Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

з навчальної дисципліни

“ОСНОВИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ”

для студентів денної та заочної форм навчання
за напрямом підготовки 6.170103 “Управління
інформаційною безпекою”,
спеціальністю 125 “Кібербезпека”

Видання оновлене та доповнене

ЗАТВЕРДЖЕНО
Протокол засідання кафедри
кібербезпеки та програмного
забезпечення
29.08.2018 № 1

Кропивницький
2018

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни “Основи криптографічного захисту інформації” [для студ. денної та заочної форми навч. за напрямом підготовки 6.170103 “Управління інформаційною безпекою”, спеціальністю 125 “Кібербезпека”] Видання оновлене та доповнене / Уклад. І. А. Лисенко – Кропивницький: ЦНТУ, 2018.– 64 с.

Укладач Лисенко І.А., канд. техн. наук

Рецензенти: Смірнов О. А., д-р техн. наук, професор;
Якименко Н. М., канд. фіз.-мат. наук, доцент.

Методичні вказівки висвітлюють організаційні та практичні аспекти виконання лабораторних робіт з навчальної дисципліни “Основи криптографічного захисту інформації” для студентів денної та заочної форм навчання за напрямом підготовки 6.170103 “Управління інформаційною безпекою”, спеціальністю 125 “Кібербезпека”, а також рекомендації щодо ходу виконання робіт, підготовки та представлення отриманих результатів.

ЗМІСТ

Вступ

Інформація з точки зору інформаційної безпеки має наступні категорії:

- конфіденційність гарантія того, що конкретна інформація доступна тільки тому колу осіб, для кого вона призначена; порушення цієї категорії називається розкраданням або розкриттям інформації
- цілісність гарантія того, що інформація зараз існує в її початковому виді, тобто при її зберіганні або передачі не було вироблено несанкціонованих змін; порушення цієї категорії називається фальсифікацією повідомлення
- автентичність гарантія того, що джерелом інформації є саме та особа, яка заявлена як її автор; порушення цієї категорії також називається фальсифікацією, але вже автора повідомлення
- апельованість досить складна категорія, але часто вживана в електронній комерції гарантія того, що при необхідності можна буде довести, що автором повідомлення є саме заявлена людина, і не може являтися ніхто інший; відмінність цієї категорії від попередньої в тому, що при підміні автора, хтось інший намагається заявити, що він автор повідомлення, а при порушенні апельованості сам автор намагається "відхреститися" від своїх слів, підписаних їм одного дня.

Відносно **інформаційних систем** застосовуються інші категорії:

- надійність гарантія того, що система поводиться в нормальному і позаштатному режимах так, як заплановано
- точність гарантія точного і повного виконання усіх команд
- контроль доступу гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів, і ці обмеження доступу постійно виконуються
- контрольованість гарантія того, що у будь-який момент може бути вироблена повноцінна перевірка будь-якого компонента програмного комплексу
- контроль ідентифікації гарантія того, що клієнт, підключений в даний момент до системи, є саме тим, за кого себе видає
- стійкість до умисних збоїв гарантія того, що при умисному внесенні помилок в межах заздалегідь обумовлених норм система поводитиметься так, як обумовлено заздалегідь.

Сама криптографія не є вищим ступенем класифікації суміжних з нею дисциплін. Навпаки, криптографія спільно з криптоаналізом (метою якого є

протистояння методам криптографії) складають комплексну науку криптологію.

Основною схемою класифікації усіх криптоалгоритмів є наступна:

1. Тайнопис. Відправник і одержувач виробляють над повідомленням перетворення, відомі тільки їм двом. Стороннім особам невідомий сам алгоритм шифрування. Деякі фахівці вважають, що тайнопис не є криптографією взагалі, і автор знаходить це досконало справедливим.

2. Криптографія з ключем. Алгоритм дії на передавані дані відомий усім стороннім особам, але він залежить від деякого параметра "ключа", який мають тільки відправник і одержувач.

3. Симетричні криптоалгоритми. Для зашифровування і розшифровки повідомлення використовується один і той же блок інформації (ключ).

4. Асиметричні криптоалгоритми. Алгоритм такий, що для зашифровування повідомлення використовується один ("відкритий") ключ, відомий таким, що усім бажає, а для розшифровки інший ("закритий"), існуючий тільки у одержувача.

Залежно від характеру дій, що здійснюються над даними, алгоритми підрозділяються на:

1. **Перестановочні** - Блоки інформації (байти, біти, більші одиниці) не змінюються самі по собі, але змінюється їх порядок дотримання, що робить інформацію недоступною сторонньому спостерігачеві.

2. **Підстановлювальні** - Самі блоки інформації змінюються за законами криптоалгоритма. Переважна більшість сучасних алгоритмів належать цій групі.

Залежно від розміру блоку інформації криптоалгоритми діляться на:

1. Потокові шифри. Одиницею кодування є один біт. Результат кодування не залежить від того, що пройшло раніше вхідного потоку. Схема застосовується в системах передачі потоків інформації, тобто в тих випадках, коли передача інформації починається і закінчується в довільні моменти часу і може випадково уриватися. Найбільш поширеними представниками потокових шифрів являються скремблери.

2. Блокові шифри. Одиницею кодування є блок з декількох байтів (нині 432). Результат кодування залежить від усіх початкових байтів цього блоку. Схема застосовується при пакетній передачі інформації і кодуванні файлів.

Криптоалгоритм іменується ідеально стійким, якщо прочитати зашифрований блок даних можна тільки перебравши усі можливі ключі, до тих пір, поки повідомлення не виявиться осмисленим. Оскільки по теорії вірогідності шуканий ключ буде знайдений з вірогідністю 1/2 після перебору половини усіх ключів, то на злом ідеально стійкого криптоалгоритма з ключем довжини N буде потрібно в середньому 2N1 перевірок. Таким чином, в загальному випадку стійкість блокового шифру залежить тільки від довжини ключа і зростає експоненціально з її зростанням. Навіть припустивши, що перебір ключів виробляється на спеціально створеній багатопроцесорній системі, в якій завдяки діагональному паралелізму на перевірку 1 ключа йде тільки 1 такт, то на злом 128 бітового ключа сучасній техніці буде потрібно не менше 1021 року. Природно, усе сказане відноситься тільки до ідеально стійких шифрів, якими, наприклад, з великою часткою упевненості являються приведені в таблиці вище алгоритми.

Окрім цієї умови до ідеально стійких криптоалгоритмам застосовується ще одна дуже важлива вимога, якій вони повинні обов'язково відповідати. При відомих початковому і зашифрованому значеннях блоку ключ, яким вироблено це перетворення, можна дізнатися також тільки повним перебором. Ситуації, в яких сторонньому спостерігачеві відома частина початкового тексту зустрічаються повсюдно. Це можуть бути стандартні написи в електронних бланках, фіксовані заголовки форматів файлів, досить довгі слова, що часто зустрічаються в тексті, або послідовності байт. У світлі цієї проблеми описана вище вимога не є нічим надмірним і також строго виконується стійкими криптоалгоритмами, як і перше.

V СЕМЕСТР
Лабораторна робота № 1

Тема: *Комбінаторні схеми (перестановки, сполучення, розміщення) без повторень*

Загальні відомості

I. Перестановки впорядкованих множин.

Перестановки враховують порядок елементів в підмножини. Дві упорядковані множини вважаються різними, якщо їх порядок не співпадає.

Приклад: Скількома різними способами можна впорядкувати скінчену множину A з n елементів.

Визначення: Впорядкована множини, що відрізняються лише порядком своїх елементів, називаються *перестановками* і позначаються як $P_n = n!$.

Теорема: Кількість перестановок із n елементів рівна $n!$, тобто $P_n = n!$.

Доведення: Будемо послідовно вибирати елементи множини A і розміщати їх в додатному порядку на n місцях. На перше місце можна поставити будь-який з n елементів, на друге – $(n-1)$ і т. д. Аж до одного, тобто: $n(n-1)(n-2)...1 = n!$ Теорема доведена.

Зауваження: Якщо провести паралель між перестановками та розміщеннями без повторень, то $P_n = A_n^n = n!$

Зауваження: Перестановки $\pi = (\pi_1 \quad \pi_2 \quad \dots \quad \pi_n)$ елементів 1, 2, ..., n записують і в матричній формі $\pi = \begin{pmatrix} 1 & 2 & \dots & n \\ \pi_1 & \pi_2 & \dots & \pi_n \end{pmatrix}$, де верхній рядок – це порядкові номера позицій елементів в перестановці; нижній рядок – той самий набір чисел взятий в будь-якому порядку; π_j - номер елемента на j-му місці перестановки. Порядок стовпців в перестановках, записаних в матричній формі, не є важливим, так як в цьому випадку номер позиції кожного елемента перестановки вказується явно у верхньому рядку.

Наприклад: перестановка (3,2,4,1) може бути записана як: $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$ або $\begin{pmatrix} 3 & 1 & 4 & 2 \\ 4 & 3 & 1 & 2 \end{pmatrix}$, $\begin{pmatrix} 2 & 1 & 4 & 3 \\ 2 & 3 & 1 & 4 \end{pmatrix}$ і т. д.

