритм електронного цифрового підпису Ель-Гамаля

При використанні схеми цифрового підпису Ель-Гамаля на текст t потрібно обчислити значення хеш-функції $h(t)$, яке потім будемо використовувати при обчисленні і перевірці цифрового підпису під текстом документа.[1]

Створення ключів. Кожний учасник створює свій відкритий ключ і йому відповідає секретний ключ. Учасник має виконати наступні дії: [6]

а) обрати випадкове просте число p і задати генератор α для мультиплікативної групи $\mathbb{Z}_p^*$;

б) обрати довільне число $q, 1 \leq q \leq p-2$;

в) обчислити $x = \alpha^q \ (mod\ p)$;

г) відкритий ключ для учасника V – це трійка чисел (p, α, x). Секретний ключ учасника – число q.

Створення підпису. Учасник V підписує свій текст t (довільного розміру). Учасник W може перевірити підпис учасника V під текстом t його документа. Учасник V повинен виконати наступні дії: [11]

а) обчислити значення хеш-функції $h(t)$;

б) обрати випадкове секретне ціле число k з $[1, p-2]$ – таке, що НОД $(k, p-1) = 1$ (сеансовий ключ, тільки на один сеанс);

в) обчислити $k^{-1}\big(mod(p-1)\big)$;

г) обчислити $r = \alpha^k (mod\ p)$;

д) обчислити $s = k^{-1}(h(t) - q \cdot r)\ (mod(p-1))$. Якщо різниця $h(t) - q \cdot r < 0$, то до цієї різниці потрібно додати модуль $p-1$;

е) підпис учасника V під його текстом t – є пара чисел (r, s).

Перевірка підпису. Щоб перевірити підпис (r, s) учасника V під його текстом t, учасник W має виконати наступні дії:[9]

а) отримати відкритий ключ (p, α, x) учасника V;

б) обчислити значення хеш-функції $h(t)$;

в) перевірити, що $r \in [1, p-1]$; якщо ні, то відмовити у підписанні;

г) обчислити $w_1 = x^r \cdot r^s (mod\ p)$;

д) обчислити $w_2 = \alpha^{k(m)} (mod\ p)$;

е) прийняти підпис, якщо $w_1 = w_2$, і відмовити в іншому випадку.

Зауваження. Для криптографічної стійкості на сучасному етапі обирають p довжиною 1024 і більше біт.[1]

2.4 Алгоритм електронного цифрового підпису DSA

Схема цифрового підпису DSA (Digital Signature Algorithm) – це певний варіант цифрового підпису Ель-Гамаля. Схема цифрового підпису DSA базується на складності обчислення дискретного логарифма в $\mathbb{Z}_p^*$. [1] Наразі не доведено, що підпис DSA криптографічно стійкий. Схема цифрового підпису DSA потребує використання хеш-функції.

Створення ключів. Кожний учасник створює свій відкритий ключ і відповідний йому секретний ключ. Учасник має виконати наступні дії:[9]

а) обрати випадкове число $q, 2^{159} < q < 2^{160}$;

б) обрати число $t, 0 \leq t \leq 8$, і випадкове просте число p, $2^{511+64t} < p < 2^{(512+64t)}$, таке, що q ділить $p - 1$;

в) обрати генератор $\alpha \in \mathbb{Z}_p^*$ для циклічної підгрупи порядку q у групі $\mathbb{Z}_p^*$. Для цього учасник V повинен зробити г) і д);

г) обрати випадковий елемент $g \in \mathbb{Z}_p^*$ і знайти $\alpha \equiv g^{\frac{p-1}{q}} \ (mod\ p)$;

д) якщо $\alpha = 1$, то перейти до кроку г) з іншим g;

е) обрати довільне число $a, 1 \leq a \leq q - 1$;

ж) обчислити $y = \alpha^a (mod\ p)$;

з) відкритий ключ учасника V – це (p, q, α, y). Секретний ключ учасника V – це число a.

Створення підпису. Учасник V підписує свій текст t (довільного розміру). Учасник W може перевірити підпис учасника V за допомогою відкритого ключа учасника V. Учасник V повинен виконати наступні дії:[1]

а) обрати довільне таємне число $k, 0 < k < q$ (сеансовий ключ, тільки для одного сеансу);

б) обчислити $r = \left(a^k \ (mod\ p)\right) (mod\ q)$;

в) обчислити $k^{-1} \ (mod\ p)$;

г) обчислити $s = k^{-1} \cdot (h(m) + a \cdot r) (mod\ q)$, де $h(m)$: $(0,1)^* \to \mathbb{Z}_p$ – це деяка хеш-функція;

д) підпис учасника V – це пара чисел (r, s).

Перевірка підпису. Щоб перевірити підпис (r, s) учасника V під його текстом t , учасник W повинен виконати наступні дії:[9]

а) взяти відкритий ключ (p, q, α, y) учасника V;

б) перевірити, що $0 < r < q$ і $0 < s < q$. Якщо ні, то відмовити у підписанні;

в) обчислити $w = s^{-1}\ (mod\ q)$і $h(m)$;

г) обчислити $h(t)$;

д) обчислити $u_1 = w \cdot h(t)(mod\ q)$ і $u_2 = r \cdot w(mod\ q)$;

е) обчислити $v = \big(\ \alpha^{u_1} \cdot y^{u_2}(mod\ p)\big)(mod\ q)$;

ж) прийняти підпис, якщо $v = r$, і відмовити в іншому випадку.

Зауваження. Для криптографічної стійкості рекомендується обирати q довжиною 160 біт, розмір p має бути кратним 64 біт й лежати між 512 і 1024 біт включно. [11]

2.5 Алгоритм електронного цифрового підпису Рабіна

Цифровий підпис за схемою Рабіна схожий до цифрового підпису за схемою RSA. Простори підписів M_S є множиною квадратичних лишків по модулю n, а підписи – це квадратні корені з них. Публікується взаємно-однозначна функція R, яка діє з простору тексту M в простір підписів M_S. [1]

Створення ключів. Кожний учасник обчислює свій відкритий ключ і відповідний йому секретний ключ. Учасник має виконати наступні дії: [1]

а) обрати два відмінних випадкових простих числа p і q приблизно одного розміру;

б) обчислити $n = pq$;

в) функція R знаходиться наступним чином. Якщо число m – це текст, що має підписатися, то $R(m)$ – це найближче до m число, для якого існують чотири відмінних кореня $\sqrt{R(m)}\ (mod\ n)$ (таке число існує). При цьому

$R(m) > m$, якщо чотири кореня $\sqrt{R(m)}\,(mod\ n)$ для $m < n$ існують, і $R(m) < m$ в іншому випадку;

г) відкритий ключ учасника – це n . Секретний ключ – це (p, q).

Створення підпису. Користуючись своїм секретним ключем (p, q), учасник V підписує свій текст t. Учасник V має діяти за наступною схемою: [9]

а) подати свій текст t у вигляді натурального числа за допомогою якого-небудь методу M;

б) обчислити $w = R(m)$ і $i = R(m) - m$;

в) обчислити $s = \sqrt{w}\ (mod\ n)$;

г) підпис учасника V (включно з текстом) це число s.

Перевірка підпису. Користуючись відкритим ключем n для V, учасник W може перевірити підпис s учасника V і вилучити з підпису s текст t від V. Учасник W має зробити наступні дії: [1]

а) отримати відкритий ключ n від учасника V;

б) обчислити $w = s^2(mod\ n)$;

в) перевірити , що $w \in M_R$. Якщо ні, то відмовити у підписанні V;

г) отримати $m = R^{-1}(w)$ і знайти текст t.

Приклад 2.1. [9] Учасник V підписує свій текст $t = RAD$, а учасник W має перевірити підпис.

Створення ключів. Учасник V створює свій відкритий ключ і відповідний йому секретний ключ. Учасник має виконати наступні дії:

а) обрати два відмінних випадкових простих числа $p = 1699$ і $q = 1597$ приблизно одного розміру;

б) обчислити $n = pq = 1699 \cdot 1597 = 2713303$;

в) відкритий ключ учасника – це $n = 21713303$. Секретний ключ – це $(p = 1699, q = 1597)$.

Створення підпису. Користуючись своїм секретним ключем $(p = 1699, q = 1597)$, учасник V підписує свій текст $t = RAD$. Учасник V має діяти за наступною схемою:

а) подати свій текст $t = RAD$ у вигляді натурального числа за допомогою якого-небудь методу M, наприклад у 27-ричній системі числення наступним чином: $m1 = 18 \cdot 27^2 + 1 \cdot 27 + 4 = 13153_{10}$;

б) взяти праве слово, яке складається з двох букв в $m1$, приписати його до $m1$ справа і отримати $m = 1315353$. Можна взяти будь-яку іншу взаємно-однозначну функцію для перетворення числа $m1$;

в) обчислити $w = R(m)$ як найближче число до $m = 1315353$, для якого існують чотири відмінних кореня $\sqrt{R(m)}\,(mod\ n)$ і отримати $w = R(m) = R(1315353) = 1315358$ $i = R(m) - m$=1315358-1315353=5;

г) обчислити $s1 = \sqrt{w}\ (mod\ n) = \sqrt{1315358}(mod\ 2713303) =$ $= (2264649, -2264649, 399147, -399147)$;

д) до кожної від'ємного компонента в $s1$ додати модуль n і отримати $s = (2264649, 4486546, 399147, 2314156)$;

е) підпис учасника V (включно з текстом) це будь-яке число s і i, наприклад $(s = 2314156, i = 5)$.

Перевірка підпису. Щоб перевірити підпис $(s = 2314156, i = 5)$ учасника V і вилучити з підпису s текст t від V, учасник W має зробити наступні дії:

а) отримати відкритий ключ $n = 2713303$ від учасника V;

б) обчислити $w = s^2(mod\ n) = 2314156^2(mod\ 2713303) = 1315358$, $m = w - i = 1315358 - 5 = 13153535$;

в) так як m має справа подвоєне двобуквене слово 53, то підпис приймається;

г) отримати $m1 = 13153_{10} = (18\ 1\ 4)_{27}$ і тому текст $t = RAD$.

2.6 Алгоритм електронного цифрового підпису GQ

Схема Гілу-Куіскуатера (Guillou–Quisquater, GQ) цифрового підпису потребує хеш-функції $h: \{0,1\}^* \to \mathbb{Z}_n$, де n – це деяке додатне ціле число. Схема заснована на складності розв'язання проблеми факторизації цілих чисел. [11]

Створення ключів. Кожний учасник створює свій відкритий ключ і відповідний йому секретний. Учасник має зробити наступні кроки. [9]

а) обрати випадкові відмінні секретні прості числа p, q приблизно одного розміру і обчислити $n = pq$;

б) обрати ціле $e \in [1, n-1]$, для якого НОД $\big(e, (p-1)\cdot(q-1)\big) = 1$;

в) обрати ціле $J_V, 1 \le J_V \le n$, для якого НОД$(J_V, n) = 1$ (Бінарне подання для можна використовувати для передачі інформації про Учасника, такої як ім'я, адреса, номер водійських прав і т.д.);

г) знайти таке ціле число $a \in \mathbb{Z}_n$, що $J_V \cdot a^e \equiv 1\ (mod\ n)$ і виконати це наступним чином:

1) обчислити $J_V^{-1} (mod\ n)$;

2) обчислити $d_1 = e^{-1} (mod\ (p-1))$ і $d_2 = e^{-1} \big(mod\ (q-1)\big)$;

3) обчислити $a_1 = (J_V^{-1})^{d_1} (mod\ p)$ і $a_2 = (J_V^{-1})^{d_2} (mod\ q)$;

4) знайти таке a, що одночасно $a \equiv a_1 (mod\ p)$ і $a \equiv a_2 (mod\ q)$;

д) відкритий ключ учасника – це набір (n, e, J_V). Секретний ключ учасника – це число a.

Створення підпису. Користуючись своїм секретним ключем, учасник одночасно V підписує бінарний текст m будь-якої довжини. Учасник V має зробити наступні дії: [9]

а) обрати випадкове ціле число k і обчислити $r = k^e (mod\ n)$;

б) обчислити $l = h(m||r)$, $m||r$ – означає конкатенацію m і r, тобто запис r до m справа;

в) обчислити $s = ka^l (mod\ n)$;

г) підпис учасника V під текстом m – це пара (s, l).

Перевірка підпису. Користуючись відкритим ключем учасника V, учасник W може перевірити підпис (s, l). Учасник W повинен виконати наступні дії:

а) отримати відкритий ключ (n, e, J_V) учасника V;

б) обчислити $u = s^e \cdot (J_V)^l \ (mod\ n)$ і $l' = h(m||u)$;

в) прийняти підпис учасника V, якщо $l = l'$. Відмовити в іншому випадку.

Зауваження. Для криптографічної стійкості потрібно обирати модуль n довжини ≥ 768 біт, число e довжини ≥ 128 біт, значення хеш-функції від 128 до 160 біт. Тоді відкритий ключ буде довжини $896 + u$ біт, де u – число біт для представлення J_V. Секретний ключ a був би 768 біт. [11]

Приклад 2.2. [1] Учасник V підписує свій текст $m = 1101110001$, а учасник W має перевірити підпис.

Створення ключів. Учасник V створює свій відкритий ключ і відповідний йому секретний. Він робить наступні кроки:

а) обирає випадкові відмінні секретні прості числа $p = 20849, q = 27457$ і обчислює $n = pq = 572450993$;

б) обирає ціле $e = 47 \in [1,572450992]$, для якого НОД$(47,20848 \cdot \cdot 27456) = 1$;

в) обирає ціле $J_V = 1091522, 1 \leq J_V \leq 572450993$, для якого НОД$(1091522,572450993) = 1$;

г) розв'язуючи $J_V \cdot a^e \equiv 1\ (mod\ n)$, отримуємо $a = 214611724$;

д) відкритий ключ учасника – це набір $(n = 572450993, e = 47, J_V = 1091522)$. Секретний ключ учасника – це число $a = 214611724$.