Задача: *(про людей)* Скількома способами можна розмістити на дошці для шахів 8 турів так, щоб вони „не побили" один одного?

Розв'язання: Умова „не могли побити" означає, що на кожній горизонталі та вертикалі може стояти лиш одна тура. Тоді кожному розміщенню турів на дошці відповідає перестановка $\pi = \begin{pmatrix} 1 & 2 & \dots & 8 \\ \pi_1 & \pi_2 & \dots & \pi_8 \end{pmatrix}$. Верхній рядок перестановки – це номери горизонталей, нижня – вертикалей, перехрещення яких визначає положення турів на дошці. Звідси слідує, що число розміщень рівне числу перестановок $P_8 = 8!$ з 8 елементів.

II. Розміщення елементів множини.

Визначення: Впорядкована k елементна підмножина множини A, всі елементи якої різні між собою, називаються *розміщенням без повторень.*

Маємо n різних предметів $a_1, a_2, ..., a_n$. Скільки з них можливо розставити розташувань довжини k? Два розташування вважаються різними, якщо вони відрізняються видом елементів, що до них входять, або порядком їх розташування. Такі розташування називаються розміщеннями без повторень, а їх число позначають як A_n^k. При створенні даних розміщень на перше місце можна поставити будь-який із n предметів. На друге місце можна поставити тільки будь-який із $n-1$, що залишилися. А на k-е місце – будь-який із $n-k+1$ решти предметів. З правила прямого добутку випливає:

Теорема: *(про кількість впорядкованих k елементних розміщень без повторень)*

$$A_n^k = k!C_n^k = n(n-1)...(n-k+1) = \frac{n!}{(n-k)!}.$$

Задача: Нехай дана деяка невпорядкована n елементна множина A. Скільки різних впорядкованих k-елементних підмножин може мати множина A?

Розв'язання: Розглянемо лише варіант, коли підмножина має k різних між собою елементів. Оскільки множина A невпорядкована, то будь-які її k елементні підмножини можуть бути впорядковані $k!$ способами, а число всіх можливих різних розміщень з n елементів по k *і рівні* C_n^k. А за основним правилом комбінаторики: $k!C_n^k$. Задача розв'язана.

Задача: В хокейному турнірі брали участь 17 команд. Розігрувалися золоті, срібні та бронзові медалі. Скількома способами можуть бути розподілені медалі?

Розв'язання: 17 команд претендують на 3 місця. Тоді трійку призерів можна вибрати способами $A_{17}^3 = \frac{17!}{14!} = 4080$.

III. <u>Сполучення (комбінації) елементів.</u>

У випадках, коли нас не цікавить порядок елементів в розташуванні, а цікавить тільки її склад, то говорять про сполучення. <u>*Сполученнями*</u> з n різних елементів по k називають всі можливі розташування довжини k, створені із цих елементів, що відрізняються один від одного складом, але не порядком елементів. Загальне число сполучень позначають через C_n^k або $\binom{n}{k}$. Визначимо це число. Складемо всі сполучення з n по k. Потім переставимо в кожному сполученні елементи всіма можливими способами. Тепер ми отримали розстановки, що відрізняються або складом, або порядком, тобто це все розміщення без повторень з n по k. Їх число рівне A_n^k. Враховуючи, що кожне сполучення дає $k!$ розміщень, то за правилом добутку можна записати $C_n^k \cdot k! = A_n^k$. Тоді $C_n^k = \frac{n(n-1)\ldots(n-k+1)}{k!}$ або $C_n^k = \frac{n!}{k!(n-k)!}$ та $C_n^k = C_n^{n-k}$.

Задача (про прямокутники)***:*** Скільки різних прямокутників можна вирізати із клітин дошки, розмір якої $m \times n$?

Розв'язання: Прямокутник однозначно визначається положенням його сторін. Горизонтальні сторони можуть займати будь-яке із $m+1$ положення. Тоді число стовпців їх вибору рівне C_{m+1}^2. Вертикальні сторони можна вибрати C_{n+1}^2 способами. З правила прямого добутку випливає, що кількість прямокутників рівне $C_{m+1}^2 \cdot C_{n+1}^2$.

Завдання:

1. Скількома різними способами можна розмістити 7 книжок на книжковій полиці?

2. Скільки можливих телефонних номерів, що складаються з: а) 6 цифр; б) 7 цифр; якщо цифри не повторюються?

3. Скількома способами можна упорядкувати n елементну множину з числами 1, 2, ..., n, таким чином, щоб два останнні елемента були зафіксованими, тобто рівними n, (n-1)?

4. В групі є 20 студенів. Скількома способами можна вибрати старосту, заступника старости та профорга групи?

5. Із цифр 1, 2, 3, 4, 5 утворюють п'ятицифрові числа, не кратні п'яти і такі, що не містять однакових цифр. Скільки існує таких чисел?

6. У турнірі беруть участь шість осіб. Скількома способами можуть розподілитися місця між ними?

7. Скількома способами можна розсадити 8 учнів за вісьмома різними партами так, щоб за кожною сидів лише один учень?

Лабораторна робота № 2

Тема: *Шифрування текстів методом простої перестановки та простої підстановки*

Загальні відомості

Шифрування симетричними ключами використовує єдиний ключ (ключ, що містить безпосередньо набір кодованих значень) і для кодування й для дешифрування. Крім того, алгоритми шифрування й дешифрування - інверсії один одного. Якщо P - звичайний текст, C - зашифрований текст, а K - ключ, алгоритм кодування Ek (x) створює зашифрований текст із вихідного тексту.

Алгоритм же дешифрування Dk (x) створює вихідний текст із зашифрованого тексту. Припустимо, що Ek (x) і Dk (x) інверсні по відношенню одне до одного. Вони застосовуються, послідовно перетворюючи інформацію з одного виду в інший і обернено. Ми маємо

Шифрування: C = Ek(P),

Розшифрування: P = Dk (C),

де, Dk(Ek(x)) = Ek(Dk(x)) = x

Принципи Керкхоффа

Хоча можна припустити, що шифр був би більш безпечний, якщо ми приховуємо й алгоритм шифрування/дешифрування, і ключ засекречування, це не рекомендується. Згідно із принципом Керкхоффа, потрібно завжди припускати, що супротивник знає алгоритм кодування /дешифрування. Протидія шифру атаці повинне базуватися тільки на таємниці ключа. Інакше кажучи, передбачається, що ключ повинен бути настільки важкий, що не треба приховувати алгоритм кодування/дешифрування. Ці принципові положення стануть більш ясні, коли ми будемо вивчати сучасні шифри. Для сучасних шифрів сьогодні існує небагато алгоритмів. Множина ключів (Ключовий домен) для кожного алгоритму, однак, настільки велике число, що заважає супротивникові знайти ключ.

Ми можемо розділити традиційні шифри із симетричним ключем на дві великі категорії: **шифри підстановки й шифри перестановки**. У шифрі підстановки ми заміняємо один символ у зашифрованому тексті на інший символ; у шифрі перестановки - міняємо місцями позиції символів у вихідному тексті.

Шифри підстановки

Шифр підстановки заміняє один символ іншим. Якщо символи у вихідному тексті - символи алфавіту, ми заміняємо одну букву іншої. Наприклад, ми можемо замінити букву A буквою D, а букву T - буквою Z. Якщо символи - цифри (від 0 до 9), ми можемо замінити 3 на 7 і 2 на 6.

Шифри підстановки можуть бути розбиті на **дві категорії**: моноалфавітні або багатоалфавітні шифри.

Розглянемо **моноалфавітні шифри**. У такій підстановці буква (або символ) у вихідному тексті завжди змінюється на ту саму букву (або символ) у зашифрованому тексті незалежно від його позиції в тексті. Наприклад, якщо алгоритм визначає, що буква A у вихідному тексті міняється на букву D, то при цьому кожна буква A змінюється на букву D. Інакше кажучи, букви у вихідному тексті й зашифрованому тексті перебувають у відношенні один до одному.

У моноалфавітної підстановці відносини між буквою у вихідному тексті й буквою в зашифрованому тексті - один до одного.

Приклад 1

Вихідний текст: hello Зашифрований текст: KHOOR

Приклад 2

Вихідний текст: hello Зашифрований текст: ABNZF

Адитивний шифр

Найпростіший моноалфавітний шифр - **адитивний шифр**, його іноді називають **шифром зрушення**, а іноді - **шифром Цезаря**, але термін адитивний шифр краще показує його математичний зміст. Припустимо, що текст складається з маленьких букв (від a до z), а і зашифрований текст складається із заголовних букв (від A до Z). Щоб забезпечити застосування математичних операцій до вихідного й зашифрованого текстів, ми привласнимо кожній букві число (для нижнього й верхнього регістру), як це показано в табл. 1.

Таблиця 1. Кодова таблиця латинських літер

вихідний текст	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
зашифрований текст	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
числове значення	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Кожному символу (нижній регістр або верхній регістр) зіставлене ціле число з . Ключ засекречування також ціле число в . Алгоритм кодування додає ключ до символу вихідного тексту; алгоритм дешифрування віднімає ключ із символу зашифрованого тексту. Усі операції проводяться ключем в .