Створення підпису. Щоб підписати бінарне повідомлення $m = 1101110001$ учасник V робить наступні дії.

а) обирає випадкове ціле число $k = 42134$ і обчислює $r = k^e (mod\ n) = 297543350$;

б) обчислює $l = h(m||r) = 2713833$;

в) обчислює $s = ka^l (mod\ n) = 42134 \cdot (214611724^{2713833})$ $(mod\ 572450993) = 252000854$;

г) підпис учасника V під текстом m – це пара $(s = 252000854,$ $l = 2713833)$.

Перевірка підпису. Користуючись відкритим ключем учасника V, учасник W може перевірити підпис $(s = 252000854, l = 2713833)$. Учасник W повинен виконати наступні дії.

а) отримати відкритий ключ $(n = 572450993, e = 47, J_V = 1091522)$ учасника V.

б) обчислити $s^e (mod\ n) = 252000854^{47} (mod\ n) = 39861962,$ $(J_V)^l\ (mod\ n) = 1091522^{2713833} (mod\ n) = 110523867,$ $u = s^e \cdot$ $(J_V)^l\ (mod\ n) = 297543350$;

в) так як, $u = r, l' = h(m||u) = h(m||r) = l$, то $l = l'$. Отже учасник W приймає підпис учасника V.

2.7 Алгоритм електронного цифрового підпису Шнорра

Схема Шнорра – це деяка модифікація схеми DSA без обмежень DSA на прості p і q при обчисленні ключів. В схемі Шнорра, як і в схемі DSA, використовується підгрупа порядку q в групі $\mathbb{Z}_p^*$, де p – це деяке велике просте число. Схема також потребує хеш-функції $h: \{0,1\}^* \to \mathrm{F}_q$. Схема заснована на складності розв'язання проблеми дискретного логарифму. [1]

Створення ключів. Кожний учасник створює свій відкритий ключ і відповідний йому секретний ключ. Учасник повинен виконати наступні дії: [9]

а) обрати приблизно однакового розміру прості числа p і q, для яких q ділить $p - 1$;

б) обрати генератор $\alpha \in \mathbb{Z}_p^*$ для циклічної підгрупи поряду q в групі $\mathbb{Z}_p^*$. Для цього Учасник повен виконати наступні дії;

1) обрати елемент $g \in \mathbb{Z}_p^*$, та знайти $\alpha = g^{\frac{p-1}{q}} (mod\ p)$;

2) якщо $\alpha = 1$, то перейти до кроку 1) з іншим g;

в) обрати довільне число $a, 1 \leq a \leq q - 1$;

г) обчислити $y = \alpha^a (mod\ p)$;

д) відкритий ключ учасника – це (p, q, α, y). Секретний ключ – це число a.

Створення підпису. Користуючись своїм секретним ключем, учасник V підписую бінарний текст m довільної довжини. Учасник V повинен виконати наступні дії:

а) обрати випадкове секретне число $k, 1 \leq k \leq q - 1$;

б) обчислити $r = \alpha^k (mod\ p), e = h(m||r), s = ae + k (mod\ q)$;

в) підпис учасника V під текстом m – це пара (s, e);

Перевірка підпису. Користуючись відкритим ключем учасника V, учасник W може перевірити підпис (s, e) учасника V. Учасник W повинен буде виконати наступні дії:

а) отримати відкритий ключ (p, q, α, y) учасника V;

б) обчислити $v = \alpha^s \cdot y^{-e} (mod\ p), e' = h(m||v)$;

в) прийняти підпис якщо $e' = e$, і відмовитись у підписанні в іншому випадку.

Зауваження. Для криптографічної стійкості рекомендується обирати q довжиною 160 біт, розмір p лежить між 512 і 1024 і більше біт. [1]

2.8 Алгоритм електронного цифрового підпису Ніберга-Рюпеля

Схема Ніберга-Рюппеля – це деяка модифікація схеми DSA без обмежень DSA на прості p і q при обчисленні ключів. Схема базується на складності розв'язання проблеми дискретного логарифма. [1]

Обчислення ключів. Кожний учасник створює свій відкритий ключ і відповідний йому секретний ключ. Кожний учасник повинен зробити наступні дії: [9]

а) обрати прості числа p і q, для яких q ділить $p - 1$;

б) обрати генератор $\alpha \in \mathbb{Z}_p^*$ для циклічної підгрупи поряду q в групі $\mathbb{Z}_p^*$. Для цього учасник повен виконати наступні дії:

1) обрати елемент $g \in \mathbb{Z}_p^*$, та знайти $\alpha = g^{\frac{p-1}{q}} (mod\ p)$;

2) якщо $\alpha = 1$, то перейти до кроку 1) з іншим g;

в) обрати довільне число $a, 1 \leq a \leq q-1$;

г) обчислити $y = \alpha^a (mod\ p)$;

д) відкритий ключ учасника – це (p, q, α, y). Секретний ключ – це число a.

Створення підпису. Користуючись своїм секретним ключем, учасник V підписую бінарний текст m довільної довжини. Учасник V повинен виконати наступні дії: [9]

а) обчислити $m' = R(m)$;

б) обрати випадкове таємне ціле $k, 1 \leq k \leq q-1$ і обчислити $r = \alpha^{-k}\ (mod\ p)$;

в) обчислити $e = m'r (mod\ p)$;

г) обчислити $s = ae + k\ (mod\ q)$;

д) підпис V під m – це пара (e, s).

Перевірка підпису. Користуючись відкритим ключем учасника V, учасник W може перевірити підпис (s, e) учасника V і вилучити з підпису повідомлення від V. Учасник W повинен буде виконати наступні дії:

а) отримати відкритий ключ (p, q, α, y) учасника V;

б) перевірити, що $e \in [1, p-1]$. Якщо ні, то відмовити у підписанні;

в) перевірити, що $s \in [1, q-1]$. Якщо ні , то відмовити у підписанні;

г) обчислити $v = \alpha^s \cdot y^{-e}\ (mod\ p)$ і $m' = ve\ (mod\ p)$;

д) перевірити, що $m' \in M_R$. Якщо ні, то відмовити у підписанні;

е) обчислити $m = R^{-1}(m')$;

Зауваження. Для криптографічної стійкості рекомендується брати q довжиною 160 біт, розмір p лежить між 512 і 1024 і більше біт. [1]

2.9 Циклічній код. Алгоритм циклічного кодування

Циклічні коди – це такий клас кодів, які виправляють та контролюють помилки. Кодування і декодування таких кодів базується на поліноміальному представленні. [12]

Лінійний (n, k)-код C' над полем F_q називається *циклічним*, якщо з того, що вектор $(c_0, c_1, \dots, c_{n-1})$ належить C', слідує, що його циклічний здвиг $(c_{n-1}, c_0, \dots, c_{n-2})$ також належить C'.[3]

Лінійний код над полем $GF(q)$ довжиною n представляє собою підпростір простору $GF(q^n)$. Кожний вектор із цього простору можна подати у вигляді многочлена від x степеня не вище $n-1$. Компоненти вектору ототожнюються з коефіцієнтами многочлена. [18] Співставимо кодовому слову c' поліном $c(x)$:

$$c' = (c_0, c_1, \dots, c_{n-1}) \rightarrow c(x) = c_0 + c_1 x + \dots + c_{n-1} x^{n-1}. \quad (2.1)$$

Змінна x є індикатором відносно положення елемента c_j у вигляді одночлена $c_j x^j$ полінома $c(x)$. В поліноміальному представленні циклічний здвиг на одну позицію $D_c\{c(x)\}$ відповідає множенню на x за модулем$(x^n - 1)$:

$$D_c\{c(x)\} = xc(x)\ mod\ (x^n - 1),$$

де D_c – оператор циклічного здвигу.

Елемент $b \in GF(q^a)$ називається *коренем многочлена* $a(x)$ *над* $GF(q^{an})$, якщо $a(b) = 0$. [12] Іншими словами, многочлен $a(x)$ ділиться без остачі на многочлен $x + b$: $a(x) = (x + b)q(x)$.

Важливою властивістю циклічних кодів є те, що всі кодові слова-поліноми кратні одному фіксованому поліному $g(x)$, який має назву *породжуючий поліном коду*.[18] Породжуючий поліном $g(x)$ є дільником

бінома $(x^n - 1)$. Для того, щоб знайти породжуючий поліном, необхідно знати розклад бінома $(x^n - 1)$ на множники $m_j(x), j = 1,2, \dots, l$:

$$(x^n - 1) = m_1(x)m_2(x) \dots m_l(x).$$

Подамо многочлен $(x^n - 1)$ у вигляді добутку мінімальних многочленів:

$$(x^n - 1) = \prod_{i \in J} m_i(x),$$

де J – множина індексів.

Розіб'ємо цю множину на дві підмножини J_g, J_h, які не перетинаються так, що $J = J_g \cup J_h$. Введемо два поліноми:

$$g(x) = \prod_{i \in J_g} m_i(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_r x^r,$$

$$h(x) = \prod_{i \in J_h} m_i(x) = h_0 + h_1 x + h_2 x^2 + \cdots + h_k x^k.$$

Многочлени $g(x)$ і $h(x)$ задовольняють умові $g(x)h(x) = (x^n - 1)$. Якщо поліном $g(x)$ означений як породжуючий, то будь-яке кодове слово можна подати у вигляді $c(x) = a(x)g(x)$, де $a(x)$ – інформаційний поліном.[12] Має місце наступна рівність:

$$c(x)h(x) = a(x)g(x)h(x) = a(x)(x^n - 1).$$

З цієї рівності слідує:

$$c(x)h(x) \equiv 0 \; mod(x^n - 1).$$

Многочлен $h(x)$ називається перевірочним поліномом і обчислюється за формулою:

$$h(x) = \frac{x^n - 1}{g(x)}.$$

Перевірочна матриця циклічного коду C' має вигляд:

$$H = \begin{bmatrix} 0 & 0 & \dots & & 0 & h_k & h_{k-1} & \dots & & h_0 \\ 0 & 0 & \dots & 0 & h_k & & h_{k-1} & \dots & h_0 & 0 \\ \dots & & \dots & & & & & \dots & & \\ 0 & h_{k-1} & \dots & & h_0 & 0 & & \dots & & 0 \end{bmatrix}.$$

Код з породжувальною матрицею H також є циклічним комодом. Кодові слова по поліному $h(x)$ створюють нуль-простір відносно кодових слів, побудованих по поліному $g(x)$.[12]

Алгоритм побудови циклічного коду над $GF(q)$:

а) задати вхідні данні: q, довжина коду n, повідомлення (двійковим кодом), що має бути закодованим;

б) провести факторизацію полінома $(x^n - 1) = \prod_{i \in J} m_i(x)$;

в) обрати поліном $g(x) = \prod_{i \in J_g} m_i(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_r x^r$, степінь $g(x)$ знаходиться за формулою: $deg\big(g(x)\big) = n - k$, де k – довжина кодуючого повідомлення;

г) подаємо кодуюче повідомлення в поліноміальному вигляді $u(x)$ за формулою (2.1);

д) обчислюємо $z(x) = u(x)g(x)$;

е) представляємо $z(x)$ у двійковій формі.

2.10 Висновки до розділу 2

У другому розділі було розглянуто основні поняття електронного підпису та циклічного кодування, досліджено алгоритми криптографії та кодування, що використовують апарат модулярної алгебри. Серед них були: алгоритми електронного підпису RSA, Ель-Гамаля, DSA, Рабіна, GQ, Шнорра і Ніберга-Рюпеля та алгоритм циклічного кодування. Приведено приклади знаходження окремих електронних цифрових підписів.

В наступному розділі буде зроблено програмну реалізацію алгоритмів криптографії та кодування, що були розглянуті в розділі 2.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КРИПТОГРАФІЇ ТА КОДУВАННЯ

У цьому розділі буде продемонстровано приклади програмної реалізації алгоритмів криптографії та кодування, які були розглянуті у другому розділі. Програмна реалізація буде відбуватися у СКА Maple.

3.1 Програмна реалізація алгоритмів криптографії

Приклад 3.1 Реалізація електронного цифрового підпису RSA. Учасник V підписує свій текст $t = DXN$. Учасник W може перевірити підпис учасника V під його текстом t.

Для реалізації електронного цифрового підпису нагадаємо суть алгоритму. Для створення ключів учасник V обчислює свій відкритий ключ і відповідний йому секретний ключ. Для початку учасник задає два прості числа, знаходить їх добуток і функцію Ейлера від значення добутку. Обирається випадкове число e, таке що НОД цього числа і функції Ейлера буде дорівнювати одиниці. Останнім кроком буде знаходження числа a за розширеним алгоритмом Евкліда, має виконуватись умова $ea = 1\ (mod\ \varphi)$. Реалізація першої частини алгоритму продемонстровано на рисунку 3.1.

```
> restart;with(numtheory):
задаємо два довільні прості числа
> n:=1019:m:=2347:
знаходимо їх добуток
> p:=n*m;
                                                    p := 2391593
знаходимо функцію Ейлера
> ph:=phi(p);
                                                    ph := 2388228
задаємоємо k, для якого НОД(k,ph)=1
> k:=35:igcd(k,ph);
                                                        1
перевіряємо виконання для q умови kq=1 (mod ph).
> q:=1569407: k*q mod ph;
                                                        1
Відкритий ключ V, секретний ключ S
> V:=(p,k);S:=q;
                                                    V := 2391593, 35
                                                    S := 1569407
```

Рисунок 3.1 – Реалізація обчислення ключів

Таким чином, відкритий ключ учасника $V-$ це $(p = 2391593, k = 35)$. Секретний ключ для громадянина V – це $q = 1569407$. Реалізація другої частини алгоритму (створення підпису) відбувається за наступною схемою: учасник V підписує документ $t = DXN$, користуючись своїм секретним ключем, а саме: подає текст у вигляді натурального числа r за допомогою 27-ричної системи числення, обчислює хеш-функцію та обчислення безпосередньо значення підпису за формулою $. s = h^q (mod\ p)$. Реалізація другої частини алгоритму продемонстровано на рисунку 3.2.