Шифр зрушення

Історично адитивні шифри називалися **шифрами зрушення** - з тієї причини, що алгоритм шифрування може інтерпретуватися як "клавіша зрушення букви вниз", а алгоритм дешифрування може інтерпретуватися як "клавіші зрушення букви нагору". Наприклад, якщо ключ = 15, алгоритм кодування зрушує букву на 15 букв униз (до кінця алфавіту). Алгоритм дешифрування зрушує букву на 15 букв нагору (до початку алфавіту).

Звичайно, коли ми досягаємо кінця або початку алфавіту, ми рухаємося по кільцю до початку (оголошені властивості операції по модулю 26).

Багатоалфавітні шифри

У багатоалфавітної підстановці кожна поява символу може мати різну заміну. Відносини між символом у вихідному тексті й символом у зашифрованому тексті - "один до багатьом". Наприклад, "a" може бути зашифроване як "D" на початку тексту, але як "N" - у середині. Багатоалфавітні шифри мають перевагу: вони приховують частоту появи символу основної мови. Супротивник не може використовувати статистичну частоту окремого символу, щоб зламати зашифрований текст.

Щоб створити багатоалфавітний шифр, ми повинні зробити кожний символ зашифрованого тексту залежним від відповідного символу вихідного тексту й позиції символу вихідного тексту в повідомленні. Тобто наш ключ повинен бути потоком підключей, у яких кожний підключ так чи інакше залежить від позиції символу вихідного тексту, який використовується для вибору підключа шифрування. Інакше кажучи, ми повинні мати ключовий потік $k = (k_1, k_2, k_3.....)$, у якім ki застосовується, щоб зашифрувати i-тий символ у вихідному тексті й створити i-тий символ у зашифрованому тексті.

Автоключовий шифр

Щоб зрозуміти залежність ключа від позиції, обговоримо простий багатоалфавітний шифр, названий "автоключовим". У цьому шифрі ключ - потік підключей, у якому кожний підключ використовується щоб зашифрувати відповідний символ у вихідному тексті. Перший підключ - певне заздалегідь значення, таємно погоджене обидвими сторонами. Другий підключ - значення першого символу вихідного тексту (між 0 і 25). Третій - i-те значення другого вихідного тексту. І так далі.

Назва шифру, автоключовий, має на увазі, що підключи створюються автоматично залежно від символів шифру вихідного тексту в процесі шифрування.

Шифр Плейфера

Інший приклад багатоалфавітного шифру - Шифр Плейфера, що використовувався британською армією протягом Першої світової війни. Ключ засекречування в цьому шифрі зроблений з 25 букв алфавіту, розміщених у матриці (букви I і J розглядаються при шифруванні як однакові). За допомогою різних угод про розміщення букв у матриці можна створити багато різних ключів засекречування.

Секретный ключ =

L	G	D	B	A
Q	M	H	E	C
U	R	N	I/J	F
X	V	S	O	K
Z	Y	W	T	P

Рис 3. Кодова таблиця шифру Плейфера

Перед шифруванням вихідний текст розбивається на пари; якщо дві букви пари однакові, то, щоб відокремити їх, вставляється фіктивна буква. Після вставки фіктивних букв, якщо число символів у вихідному тексті непарне, наприкінці додається один додатковий фіктивний символ, щоб зробити число символів парним.

Шифр використовує три правила для шифрування:

1. якщо ці дві букви-пари розташовані в одній і тій же рядку таблиці ключа засекречування зашифрований символ, що відповідає, для кожної букви - наступний символ праворуч у тому ж самому рядку (з поверненням до початку рядка; якщо символ вихідного тексту - останній символ у рядку);
2. якщо ці дві букви-пари розташовані в тому самому стовпці таблиці ключа засекречування зашифрований символ, що відповідає, для кожної букви - символ нижче цього в тому ж самому стовпці (з поверненням до початку стовпця; якщо символ вихідного тексту - останній символ у стовпці);
3. якщо ці дві букви-пари не перебувають в одному рядку або стовпці таблиці засекречування зашифрований символ, що відповідає, для кожної букви - символ, який перебуває в його власному рядку, але в тому ж самому стовпці, що й інший символ.

Шифр Віженера

Один цікавий вид багатоалфавітного шифру був створений Блезом де Віженером, французьким математиком шістнадцятого сторіччя. Шифр Віженера використовує різну стратегію створення потоку ключів. Потік ключів - повторення початкового потоку ключа засекречування довжини m, де ми маємо $1 < m < 26$. Шифр може бути описаний у такий спосіб: (k_1, k_2, …., km) - первісний ключ засекречування, погоджений сторонами.

$P = P_1P_2P_3$…..

$C = C_1C_2C_3$…..

$k = [(k_1,k_2), (k_3,k_4),…..]$

Шифрування Ci = ki

Дешифрування Pi = ki

Одна важлива відмінності між шифром Віженера й іншими двома багатоалфавітними шифрами, які ми розглянули: потік ключів Віженера не залежить від символів вихідного тексту; він залежить тільки від позиції символу у вихідному тексті. Інакше кажучи, потік ключів може бути створений без знання суті вихідного тексту.

Одноразовий блокнот

Одна із цілей криптографії - ідеальна таємність. Дослідження Шенона показали, що ідеальна таємність може бути досягнута, якщо символи вихідного тексту зашифровані за допомогою ключа, обраного випадково з деякої області ключів.

Ця ідея використовується в шифрі, який називається **одноразовим блокнотом**. Його винайшов американський інженер Вернам. У цьому шифрі ключ має ту ж саму довжину, що й вихідний текст, і обраний зовсім випадково.

Одноразовий блокнот - ідеальний шифр, але його майже неможливо реалізувати комерційно. Якщо ключ щораз генерується заново, як Аліса може щораз повідомляти Боба новий ключ? Для цього щораз потрібно передавати повідомлення. Однак є деякі випадки, коли можливо використання одноразового блокнота. Наприклад, якщо президент країни повинен передати повністю секретне повідомлення президентові іншої країни, він може перед посилкою повідомлення передати за допомогою довіреного посланника випадковий ключ.

Роторний шифр

Хоча шифри одноразового блокнота не застосовуються на практиці, один крок від нього до більш захищеного шифру - роторний шифр. Він вертається до ідеї моноалфавітної підстановки, але міняє принцип відображення вихідного тексту в символи зашифрованого тексту для кожного символу вихідного тексту

Слово із трьома буквами, такими як "bee", зашифроване як "BAA", якщо ротор нерухливий (моноалфавітний шифр підстановки), але воно буде зашифровано як "BCA", якщо він обертається (роторний шифр). Це показує, що роторний шифр - багатоалфавітний шифр, тому що дві появи того ж самого символу вихідного тексту зашифровані як різні символи.

Роторний шифр є стійким до атаки грубої сили, як моноалфавітний шифр підстановки, тому що супротивник повинен знайти першу множину відображень серед можливих 26! (факторіал). Роторний шифр є набагато більш стійким до статистичної атаки, чому моноалфавітний шифр підстановки, тому що в ньому не зберігається частота вживання букви.

Машина "Энігма"

Машина "Энігма" була спочатку винайдена в Сербії, але була змінена фахівцями німецької армії й інтенсивно використовувалася протягом Другої Світової Війни. Машина базувалася на принципі шифрів ротора.

Шифри перестановки

Шифр перестановки не заміняє одним символом іншої, замість цього він змінює місце розташування символів. Символ у першій позиції вихідного тексту може з'явитися в десятій позиції зашифрованого тексту. Символ, який перебуває у восьмій позиції вихідного тексту, може з'явитися в першій позиції зашифрованого тексту. Інакше кажучи, шифр перестановки ставить в іншому порядку (переміщає) символи.

Шифри перестановки без використання ключа

Прості шифри перестановки, які застосовувалися в минулому, не використовували ключ. Є два методи для перестановки символів. У першому методі текст записується в таблиці стовпець за стовпцем і потім передається рядок за рядком. У другому методі текст написаний у таблиці рядок за рядком і потім передається стовпець за стовпцем.

Ключові шифри перестановки

Безключові шифри переставляють символи, використовуючи запис вихідного тексту одним способом (наприклад, рядок за рядком) і передачу цього тексту в іншому порядку (наприклад, стовпець за стовпцем). Перестановка робиться у всьому вихідному тексті, щоб створити весь зашифрований текст. Інший метод полягає в тому, щоб розділити вихідний текст на групи заздалегідь певного розміру, названі блоками, а потім використовувати ключ, щоб переставити символи в кожному блоці окремо.

Об'єднання двох підходів

Сучасні шифри перестановки, щоб досягти кращого скремблювання, поєднують два підходи. Шифрування й дешифрування робиться в три кроки. Перший: текст пишеться таблицею рядок за рядком. Другий: робиться перестановка, змінюючи порядок проходження стовпців. Третій: стовпець за стовпцем читається нова таблиця. Перші й треті кроки забезпечують безключове глобальна зміна порядку проходження; другий крок забезпечує блокову ключову перестановку. Ці типи шифрів згадуються часто як ключові шифри перестановки стовпців.

Шифри з подвійною перестановкою можуть утруднити роботу криптоаналітика. На кожному кроці може застосовуватися різний ключ, але звичайно ключ використовується той самий.