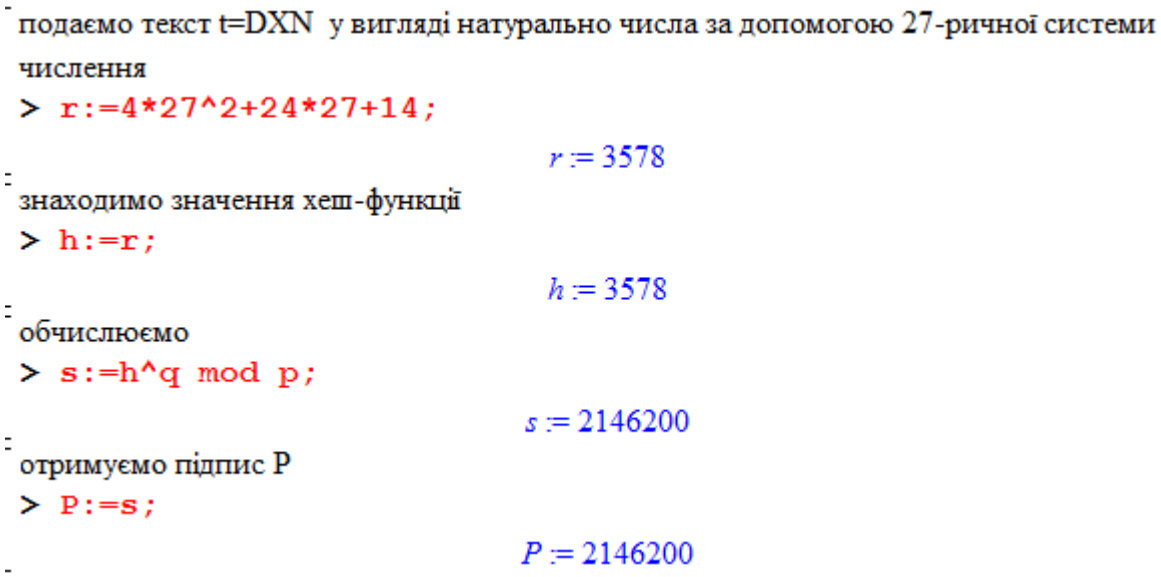

Рисунок 3.2 – Реалізація обчислення підпису

Наступним кроком він відправляє текст документа $t = DXN$ з підписом $s = 2146200$ учаснику W. Далі йде третій етап алгоритму – перевірка підпису. Щоб здійснити перевірку підпису $s = 2146200$ учасника V під його текстом $t = DXN$, учасник W повинен слідувати алгоритму, а саме: отримати відкритий ключ та текст учасника, подати текст у вигляді натурального числа r за допомогою 27-ричної системи числення, задати значення хеш-функції та обчислити $h_1 = s^k (mod\ p)$. Якщо $h = h_1$, то прийняти підпис. Якщо $h \neq h_1$, то відмовити у підписанні. Реалізація третьої частини алгоритму зображена на рисунку 3.3.

```
отримуємо відкритий ключ V. за домогою методу з попереднього етапу
розладаємо текст t=DXN  у вигляді натурально числа за допомогою
27-ричної системи числення
> V:=(p,k); r:=4*27^2+24*27+14;
                              V := 2391593, 35
                                 r := 3578
Обчислюємо
> h:=r;
                                 h := 3578
Обчислюємо
> h1:=s^k mod p;
                                h1 := 3578
Так як  h=h1, то підпис приймаємо
```

Рисунок 3.3 – Реалізація перевірки підпису

Таким чином, бачимо, що $h = 3578 = h_1$, то підпис приймається.

Приклад 3.2 Реалізація електронного цифрового підпису Ель-Гамаля. Учасник V підписує свій текст $t = FIL$. Учасник W може перевірити підпис ечасника Vпід його текстом $t = FIL$.

Для реалізації першої частини алгоритму (створення ключів) учасник V має виконати наступні дії: задати випадкове число p, знайти генератор α і q, та обчислити $x = \alpha^q \ (mod\ p)$. Реалізацію першої частини продемонстровано на рисунку 3.4.

```
Задаємо числа p, q і генератор alpha
> p:=5903:alpha:=2:q:=1751:
Обчислюємо
> x:=alpha^q mod p;
                                 x := 5395
Відкритий ключ - V, секретний ключ - S
> V:=(p,alpha,x);S:=q;
                           V := 5903, 2, 5395
                                 S := 1751
```

Рисунок 3.4 – Реалізація обчислення ключів

Таким чином відкритий ключ для учасника V – це трійка чисел $(p = 5903, \alpha = 2, x = 5395)$. Секретний ключ – число $q = 1751$. Наступним

етапом реалізуємо створення підпису: учасник V підписує свій текст $t = FIL$. Учасник V повинен виконати наступні дії: подати свій текст $t = FIL$ у вигляді натурального числа $m \in [0, p-1]$ за допомогою 27-ричної системи числення числом m, обчислити значення хеш-функції $h(t)$, обрати випадкове секретне ціле число k з $[1, p-2]$ – таке, що НОД $(k, p-1) = 1$, обчислити $k^{-1}\big(mod(p-1)\big)$, $r = \alpha^k (mod\ p)$, $s = k^{-1}(h(t) - q \cdot r)\ \big(mod(p-1)\big)$. Реалізація другої частини алгоритму зображена на рисунку 3.5.

подаємо текст t=FIL у вигляді натурально числа за допомогою 27-ричної системи числення

```
> r:=6*27^2+9*27+12;
```

$r := 4629$

знаходимо значення хеш-функції

```
> h:=r;
```

$h := 4629$

задаємоємо k, для якого НОД(k,ph)=1

```
> k:=1529:igcd(k,p-1);
```

1

Обчислюємо

```
> k^(-1) mod (p-1);
```

1903

Обчислюємо

```
> r:=alpha^k mod p;
```

$r := 2373$

Обчислюємо

```
> s:=k^(-1)*(h-q*r) mod (p-1);
```

$s := 2732$

Підпис - це P

```
> P:=(r,s);
```

$P := 2373, 2732$

Рисунок 3.5 – Реалізація обчислення підпису

Отже, підпис учасника V під його текстом t – є пара чисел $(r = 2373, s = 2732)$. Останнім кроком реалізуємо перевірку підпису. Щоб перевірити підпис $(r = 2373, s = 2732)$ учасника V під його текстом $t = FIL$, учасник W має виконати наступні дії: отримати відкритий ключ $(p = 5903, \alpha = 2, x = 5395)$ учасника V, подати текст $t = FIL$ у вигляді натурального числа $m \in$

$[0, p-1]$ за допомогою 27-ричної системи числення, обчислити значення хеш-функції $h(t)$, перевірити, що $r = 2373 \in [1, p-1]$; якщо ні, то відмовити у підписанні, обчислити $w_1 = x^r \cdot r^s (mod\ p)$ та $w_2 = \alpha^{k(m)} (mod\ p)$. Реалізація третьої частини алгоритму зображена на рисунку 3.6.

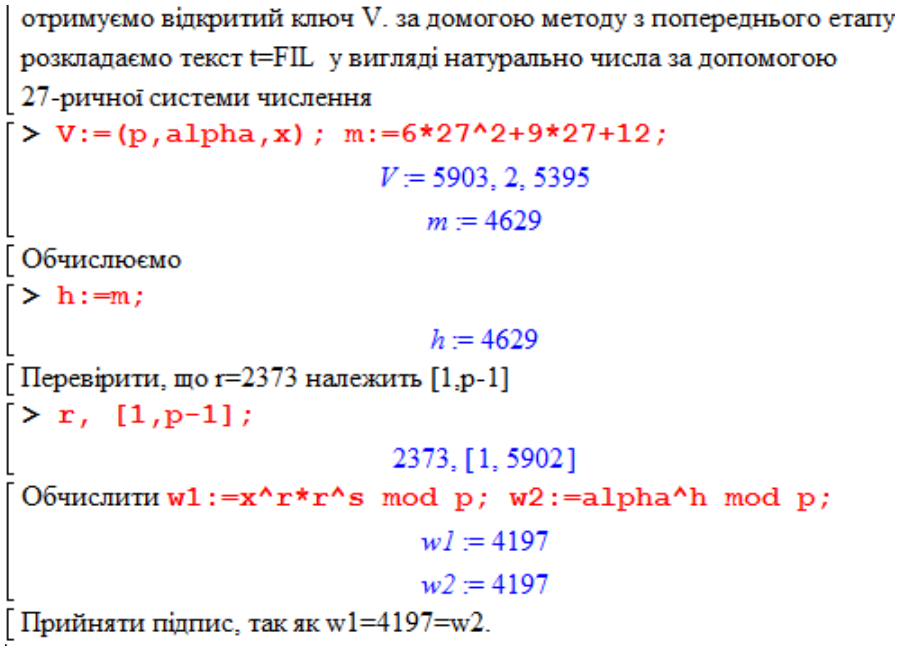

Рисунок 3.6 – Реалізація перевірки підпису

Таким чином, згідно програмною реалізацію алгоритму електронного цифрового підпису, робимо висновок, про прийняття підпису, так як $w_1 = 4197 = w_2$.

Приклад 3.3 Реалізація електронного цифрового підпису DSA. Учасник V підписує свій текст $t = BAN$, а учасник W має перевірити підпис.

Для реалізації першої частини алгоритму (створення ключів) учасник V має виконати наступні дії: обрати випадкові числа q і $p,$, такі, що q ділить $p-1$, обрати випадковий елемент $g \in \mathbb{Z}_p^*$ і знайти $\alpha = g^{\frac{p-1}{q}} (mod\ p)$, якщо $\alpha \neq 1$, то α є генератором для єдиної циклічної підгрупи порядку q у групі $\mathbb{Z}_p^*$, обрати довільне число $a, 1 \leq a \leq q-1$, обчислити $y = \alpha^a (mod\ p)$. Реалізацію першої частини продемонстровано на рисунку 3.7.

```
> restart;with(numtheory):

> q:=27367:
> p:=656809: (p-1)/q;
                                    24
> g:=2732:alpha:=g^((p-1)/q) mod p;
> a:=80;
                                α := 68909
                                 a := 80
> y:=alpha^a mod p;
                                y := 50951
> V:=(p,q,alpha,y);S:=a;
                        V := 656809, 27367, 68909, 50951
                                 S := 80
```

Рисунок 3.7 – Реалізація обчислення ключів

Таким чином відкритий ключ для учасника V – це $(p = 656809, q = 27367, \alpha = 68909, y = 50951)$. Секретний ключ – число $a = 80$. Наступним етапом реалізуємо створення підпису: учасник V підписує свій текст $t = BAN$, користуючись своїм секретним ключем $a = 80$.. Учасник V повинен слідувати наступній схемі: подати текст $t = BAN$ числом m, за допомогою будь-якого методу M, наприклад, в 27-ричній системі числення, обчислити значення хеш-функції $h(t) = h(m) = m$. Обрати довільне таємне ціле число $k, 0 < k < q$, для його НОД$(k, q) = 1$, обчислити $r = \left(a^k \ (mod\ p)\right)(mod\ q)$, $k^{-1}\ (mod\ p)$, $s = k^{-1} \cdot (h(m) + a \cdot r)(mod\ q)$. Реалізація другої частини алгоритму зображена на рисунку 3.8.

```
> m:=2*27^2+1*27+14;
                                m := 1499
> h:=m;
                                h := 1499
> k:=74;igcd(k,q);
                                 k := 74
                                    1
> r:=(alpha^k mod p) mod q;
                                r := 8490
> k^(-1) mod q;
                                  21080
> s:=k^(-1)*(h(m)+a*r) mod q;
                                s := 14746
> P:=(r,s);
                            P := 8490, 14746
```

Рисунок 3.8 – Реалізація обчислення підпису

Отже, підпис учасника V під його текстом t – є пара чисел ($r = 8490$, $s = 14746$). Останнім кроком реалізуємо перевірку підпису. Щоб перевірити підпис $(r = 8490, s = 14746)$ учасника V під його текстом $t = BAN$, учасник W повинен виконати наступні дії: взяти відкритий ключ ($p = 656809$, $q = 27367, \alpha = 68909, y = 50951$) учасника V, подати текст $t = BAN$ числом m в 27-ричній системі числення, обчислити значення хеш-функції $h(t) = h(m)$, перевірити, що $r = 8490$ $0 < r < 27367$ і $s = 14746$ $0 < s < 27367$, обчислити $w = s^{-1}\,(mod\ q)$, $u_1 = w \cdot h(t)(mod\ q)$ і $u_2 = r \cdot w(mod\ q)$, $v = \left(\alpha^{u_1} \cdot y^{u_2}(mod\ p)\right)(mod\ q)$. Реалізація третьої частини алгоритму зображена на рисунку 3.9.

```
отримуємо відкритий ключ V. за домогою методу з попереднього етапу розкладаємо текст
t=BAN  у вигляді натурально числа за допомогою 27-ричної системи числення
> V:=(p,q,alpha,y); m:=2*27^2+1*27+14;
                        V := 656809, 27367, 68909, 50951
                                    m := 1499
> h:=m;
                                    h := 1499
> w:=s^(-1) mod q;
                                   w := 15699
> u1:=w*h mod q; u2:=r*w mod q;
>
                                  u1 := 24548
                                  u2 := 7220
> v:=(alpha^u1*y^u2 mod p)mod q;
                                   v := 8490
> v,r;
                                  8490, 8490
```

Рисунок 3.9 – Реалізація перевірки підпису

Таким чином, згідно з програмною реалізацію алгоритму електронного цифрового підпису, робимо висновок, про прийняття підпису, так як $v = 8490 = r$.