Завдання:

За допомогою будь-якої мови програмування, реалізувати наступні задачі:

1. Використовуючи свій номер у журналі групи як ключ шифрування, зашифрувати методом Цезаря та Віженера своє прізвище, ім’я та по-батькові.

2. За допомогою ключового шифру перестановки зашифрувати та розшифрувати будь-який текст розміром не менше 3 сторінок.

Лабораторна робота № 3

Тема: *Бінарні операції. Алгебра логіки. Логічні константи та змінні*

Загальні відомості

Сучасна логіка — надзвичайно абстрактна дисципліна, але, оскільки немає нічого більш практичного за гарну теорію, не дивно, що логіка має цілий ряд емпіричних застосувань.

Найочевидніше застосування — електроніка та електротехніка: по-перше, всі сучасні електричні та електронні схеми проектуються засобами алгебри логіки (тобто, зрештою, логіки висловлювань); по-друге, процесори електронних приладів є прямим матеріальним втіленням логічної теорії обчислюваності, теорії алгоритмів. Це — практично матеріалізація логіки. Тому кожного дня, вмикаючи комп'ютер чи приймаючи дзвінок на мобільний телефон, ми неявно визнаємо практичний характер логіки.

Другим полем для застосування вже давно є криптографія. Приміром, сучасні способи шифрування з розділенням алгоритмів зашифровки і розшифровки не піддаються зламу і евристичному аналізу — бо засновані на теоремі Чорча про нерозв'язність. Кожен, хто створює для своєї електронної кореспонденції в інтернеті ключ GPG або PGP, тим самим визнає прикладний характер цієї теореми. (теза Чорча у загальному вигляді стверджує: будь-яка інтуїтивно розв'язна функція може бути частково розв'язною за допомогою будь-якого фізичного пристрою, а також – якщо у будь-якій мові програмування виклик деякої функції за іменем та за значенням приводить до деякого результату, то це завжди однаковий результат).

Звернемося до основних понять алгебри логіки.

Висловлення – це форма мислення, у якій стверджується зв'язок між предметом і його ознакою або відношення між предметами і яка виражає або істину, або хибність.

Логічні операції

Кон'юнкція(лат. conjangere – об'єднувати)(операція AND) – бінарна логічна операція, що має значення «істина», якщо всі операнди мають значення «істина». Позначають у математиці та логіці як ^, у програмуванні – як and.

A	B	A^B
хибність	хибність	хибність
хибність	істина	хибність
істина	хибність	хибність
істина	істина	істина

Диз'юнкція(лат.disjunctio)(операція OR)- бінарна логічна операція, що має значення , що має значення «істина», якщо хоча б один з операндів має значення «істина». Позначають у математиці та логіці як v, у програмуванні – як or.

A	B	A v B
хибність	хибність	хибність
хибність	істина	істина
істина	хибність	істина
істина	істина	істина

Виключна диз'юнкція(також операція XOR, додавання за модулем 2) - бінарна логічна операція, що набуває значення «істина» тоді і тільки тоді, коли значення «істина» має рівно один з її операндів.

Для запису операції використовують позначення: a xor b.

A	B	A xor B
хибність	хибність	хибність
хибність	істина	істина
істина	хибність	істина
істина	істина	хибність

Заперечення в логіці – унарна операція над судженнями, результатом якої є судження, у відомому сенсі «протилежне» початковому.

A	not A
хибність	істина
істина	хибність

Результат виконання логічних операторів має тип boolean.

Логічні вирази

Логічні вирази – це вирази, що складаються з висловлювань, які можна з'єднати логічними зв'язками. Ці вирази набувають логічного значення(«хибне» або «істинне»).Логічні вирази можуть бути простими та складними.

У простому логічному виразі використовують змінні та константи логічного типу, операції порівняння. Сполучення простих логічних виразів за допомогою логічних операцій утворює складений логічний вираз.

Приклади складання умов з логічною операцією and

Твердження	мова
Число x належить інтервалу [-2, 5], тобто -2<=x<=5	(x>=-2)and(x<=5)
Ціле число X двозначне	(x div 100=0)and(x div 10=0)
Натуральне число X кратне 7 та закінчується цифрою 3	(x mod 7=0)and(x mod 10=3)
Кожне з двох цілих чисел X та Y парне	(x mod 2=0)and(y mod 2=0)
Точка з координатами (x, y) попадає в заштриховану ділянку площини	(x>=-3)and(x<=-1) and (y>=1)and(y<=3)

Приклади складання умов з логічною операцією or

Твердження	мова
Хоча б одне з чисел X та Y додатне	(x>0)or(y>0)
Натуральне число X закінчується цифрою 2 або 3	(x mod 10=2)or(x mod 10=3)
Точка з координатами X та Y належить I або III чверті координатної площини	(X>0)and(y>0)or(x<0)and(y<0)

Приклади складання умов з логічною операцією xor

Твердження	мова без xor	мова з xor
Тільки одне з чисел X та Y парне	(x mod 2=0)and(y mod 2<>0) or (x mod 2<>0)and(y mod 2=0)	(x mod 2=0) xor (y mod 2=0)

Приклади складання умов з логічною операцією not

Ствердження	мова без not	мова з not
Число X не належіть інтервалу [-2, 5]	(x<-2)or(x>5)	Not((x>=-2)and(x<=5))
Ціле число X не ділиться на 3	X mod 3<>0	Not (x mod 3=0)

Завдання:

За допомогою будь-якої мови програмування, реалізувати наступні задачі:

1. Створити "калькулятор бінарних операцій" для числових значень.
2. Створити програму, яка буде перевіряти істинність та хибність логічних виразів.

Лабораторна робота № 4

Тема: *Розширений алгоритм Евкліда*

Загальні відомості

Алгоритм Евкліда (також називається евклідів алгоритм) — ефективний метод обчислення найбільшого спільного дільника (НСД). Названий на честь грецького математика Евкліда, котрий описав його в книгах VII та X Начал.

Алгоритм Евкліда ефективно обчислює НСД великих чисел, оскільки виконує операцій не більше, ніж вп'ятеро більше кількості цифр меншого числа (в десятковій системі).

Для визначення НСД ми використовуємо дві змінні, r1 і r2, щоб запам'ятовувати значення які змінюються протягом всього процесу. Вони мають початкове значення a і b. На кожному кроці ми обчислюємо залишок від ділення r1 на r2 і зберігаємо результат у вигляді змінної r. Потім замінюємо r1, на r2 і r2 на r і продовжуємо кроки, поки r не стане рівним 0. У цей момент процес зупиняється і НСД (a, b) дорівнює r1.

Приклад 1.

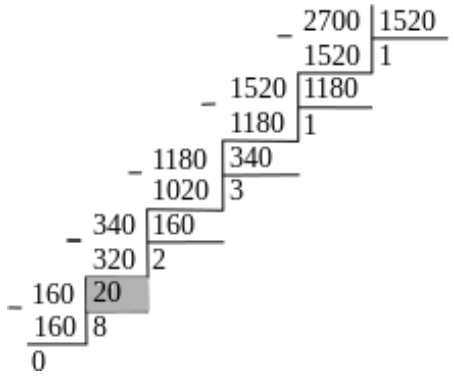
Потрібно знайти найбільший спільний дільник 2740 і 1760.

Рішення

Застосуємо вищезгадану процедуру, використовуючи таблицю. Ми присвоюємо початкове значення r1 2740 і r2 значення 1760. У таблиці також показані значення q на кожному кроці. Ми маємо НСД (2740, 1760) = 20.

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

У випадку невеликих чисел, можна використати послідовне ділення у стовпчик. Наприклад, нам потрібно знайти НСД (2700, 1520):



Дано два цілих числа a і b. Нам часто треба знайти інші два цілих числа, s і r, такі, які $s \times a + t \times b = НОД(a,b)$

Розширений алгоритм Евкліда може обчислити НОД (a, b) і в той же самий час обчислити значення s і t.

Розширений алгоритм Евкліда використовує ті ж самі кроки, що і простий алгоритм Евкліда. Проте в кожному кроці ми застосовуємо три групи обчислень замість однієї. Алгоритм використовує три набори змінних: r, s і t.

На кожному кроці змінні r1, r2 і r використовуються так само, як в алгоритмі Евкліда. Змінним r1 і r2 присвоюються початкові значення a і b відповідно. Змінним s1 і s2 присвоюються початкові значення 1 і 0 відповідно. Змінним t1 і t2 присвоюються початкові значення 0 і 1, відповідно. Обчислення r, s і t однакові,але з однією відмінністю. Хоча r - залишок від ділення r1 на r2, такої відповідності в інших двох групах обчислень немає. Є тільки одна приватна, q, яке обчислюється як r1/r2 і використовується для інших двох обчислень.

Приклад 2

Дано a = 161 і b = 28, треба знайти НСД (a, b) і значення s і t.

Рішення

$r = r1 - q \times r2$, $s = s2 – q \times s2$, $t = t1 - q \times t2$

q	**r_1**	**r_2**	**R**	**s_1**	**s_2**	**s**	**t_1**	**t_2**	**t**
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

Для відображення алгоритму ми використовуємо таку таблицю:

Ми отримуємо НСД (161, 28) = 7, s = -1 і t = 6. Відповіді можуть бути перевірені, як це показано нижче.

$(–1) \times 161 + 6 \times 28 = 7$

Завдання:

За допомогою будь-якої мови програмування, реалізувати наступні задачі:

1. Знайти НСД чисел по алгоритму Евкліда:

1) (589, 343);

2) (987, 610);

3) (2584, 377);

4) (9751, 6931);

5) (9665, 2180);

6) (10946, 2584).

2. Нехай d = (a, b). Знайти цілі числа x, y такі, що ax + by = d, де

1) a = 70, b = 33;

2) a = 60, b = 91;

3) a = 571, b = 359;

4) a = 13, b = 17;

5) a = 144, b = 233;

6) a = 1263, b = 204;

7) a = 504, b = 726.

Лабораторна робота № 5

Тема: *Алгоритм Гордона для генерації сильних простих чисел*

Загальні відомості

Ефективна генерація параметрів відкритих ключів є необхідною умовою для криптосистем з відкритим ключем. Наприклад, для використання протоколу Діффі-Хеллмана (Diffie-Hellman) і його варіацій необхідно згенерувати просте число p для завдання кінцевого поля Z_p. Іншим прикладом є генерація простих чисел p і q для отримання модуля $n = p \times q$ у криптоалгоритмі RSA. У цьому випадку прості числа повинні бути великих розмірів і випадковими в тому сенсі, що ймовірність вибору будь-якого конкретного простого числа повинна бути досить малою для того, щоб не дати зловмисникові можливість використовувати цю ймовірність для оптимізації стратегії пошуку. Іноді до простих чисел можуть пред'являтися додаткові вимоги для того, щоб вони були стійкі відносно будь-яких спеціалізованих криптоатак.

Криптосистема RSA використовує модуль виду $n = p \times q$, де p і q - різні прості числа. Прості числа p і q повинні бути значного розміру, щоб факторизация їх добутку була достатньо обчислювально складною (її неможливо було виконати за прийнятний час, користуючись сучасними обчислювальними можливостями). На практиці, прості числа повинні бути заданої довжини для того, щоб відповідати специфікаціям систем. Створення криптосистеми RSA призвело до появи деяких додаткових обмежень на вибір p і q, які необхідні для впевненості, що підсумкова система RSA захищена від атак криптоаналитика, і було введено поняття сильних простих чисел. Проте, доведено, що сильні прості числа забезпечують захист не набагато більше, ніж просто випадково вибрані прості числа.

Просте число p називається сильним простим, якщо існують цілі числа r, s, t, що задовільняють такі умови:

1. $p + 1$ має великий простий дільник s;

2. $p-1$ має великий простий дільник r;

3. $r-1$ має великий простий дільник t.

У цьому визначенні величина дільників залежить від виду атаки, від якої необхідно захиститися.

Алгоритм Гордона (Gordon's algorithm) для генерації сильного простого числа:

Вихід: сильне просте число p.

1. Згенерувати два великих простих числа s і t однієї довжини;

2. Обрати число i_0. Знайти перше просте число з послідовності $2It+1$, де $I = i_0$, $I = I + 1$. Позначити це число $r = 2It + 1$;

3. Обчислити $p_0 = 2(s^{r-2} \bmod r)s - 1$;

4. Обрати ціле число j_0. Знайти перше просте число з послідовності $p_0 + 2jrs$, де $j = j_0$, $j = j + 1$. Позначити це число $p = p_0 + 2jrs$;

5. Повернути *p*.

Завдання:

За допомогою будь-якої мови програмування, реалізувати алгоритм Гордона для генерації сильних простих чисел. Обравши за початкові параметри алгоритма: i_0 - порядковий номер в журналі групи, j_0 - остання цифра номера залікової книжки, згенерувати сильне просте число.

Лабораторна робота № 6

Тема: *Шифр "одноразовий блокнот"*

Загальні відомості

Подання тексту у цифровій формі. У період класичної криптографії як правило не виникали потреби записувати відкритий текст та криптотекст якось інакше, ніж у звичайній абетці. Завдяки цьому криптограф-практик не потребував для роботи нічого, крім письмового приладдя свого часу, чого було достатньо і для шифрування, і для пересилання повідомлення. Але як тільки ми забажаємо скористатися модерними засобами зв'язку для передачі повідомлення, або доручити шифрування комп'ютерові, то виявимо, що у технічному відношенні традиційний текст не є найзручнішою формою для перетворення та передачі інформації.

З цього погляду вигіднішим є подання інформації у *цифровій формі.* Ідея є зовсім простою — кожен символ тексту заміняємо його номером у алфавіті. Нагадаємо, що нумерацію ми домовились починати з 0. Для прикладу, слово банан буде подане як 01 00 17 00 17. Кожна літера представлена своїм номером, записаним двома цифрами, перша з яких може бути нулем. При потребі в алфавіт можна включити окрім букв також знаки пунктуації, пропуск, цифри тощо.

Номери букв ми можемо записувати не в десятковій системі числення, а у двійковій. Для того ж слова банан матимемо запис 000001000000010001000000010001, де кожний блок із шести цифр є номером відповідної букви у двійковому записі. Таку форму подання тексту називатимемо *двійковою.* **Ми потребуємо не менше шести цифр, бо п'ятьма можна записати щонайбільше 32 числа, в той час як українська абетка налічує 33 літери.**

Таким чином, довільний текст можна записати у двійковій формі, використовуючи всього лише два символи — 0 та 1 (один біт). Будь-яку послідовність бітів називають *двійковим словом.*

Шифр одноразового блокноту був винайдений у 1917 році Гілбертом Вернамом. Він використовує операцію поступового додавання бітів за модулем

2, яку ми розглянемо перед тим як описати сам шифр. Операція позначається символом ⊕ і задається так:

$$0\oplus0 = 0, \quad 0\oplus1=1, \quad 1\oplus0=1, \quad 1\oplus1 = 0.$$

Поширимо цю операцію на двійкові слова однакової довжина домовившись, що додавання таких слів відбувається побітово (не слід плутати цю операцію із звичайним додаванням чисел у двійковій системі. Зокрема, при побітовому додаванні не виникає ніяких переносів у наступний розряд). Наприклад,

$$00000100000001000100000010001$$
$$00110011010111001101100010101 1$$

Для двійкових слів X *і* Y однакової довжини результат їх побітового додавання позначатимемо як $X\oplus Y$. Легко перевірити рівності $X\oplus Y = Y \oplus X$ та $(X \oplus Y) \oplus Z - X \oplus(Y\oplus Z)$, *що* справджуються для будь-яких двійкових слів X, Y і Z однакової довжини. Для фіксованого k позначимо через 0 двійкове слово 000... 00, що складається із k нулів. Очевидно, що для будь-якого двійкового слова X довжини k виконуються рівності $X\oplus 0 = XiX\oplus X = 0$.

Перейдемо тепер до опису шифру. Перед шифруванням повідомлення M записують у двійковій формі. Ключем K служить довільне двійкове слово однакової з M довжини. Криптотекст C отримують побітовим додаванням повідомлення і ключа, тобто $C \quad M\oplus K$.

Для прикладу, нехай ми хочемо зашифрувати слово банан. У попередньому пункті ми подали його у двійковій формі:

M - 00000100000001000100000010001. В якості ключа оберемо

K = 00110111010110001001100011101 0.

Сумування цих двох двійкових послідовностей вже проведене нами вище. Отож маємо криптотекст

C – 00110011010111001101100010101 1.

Дешифрування у шифрі одноразового блокноту збігається із шифруванням — щоб отримати вихідне повідомлення M , слід додати до криптотексту С той же ключ K. Це легко обгрунтувати: оскільки $C - M \oplus K$, то

$$C \oplus K - (M \oplus K) \oplus K - M\oplus (K \oplus K) - M\oplus 0 - M.$$

Шифр одноразового блокноту є *абсолютно надійним* або, як ще кажуть, *надійним у теоретико-інформаційному сенсі.* Якщо суперник не знає ключа K, то з підслуханого криптотексту C він *зовсім нічого* не може довідатись про повідомлення M. Справді, двійкове слово C могло би бути криптотекстом для *будь-якого* повідомлення M' , якби шифрування здійснювалось з деяким іншим ключем K' , а саме $K' — M' \oplus$ С, в той час як

для суперника всі ключі однаково вірогідні. Наприклад, за допомогою деякого ключа K ми щойно перетворили повідомлення банан у криптотекст

C = 001100110101110011011000101011.

Але такий же криптотекст ми отримали б, якби зашифрували повідомлення **груша** з ключем

K' – 001111100001100100000100101011.

Назва шифру походить від того, що агент, який здійснював шифрування вручну, отримував свої копії ключів записаними у блокноті. Як тільки ключ використовувався, сторінка з ним знищувалась. Зрозуміло, що шифр просто реалізується і технічними засобами. Кажуть, що саме шифр одноразового блокноту використовувався для захисту від підслуховування встановленої під час холодної війни гарячої лінії зв'язку між Москвою і Вашингтоном.