Приклад 3.4 Реалізація електронного цифрового підпису Шнорра. Учасник V підписує свій текст $m = 11101101$, а учасник W має перевірити підпис.

Для реалізації першої частини алгоритму (створення ключів) учасник V має виконати наступні дії: обрати прості числа p і q, такі, щоб $p-1 \backslash q$ *націлo*, обрати елемент $g \in \mathbb{Z}_p^*$ та знайти $\alpha = g^{\frac{p-1}{q}} (mod\ p)$, якщо $\alpha \neq 1$, то генератор $\alpha \in \mathbb{Z}_p^*$ утворює єдину циклічну підгрупу поряду q в групі $\mathbb{Z}_p^*$, обрати довільне число $a, 1 \leq a \leq q-1$, обчислити $y = \alpha^a (mod\ p)$. Реалізацію першої частини продемонстровано на рисунку 3.10.

```
> restart;with(numtheory):
> p:=129841;q:=541;(p-1)/q;
                              p := 129841
                               q := 541
                                 240
> g:=26346:alpha:=g^((p-1)/q) mod p;
                               α := 26
> a:=423: y:=alpha^a mod p;
                              y := 115917
> V:=(p,q,alpha,y);S:=a;
                        V := 129841, 541, 26, 115917
                               S := 423
```

Рисунок 1.10 – Реалізація створення ключів

Таким чином відкритий ключ для учасника V – це $(p = 129841, q = 541, \alpha = 26, y = 115317)$. Секретний ключ – число $a = 423$. Наступним етапом реалізуємо створення підпису: учасник V підписує свій бінарний текст $m = 11101101$, користуючись своїм секретним ключем a.. Учасник V повинен слідувати наступній схемі: обрати випадкове секретне число $k, 1 \leq k \leq q-1$, обчислити $r = \alpha^k (mod\ p), e = h(m||r), s = ae + k (mod\ q)$.. Реалізація другої частини алгоритму зображена на рисунку 3.11.

```
> k:=327:r:=alpha^k mod p;
                                  r := 49375
> e:=155:s:=a*e+k mod q;
                                   s := 431
> P:=(s,e);
                                 P := 431, 155
```

Рисунок 3.11 – Реалізація утворення підпису

Отже, підпис учасника V під його текстом t – є пара чисел $(s = 431, e = 155)$. Останнім кроком реалізуємо перевірку підпису. Щоб перевірити підпис $(s = 431, e = 155)$ учасника V, учасник W повинен виконати наступні дії: отримати відкритий ключ $(p = 129841, q = 541, \alpha = 26, y = 115317)$ учасника V, oбчислити $v = \alpha^s \cdot y^{-e}(mod\ p), e' = h(m||v)$. Реалізація третьої частини алгоритму зображена на рисунку 3.12.

```
3
> v:=alpha^s*y^(-e) mod p;
                                  v := 49375
> e;e1:=155;
                                     155
                                  e1 := 155
```

Рисунок 3.12 – Реалізація перевірки підпису

Таким чином, згідно з програмною реалізацію алгоритму електронного цифрового підпису, робимо висновок, про прийняття підпису, так як $e' = e$.

3.2 Програмна реалізація алгоритму кодування

Приклад 3.5 Реалізація алгоритму циклічного кодування. Потрібно закодувати повідомлення $l = 0110110$ за допомогою циклічного коду довжиною $n = 15$ над $GF(2)$.

Для реалізації першої частини алгоритму (знаходження породжувального многочлена) виконаємо наступні дії: задамо вхідні данні: p, довжина коду n, повідомлення, що має бути закодованим (двійковим кодом); проведемо факторизацію полінома $(x^n - 1) = \prod_{i \in J} m_i(x)$; оберемо поліном $g(x) = \prod_{i \in J_g} m_i(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_r x^r$, степінь $g(x)$ знаходиться за формулою: $deg\big(g(x)\big) = n - k = 15 - 7 = 8$. Реалізацію першої частини продемонстровано на рисунку 3.13.

```
> n:=15: k:=7: l:=[0,1,1,0,1,1,0];
                    l := [0, 1, 1, 0, 1, 1, 0]
> o:=Factor(x^15+1) mod 2;
o := (x + 1) (x^2 + x + 1) (x^4 + x + 1) (x^4 + x^3 + 1)
   (x^4 + x^3 + x^2 + x + 1)
> m0:=x+1;m1:=x^2+x+1;m3:=x^4+x+1;m5:=x^4+x^3+
  1;m7:=x^4+x^3+x^2+x+1;
                        m0 := x + 1
                      m1 := x^2 + x + 1
                      m3 := x^4 + x + 1
                     m5 := x^4 + x^3 + 1
                  m7 := x^4 + x^3 + x^2 + x + 1
> g(x):=expand(m3*m7) mod 2;
                  g(x) := x^8 + x^7 + x^6 + x^4 + 1
```

Рисунок 3.13 – Реалізація обчислення породжуючого полінома

Друга частина алгоритму циклічного кодування повідомлення полягає в наступному: подамо кодуюче повідомлення в поліноміальному вигляді $u(x)$ за

формулою (2.1), обчислимо $z(x) = u(x)g(x)$, представимо $z(x)$ у двійковій формі. Програмну реалізацію можна побачити на рисунку 3.14.

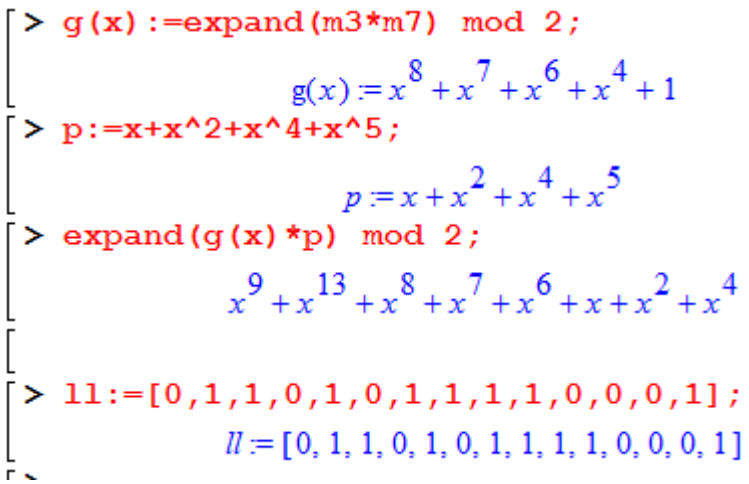

Рисунок 3.14 – Реалізація циклічного кодування повідомлення

Таким чином, повідомлення $l = 0110110$ циклічним кодуванням перетворюється в $ll = 01101011110001$.