Однак обмеженість сфери застосувань шифру очевидна, оскільки він вимагає ключа такої ж довжини як саме повідомлення. З цією обставиною пов'язані дві проблеми. Перша полягає в генеруванні довгої послідовності випадкових бітів. Другою проблемою є необхідність у надійному каналі для регулярного обміну довгим ключем (на зразок дипломатичної пошти). У більшості ситуацій такого каналу або взагалі нема, або ж він не є достатньо швидким.

На завершення зауважимо, що достеменно такими ж перевагами та недоліками володітиме шифр Віженера, якщо у ньому брати ключ тої ж довжини, що й повідомлення.

Завдання:

а) Подати у двійковій формі слова:

1. ***ананас*** та ***яблуко.***

2. груша та ***слива;***

3. пшениця та ***овес;***

4. малина та ***полуниця;***

5. горіх та ***лимон;***

6. земля та ***небо;***

7. море та ***океан;***

8. річка та ***озеро.***

б) Зашифрувати у якості повідомлення перше слово з пари, використавши шифр одноразового блокноту з ключем

110000 011110 010100 110010 010110 011110.

в) Визначити з яким ключем той же результат дало б шифрування повідомлення другого слова з пари?

Лабораторна робота № 7

Тема: *Алгоритм та програма частотного аналізу текстів*

Загальні відомості

Як вам відомо з попередніх лабораторних робіт, шифр заміни над n-символьним алфавітом має ***n!*** ключів. Для значень $n = 26,33$ (латинський та український алфавіти) це число є дуже великим. Для його оцінки можна скористатися варіантом формули Стірлінга, звідки для n=26 отримуємо $n > 1026$. Число справді велике — нагадаємо, що наша планета існує лише 109 років, а наступний льодовиковий період очікується через 14000 років, тобто 4,41504 • 1011 секунд. Це співставлення переконливо засвідчує безперспективність брутальної атаки на шифр заміни, однак цього недостатньо аби стверджувати, що він є надійним. Виявляється, успішний криптоаналіз можливий за допомогою ***частотного методу.***

Частота символу у тексті дорівнює кількості його входжень у цей текст, поділеній на загальну кількість букв у тексті. Наприклад, частота букви а у тексті **купила мама коника** дорівнює 4/18 = 2/9, а частота пропуску між словами у цьому ж тексті дорівнює 2/18 = 1/9. Для кожної мови справджується такий емпіричний факт:

У досить довгих текстах кожна буква зустрічається із приблизно однаковою частотою, залежною від самої букви та незалежною від конкретного тексту.

Частотним методом можна здійснити дешифрування, навіть не знаючи ключа. Для цього обчислюють частоти кожного символу в криптотексті і порівнюють отримані результати з табличкою частот для мови, якою написане повідомлення. Не слід сподіватися, що таким чином можна буде однозначно встановити ключ, але перебір це дозволить скоротити радикально. Наприклад, якщо при шифруванні не ігноруються пропуски між словами, то найпоширеніший символ у криптотексті поза сумнівом відповідає саме пропуску. А відтак стає відомою сукупність символів, що відповідають словам з одної букви (в українській мові це а,б,в,є,ж,з,і,й,о,у,я) та словам з двох букв (це,не,на,до та інші), що дозволяє ці символи розпізнати ціною справді невеликого перебору. Свою роль при частотному аналізі відіграє та обставина, що кожна мова володіє властивістю *надлишковості,* тобто текст можна поновити навіть коли частина його букв невідома.

Миттєвою є користь від частотного аналізу при розкритті шифру зсуву. Проілюструємо загальну ідею таким прикладом. Нехай нам належить розшифрувати криптотекст ***пцпспофнпмплпибгпефрпттмбвмєоп,*** який був отриманий шифром зсуву, причому пропуски та розділові знаки ігнорувались. Підраховуємо частоти і зауважуємо, що найбільша, а саме 9/29, припадає на літеру п. Природньо припустити, що у відкритому тексті їй відповідає найпоширеніша в українській мові літера о. Це означало б, що довжина зсуву дорівнює 1. Виконуємо обернений зсув, тобто на одну позицію вліво, і справді отримуємо змістовне повідомлення, що ***охоронумолокозаводупослаблено***.

Гомофонний шифр заміни був винайдений великим німецьким математиком

Карлом Фрідріхом Гаусом. Цей шифр грунтується на ідеї, яка робить підрахунок частот символів безперспективним. Кожна буква відкритого тексту замінюється не єдиним символом як у шифрі простої заміни, а будь-яким символом із декількох можливих. Наприклад, замість а може здійснюватись підстановка будь-якого із чисел 10,17,23,46,55, а замість б — будь-якого із 12,71. Головне, щоб замість різних букв завжди підставлялись різні символи — ця вимога забезпечує можливість дешифрування. Вибір одного з можливих варіантів щоразу робиться випадково. Якщо кількість варіантів для кожної букви пропорційна її частоті в мові, то всі символи у досить довгому криптотексті зустрічатимуться з приблизно однаковою частотою, що не дозволить пов'язати їх з якимись буквами відкритого тексту. Однак гомофонний шифр піддається ретельнішому і трудомісткішому різновиду частотного аналізу, який окрім частот окремих символів враховує також частоти пар символів. Подібний аналіз дозволяє ламати ще один клас шифрів заміни, про що йдеться у наступному пункті.

Частотний аналіз може бути корисним і в інших ситуаціях. Наприклад, він дозволяє комп'ютерові без участі людини відрізнити осмислений текст від хаотичного набору символів. Завдяки цьому на машину можна перекласти здійснення брутальної атаки, тобто повного перебору ключів.

Завдання:

1. Порахувати частоти всіх символів у тексті, узявши текст довжиною не менше 3 сторіок.

2. В потоці зашифрованих донесень від інформатора з Марийського палацу домінує буква н. Припустивши, що використовується шифр зсуву і пропуски між словами ігноруються, знайти за допомогою частотного аналізу ключ і розшифрувати повідомлення:

опдрснкнмярстомзйнлопнвнкнчдмннкдкшйяспдсшнвн

VI СЕМЕСТР

Лабораторна робота № 1

Тема: *СИМЕТРИЧНІ КРИПТОПЕРЕТВОРЕННЯ*

Загальні відомості

Задача 1.

Зашифрувати повідомлення «Фамилия имя» методом простої підстановки, алфавіт російський. Визначити розмірність простору ключів n_k, ентропію $H(k)$, безпечний час $t_б$ та відстань єдності l_0, якщо потужність криптоаналітичної системи $\gamma = 10^7$ вар/с. Розшифрувати повідомлення і пересвідчитися в однозначності процедури зашифрування.

Ключ:

Вх:	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
Вих:	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ
Вх:	ъ	ы	ь	э	ю	я	_																		
Вих:	ы	ь	э	ю	я	_	а																		

Розв'язок задачі:

Використовуючи ключ підстановки, зашифруємо повідомлення «Фамилия имя». Зашифроване повідомлення має вигляд:

С = «хбнкмк_акн_».

Знаходимо розмір простору ключів:

$$N_{кл} = m! = 32! = 2{,}6 \cdot 10^{35} \approx 2^{117},$$

де m – основа алфавіту.

Знаходимо ентропію джерела ключів:

$$H(k) = -\sum P_i \log_2 P_i = N_{кл} \frac{1}{N} \log_2 N_{кл} = \log_2 2^{117} = 117.$$

Далі визначаємо безпечний час:

$$t_б = \frac{N_{кл}}{\gamma k} P_K,$$

де P_K – імовірність, з якою має бути здійснений криптоаналіз.

У результаті маємо:

$$t_б = \frac{2{,}6 \cdot 10^{35}}{10^7 \cdot 3{,}1 \cdot 10^7} = 10^{21} \text{ років}.$$

Отже шифр може володіти дуже високою стійкістю, але у зв'язку з тим, що природні мови володіють значною надмірністю, пов'язаною з нерівномірністю появи букв у мові і залежністю букв між собою, існує ефективний метод частотного криптоаналізу. Суть аналізу: набирається об'єм не менше за 2000 символів, далі будується гістограма частот появи символів у криптограмі.

Відстань єдності для шифру:

$$l = \frac{H(k)}{r} = \frac{117}{0{,}5} = 234 \approx 47 \text{ літер}.$$

Задача 2.

Зашифрувати повідомлення (див. задача 1), використовуючи шифр Віжінера. Ключ: “СЛУЧАЙНОСТЬ”. Тобто ключ має довжину 11 символів.

Потужність криптоаналітичної системи $\gamma = 10^{13}$ вар/с.

Розв'язок задачі:

Проаналізуємо букви алфавіту, згідно з табл. 1.1.

Таблиця 1.1

а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	р	с
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	–		
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		

Запишемо наше повідомлення “Фамилия имя” у цифровому вигляді:

$M_i = 19, 0, 11, 8, 10, 8, 30, 31, 8, 11, 30.$

Знаходимо Γ_i, використовуючи ключ “СЛУЧАЙНОСТЬ”.

$$\Gamma_i = \varphi(K_j), \quad \Gamma_i = 16, 10, 18, 22, 0, 8, 12, 13, 16, 17, 27.$$

Визначаємо символи криптограми C_i:

$C_i = (M_i + \Gamma_i) \bmod 32$ згідно з табл. 1.2.

Таблиця 1.2

M_i	19	0	11	8	10	8	30	31	8	11	30
Γ_i	16	10	18	22	0	8	12	13	16	17	27
C_i	3	10	29	30	10	16	10	12	24	28	25

$C_i = 3, 10, 29, 30, 10, 16, 10, 12, 24, 28, 25.$

Замінемо символи цифрової криптограми буквами згідно з таблицею 1.1. В результаті маємо:

$C_i =$ “ГЛЮЯЛСЛНЩЭЪ”

Розшифрування здійснюємо в зворотному порядку згідно з формулою $M_i = (C_i - \Gamma_i) \bmod 32$.

Знаходимо розмір простору ключів:

$$N_{кл} = m^{l_K},$$

де m – основа алфавіту;

l_K – розмір ключа (довжина);

$N_{кл} = (32)^{11} = 2^{55} \approx 3{,}6 \cdot 10^{16}$.

Знаходимо ентропію джерела ключів за формулою:

$$H(K) = -\sum_{i=1}^{n_m} \frac{1}{m} \log_2 \frac{1}{m} = \log_2 2^{55} = 55.$$

Далі визначаємо безпечний час:

$$t_б = \frac{N_{кл}}{\gamma k} P_k ; \quad при \ \ P_k = 1 .$$

В результаті маємо

$$t_б = (3{,}6 \cdot 10^{16})/10^{13} = 3{,}6 \cdot 10^{3} = 58 \text{ хв. } 20 \text{ с.}$$

$l_0 - \frac{H(K)}{d} = \frac{55}{0{,}5}$ – 110 бітів або 22 п'ятибітних символів (букв російського алфавіту).

Задача 3.

Зашифруйте повідомлення «2,7,9, D, A, 3,8, D, C, 1, A» яке подано в шістнадцятковій системі числення, потоковим методом, використовуючи ключ K={7,9, A, 3,8,4, C, B, 1, 5, 4}. Знайдіть повну безліч ключів і безпечний час такої криптосистеми, якщо символи ключа з'являються рівноймовірнісно і незалежно.

Пронумеруємо символи повідомлення згідно з таблицею 1.3 (m=16):

Таблиця 1.3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Розв'язок задачі:

$C_i = (M_i + \Gamma_i)(\text{mod}16)$;

$M_i = (C_i - \Gamma_i)(\text{mod}16)$.

Записуємо наше повідомлення:

M_i = 2, 7, 9, 10, 3, 8, 13, 12, 1, 10.

Знаходимо Γ_i, перетворивши ключ у цифрову послідовність

Γ_i = 7, 9, 10, 3, 8, 4, 12, 11, 1, 5, 4.

Визначаємо символи криптограми Сі:

$C_i = (M_i + \Gamma_i)$ mod 16.

В результаті маємо

C_i = 9, 0, 3, 0, 2, 7. 4, 8, 13, 6, 14 = 9, 0, 3, 0, 2, 7. 4, 8, D, 6, E.

Знаходимо розмір простору ключів:

$N_k = m^{Lk} = (16)^{11} = 2^{44} = 10^{13{,}2}$.

Знаходимо ентропію мови повідомлення за формулою:

$H_0 = \log m = \log 2^4 = 4$ біти, де $H_0(M)$ – ентропія мови, де всі символи рівно ймовірні і незалежні.

Далі визначаємо безпечний час:

$t_б = \frac{N_{кл}}{\gamma k} P_k ; \quad при \ \ P_k = 1$,

$$t_б = \frac{10^{10.6}}{10^7 \cdot 3{,}1 \cdot 10^7} = 5 * 10^{-2} \text{ років.}$$

Задача 4.

Визначити основні показники обчислювально стійких систем, реалізованих з використанням: ГОСТ 28147-89, DES та RIJNDAEL.

Розв'язок задачі:

У криптоалгоритмах, що розглядаються, довжини ключів та число ключів мають такі значення.

ГОСТ-28147-89	$l_k =$ 256 біт	$n_k = 2^{256}$
DES	$l_k =$ 56 біт	$n_k = 2^{56}$
RIJNDAEL	$l_k =$ 128 біт	$n_k = 2^{128}$

Основними показниками, за якими може бути оцінена стійкість криптоалгоритмів, є такі: $t_б$ (безпечний час), $N_{кл}$ (кількість ключів), $H(K)$ (ентропія джерела ключів), l_0 (відстань єдності).

$N_{кл}(\Gamma) = 2^{256}$ ключів,

$N_{кл}(\text{D}) = 2^{56}$ ключів,

$N_{кл}(\text{R}) = 2^{128}$ ключів.

Вважатимемо, що ключі з'являються рівноймовірно і незалежно, визначимо ентропію джерела ключів $H(K)$:

$$H(K) = -\sum_{i=1}^{N_k} P(K_i) \log P(K_i) = -\sum_{i=1}^{N_k} \frac{1}{N_{кл}} \log N_{кл}^{-1} .$$

$H(K) = 256$ біт

$H(K) = 56$ біт

$H(K) = 128$ біт

Якщо

$$P(K_i) = \frac{1}{N_{кл}} = N_{кл}^{-1},$$

тоді

$$t_б^{DES} = \frac{N_{кл}^{DES}}{\gamma \cdot K} \cdot 1 = \frac{2^{56}}{10^9 \text{ операцій} / c \cdot 3 \cdot 10^7} = \frac{10^{16.8}}{3 \cdot 10^{16}} \approx 2{,}1 \text{ (роки)},$$

де γ – продуктивність криптоаналітичної системи;

K – кількість секунд у році.

Таким чином, для здійснення криптоаналізу методом грубої сили необхідно приблизно 2 роки.

Знайдемо $t_б$ для випадку диференціального або лінійного криптоаналізу, для яких $N_{вар} = 2^{47}$. В результаті маємо, що

$$t_б = \frac{2^{47}}{\gamma \cdot K} = \frac{10^{14.1}}{3 \cdot 10^{16}} = 3 \cdot 10^{-3} \text{ (років)} \approx 1{,}2 \text{ дні.}$$

Вважатимемо, що в системі передаються повідомлення англійським текстом, надмірність якого 0,45.

Відстань єдності:

$$l_0^{DES} = \frac{H(K)}{d \cdot \log m} = \frac{H(K)}{d \cdot \log 2} = \frac{56}{0{,}45 \cdot 1} = 124 \text{ (біт).}$$

Зі значення l_0 робимо висновок, що для того, щоб успішно здійснити криптоаналіз, треба перехопити не менш ніж 124 біт символів криптограми, тоді, затративши $t_б$, ми можемо успішно розв'язати задачу криптоаналізу.

Визначте основні параметри стійкості ГОСТ 28147-89 та RIJNDAEL самостійно.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання наступних задач. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

1. Зашифрувати повідомлення «(Своє повне ім'я і прізвище)» методом простої підстановки, алфавіт український або російський. Визначити розмірність простору ключів n_k, $H(k)$, $t_б$ і l_0, якщо потужність криптоаналітичної системи у = 10 вар/с. Розшифруйте повідомлення і перевірте однозначність процедури зашифрування-розшифрування.

2. Зашифрувати повідомлення з попереднього завдання, використовуючи шифр Віжінера. Визначити розмірність простору ключів n_k, $H(k)$, $t_б$ і l_0, якщо

потужність криптоаналітичної системи у = 10 вар/с. Розшифруйте повідомлення і перевірте однозначність процедури зашифрування-розшифрування.
Ключ: "небосхил", $^{l}k = {}^{8}$.

3. Зашифруйте повідомлення «2,7,9, D, A, 3, 8, D, C, 1, A», яке подано в шістнадцятковій системі числення, потоковим методом, використовуючи ключ

{0110,1110,1010,1000,1011,1001,1111,0011,1100,1000,1101}.

Знайдіть повну множину ключів і безпечний час такої криптосистеми, якщо символи ключа з'являються рівноймовірно і незалежно, у = 10 вар/с. Розшифруйте повідомлення.

Контрольні запитання:

1. В чому сутність алгоритму потокового шифрування.
2. Якими властивостями володіють безумовно стійкі криптоалгоритми?
3. Дайте визначення обчислювально стійких систем. Що розуміють під

відстанню єдності для безумовно і обчислювально стійких криптосистем?

4. Дайте визначення надмірності.

5. У чому відмінність безумовно стійких і обчислювально стійких криптосистем?

6. Що розуміють під структурною скритністю?

7. Які основні показники оцінки криптостійкості?

8. Що собою являють афінні перетворення?

9. Дати визначення блокового шифру.

10. В чому сутність складового шифру.

11. В чому сутність шифру монопідстановки.

12. Визначіть кількість ключів, які можуть бути використані в шифрі монопідстановки.

13. Який основний недолік шифру монопідстановки, якщо він використовується для зашифрування повідомлення українською мовою.

14. Дайте визначення підстановки та назвіть її властивості.

Лабораторна робота № 2

Тема: *НЕСИМЕТРИЧНІ КРИПТОПЕРЕТВОРЕННЯ: АЛГОРИТМ RSA*

Загальні відомості

Задача 1.

Нехай P=11, Q=7, E_k=37. Побудуйте ключову пару (E_k,D_k) для RSA-перетворення.