3.3 Висновки до розділу 3

Отже, у третьому розділі виконанана програмна реалізацію таких алгоритмів криптографії, як: алгоритм електронного цифрового підпису RSA, Ель-Гамаля, DSA, Шнорра. Також зроблено програмну реалізацію циклічного кодування.

ВИСНОВКИ

В кваліфікаційній роботі було розглянуто основні поняття криптографії, кодування та модулярної алгебри, основні відомості про електронний цифровий підпис та про циклічні кодування, ознайомлено з класифікацію криптосистем, більш детально було розглянуто криптосистеми: RSA, Ель-Гамаля, Рабіна.

У другому розділі було досліджено алгоритми криптографії та кодування, а саме: алгоритми електронного цифрового підпису RSA, Ель-Гамаля, DSA, Рабіна, GQ, Шнорра, Ніберга-Рюпеля та алгоритм циклічного кодування. Наведено приклади створення та перевірки електронного цифрового підпису.

В третьому розділі було написано програмна реалізація алгоритмів криптографії та кодування: алгоритми електронного цифрового підпису RSA, Ель-Гамаля, DSA, Шнорра і алгоритм циклічного кодування в системі Maple. Розглянуто основні проблеми, що вникають під час використання симетричних і асиметричних криптосистем. Отримано розв’язки конкретних прикладів.

Таким чином, мета кваліфікаційної роботи магістра досягнута, досліджено алгоритми криптографії та кодування, що використовують апарат модулярної алгебри, наведено приклади програмної реалізації досліджених алгоритмів в СКА Maple15.

ПЕРЕЛІК ПОСИЛАНЬ

1. Авдошин С. М., Набебин А. А. Дискретная математика. Модулярная алгебра, криптография, кодирование: учеб. пособ. Москва : ДМК Пресс, 2016. 352 с.

2. Алфёров А. П., Зубов А. Ю., Кузьмин А. С., Черёмушкин А. В. Основы криптографии: учеб. пособ. Москва : Гелиос АРВ, 2001. 480 с.

3. Бабак В. П. Теоретичні основи захисту інформації. Київ : НАУ, 2008. 752 с.

4. Бабаш А. В., Шанкин Г. П. Криптография. Москва : СОЛОН-Р, 2002. 512 с.

5. Бабенко Л. К., Ищукова Е. А., Маро Е. А., Сидоров И. Д., Кравченко П. П. Развитие криптографических методов и средств защиты информации. *Известия ЮФУ. Технические науки*. 2012. №4. URL: https://cyberleninka.ru/article/n/razvitie-kriptograficheskih-metodov-i-sredstv-zaschity-informatsii (дата звернення 12. 07. 2020).

6. Баричев С. Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. Москва : Горячая линия – Телеком, 2001. 120 с.

7. Богуш В. М., Мухачов В. А. Криптографічні застосування елементарної теорії чисел : навч. посіб. Київ : ДУІКТ, 2006. 126 с.

8. Данилова О. Ю., Думачев В. Н. Математические основы криптографии : учеб. Воронеж : Воронежский институт МВД России, 2008. 240 с.

9. Домашев А. В. Программирование алгоритмов защиты информации: учеб. пособ. Москва : Нолидж, 2000. 288 с.

10. Жуков-Емельянов О. Д. Информационные технологии на основе модулярной алгебры: учеб. пособ. Москва : Красанд, 2010. 248 с.

11. Жураковський Ю. П., Полторак В. П. Теорія інформації та кодування. Киїі : Вища школа, 2001. 255 с.

12. Евдокимов А. А. Разработка математических моделей модулярных нейронных вычислительных структур для решения задач защиты данных в

компьютерных сетях : дис… канд. тех. наук / Ставропольский государственный университет. Ставрополь, 2004.

13. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. Львів : БаК, 2003. 144 с.

14. Кононович В. Г., Стайкуца С. В., Тардаскіна Т. М., Шинкарчук Т. М. Забезпечення інформаційної безпеки цифрових програмно керованих АТС. Інформаційна безпека телефонного зв'язку: навч. посіб. Одеса : ОНАЗ ім. О.С. Попова, 2010. 168 с.

15. Молдовян Н. А., Абросимов И. К., Ковалева И. В. Постквантовый протокол бесключевого шифрования. *Вопросы защиты информации*. 2017. № 3 (118). С. 3–13.

16. Нестерова Л. Ю., Карпенкова Н. В. Создание криптографии с помощью модулярной математики. *Молодой ученый*. 2014. № 21.1. С. 237–240.

17. Романюк М. І., Савченко Ю. Г. Основи теорії інформації та кодування. Київ : КПІ ім. Ігоря Сікорського,2019. 70 с.

18. Романьков В. А. Эффективные методы алгебраического криптоанализа и защита от них. *ПДМ.* 2019. №3 (21) URL: https://cyberleninka.ru/article/n/effektivnye-metody-algebraicheskogo-kriptoanaliza-i-zaschita-ot-nih (дата звернення 30.07.2020)

19. Романьков В. А. Диофантова криптография на бесконечных группах. *ПДМ*. 2012. №2 (16). URL: https://cyberleninka.ru/article/n/diofantova-kriptografiya-na-beskonechnyh-gruppah (дата звернення 27. 08. 2020).

20. Саломатин С. Б. Исследование циклических кодов: метод. пособие по курсам «Теория кодирования и защита информации» для студ. радиотех.спец. всех форм обуч. Минск : БГУИР, 2008. 32 с.

21. Семёнова Н. Ф. Математические модели модулярной алгебры для систем пролонгированной защиты данных с "блуждающими" ключами в распределённых вычислительных системах : дис… канд. ф.-м. наук / Ставропольский государственный университет. Ставрополь, 2004.

22. Черемушкин, А. В. Криптографические протоколы. Основные свойства и уязвимости. Москва : Академия, 2009. 272 с.

23. Фільштінський В. А., Бережний А. В. Математичні основи криптографії. Суми : Сумський державний університет, 2011. 138 с.

24. Франчук В. М. Захист інформаційних ресурсів : криптографічні та стеганографічні методи захисту даних: посібник для викладачів, вчителів та студентів інформатичних спеціальностей. Київ : НПУ ім. М. П. Драгоманова. Інститут інформатики, 2012. 120 с.

25. Ященко В. В., Нестернко Ю. В. Введение в криптографию : учебник, 4-е издание. Москва : МЦНМО, 2012. 348 с.