Розв'язок задачі:

Модуль перетворення має значення

$N = P \cdot Q = 11 \cdot 7 = 77.$

Розраховуємо значення функцій Ойлера

$\varphi(N) = (P\text{-}1)(Q\text{-}1) = 10 \cdot 6 = 60.$

Для знаходження D_k ключа розв'яжемо порівняння

$E_k \cdot D_k = 1(\mathrm{mod}\varphi(N_j))$.

Подамо це порівняння у вигляді (1.58)

$\varphi(N_j) \cdot x + E_k \cdot y = 1$.

Підставивши значення $\varphi(N_j)$ та E_k, маємо

$60 \cdot x + 37 \cdot y = 1$.

Подамо a/b у вигляді ланцюгового дробу

$$\frac{a}{b} = \frac{\varphi(N_j)}{E_k} = \frac{60}{37};$$

60/37=1+23/37; 37/23=1+14/23; 23/14=1+9/14; 14/9=1+5/9; 9/5=1+4/5; 5/4=1+1/4; 4/1=4+0.

Це означає, що $\mu = 6$.

Тоді значення $y = D_k$ можна знайти з виразу

$y = (-1)^{\mu} a_{\mu-1} = (-1)^6 a_5 = a_5$.

Підрахуємо коефіцієнти, a_0, a_1, a_2, a_3, a_4 та a_5.

$a_0 = r_0 = 1;$

$a_1 = r_0 r_1 + 1 = 1 \cdot 1 + 1 = 2;$

$a_2 = a_1 r_2 + a_0 = 2 \cdot 1 + 1 = 3;$

$a_3 = a_2 r_3 + a_1 = 3 \cdot 1 + 2 = 5;$

$a_4 = a_3 r_4 + a_2 = 5 \cdot 1 + 3 = 8;$

$a_5 = a_4 r_5 + a_3 = 8 \cdot 1 + 5 = 13.$

Визначимо ключ розшифрування

$y = D_k = (-1)^6 \cdot 13 = 13;$

$(E_k, D_k) = (37,13)$.

Перевірку здійснюємо підстановкою значень E_k та D_k в основне порівняння $37 \cdot 13 \equiv 1 (\mathrm{mod}\, 60)$.

Таким чином (E_k =37 та D_k= 13) є ключовою парою RSA-перетворення.

Задача 2.

Нехай P=11 і Q=7. Побудувати пару (E_k, D_k) для RSA-перетворення, обґрунтувавши та вибравши один із випадкових ключів.

Розв'язок задачі:

Знаходимо модуль перетворення та значення функції Ойлера φ (N)

Модуль перетворення має значення

$N = P \cdot Q = 11 \cdot 7 = 77.$

Розраховуємо значення функцій Ойлера

$\varphi(N) = (P\text{-}1)(Q\text{-}1) = 10 \cdot 6 = 60.$

Порівняння виду

$E_k \cdot D_k = 1 (\mathrm{mod} \varphi(N_j))$

запишемо у вигляді Діафантового рівняння

$\varphi(N_j) \cdot x + E_k \cdot y = 1$.

Вибравши випадково E_k =17 ключ, взаємопростий з функцією Ойлера, тобто (E_k, φ (N)) = 1, маємо

$60 \cdot x + 17 \cdot y = 1$.

Тепер подамо a/b у вигляді ланцюгового дробу:

$\frac{a}{b} = \frac{\varphi(N_j)}{E_k} = \frac{60}{17};$

60/17=3+9/17; 17/9=1+8/9; 9/8=1+1/8; 8/1=8+0.

Таким чином

$\mu = 3$.

Розраховуємо значення $y - D_k$

$y = (-1)^{\mu} a_{\mu-1} = (-1)^3 a_2 = -a_2$.

Знаходимо рекурентно значення a_2:

$a_0 = r_0 = 3;$

$a_1 = r_0 r_1 + 1 = 3 \cdot 1 + 1 = 4;$

$a_2 = a_1 r_2 + a_0 = 4 \cdot 1 + 3 = 7.$

Далі знаходимо

$y = D_k = (-7) \bmod 60 = 53$.

Отже пара

$(E_k, D_k) = (17,53)$ складає RSA ключову пару.

Перевірка:

Підставивши значення ключів E_k =17та D_k = 53, маємо

$17 \cdot 53 \equiv 1 (\bmod 60)$.

Задача 3.

Нехай P=11 і Q=7. Виберемо E_k=9 як випадковий ключ і побудуємо пару (E_k,D_k) для RSA ключів.

Розв'язок задачі:

Знаходимо модуль RSA перетворення та значення функції Ойлера:

Модуль перетворення має значення

$N = P \cdot Q = 11 \cdot 7 = 77.$

Розраховуємо значення функцій Ойлера

$\varphi(N) = (P\text{-}1)(Q\text{-}1) = 10 \cdot 6 = 60.$

Ключ D_k знайдемо із порівняння

$E_k \cdot D_k = 1 (\bmod \varphi(N_j))$,

далі зведемо вищенаведене порівняння до Діафантового рівняння

$\varphi(N_j) \cdot x + E_k \cdot y = 1$

$60 \cdot x + 9 \cdot y = 1.$

Знайдемо розклад ланцюгового дробу:

$\frac{a}{b} = \frac{\varphi(N_j)}{E_k} = \frac{60}{9};$

60/9=6+6/9; 9/6=1+3/6; 6/3=2+0.

Оскільки НСД $(E, \varphi(N)) \neq 1$, то розв'язку для пари ключів (E_k, D_k) немає.

2.4 Задачі для самостійного розв'язку

Задача 1. Побудувати пару (E_k,D_k) для RSA криптоалгоритму, якщо $P = P_n, Q = Q_n$ (значення P і Q дивись у табл. 2.1).

Таблиця 2.1 – Значення P і Q для задачі 1

n	1	2	3	4	5	6	7	8	9	10
P_n	29	11	11	11	23	7	29	17	19	19
Q_n	11	17	23	13	17	23	7	7	7	11

$n = k \bmod 11$, де k – номер за списком.
Якщо $k = 11$, то $n = (k+1) \bmod 11$.

Задача 2. Розв'язати порівняння $E_k \cdot D_k \equiv 1 \bmod(\varphi(N))$, якщо $N = N_n$ (значення N дивись у табл. 2.2).

Таблиця 2.2 – Значення P і Q для задачі 2

n	1	2	3	4	5	6	7	8	9	10
N_n	187	203	297	189	351	209	133	391	143	145

$n = k \bmod 11$, де k – номер за списком.
Якщо $k = 11$, то $n = (k+1) \bmod 11$.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Дайте визначення асиметричного криптоперетворення.
2. Як взаємопов'язані ключі в асиметричній RSA криптосистемі?
3. Поясніть основні співвідношення зашифрування та розшифрування в RSA системі.
4. Які вимоги висуваються до модуля криптоперетворення в RSA системі?
5. Які вимоги висуваються до простих чисел, що входять співмножниками в модуль преретворення?
6. Які параметри RSA перетворення є конфіденційними, а які відкритими, та чому ?
7. Якими властивостями володіють сильні прості числа?
8. Назвіть основні методи криптоаналізу RSA криптоперетворень.
9. Порівняйте складність різних методів RSA криптоперетворень.
10. Який метод RSA криптоаналізу має найменшу складність?
11. Які ключі є ключами-близнюками? Чи можна їх використовувати?

12. Назвіть основні переваги та недоліки RSA криптоперетворення.
13. Які вимоги висуваються до довжин простих чисел модулів перетвореня?
14. Обґрунтуйте методику RSA криптоаналізу, що забезпечує найменшу складність.
15. Обґрунтуйте слабкі та сильні сторони RSA криптоперетворення. Яку перспективу має RSA?

Лабораторна робота № 3

Тема: *КРИПТОГРАФІЯ НА ЕЛІПТИЧНИХ КРИВИХ*

Загальні відомості

Задача 1.

Еліптична крива має вигляд $y^2 = (x^3 + x + 1) \bmod 23$, причому $a = 1$, $b = 1$, $P = 23$.

1. Перевірити, що точки P_1=(3,10), P_2=(9,7) належать кривій та знайти $P_1 + P_2 = P_3$.

Підставивши значення P_1 та P_2 у рівняння еліптичної кривої бачимо, що вони належать кривій (ліва та права частини порівнюються).

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\bmod P) = \frac{7-10}{9-3} (\bmod 23) = -\frac{3}{6} (\bmod 23) = -\frac{1}{2} (\bmod 23) = \frac{22}{2} (\bmod 23) = 11.$$

Знаходимо

$x_3 = (121 - 3 - 9) m23 = 109 (m23) = 17$.

$y_3 = (11(3-17) - 10) m23 = (-11 \cdot 14 - 10) m23 = -164 (\bmod 23) = 20$.

Отже P_1+P_2=(3,10)+(9,7)=(17,20).

Задача 2.

Скласти точки P_1 і P_2. P_1=(12,19), P_2=(5,4). Якщо еліптична крива має вигляд $y^2 = x^3 + x + 1 (\bmod 23)$, тобто $a = 1$, $b = 1$, $P = 23$.

Розв'язок задачі:

$$\lambda = \frac{4-19}{5-12} = \frac{-15}{-7} (\bmod 23) = \frac{15}{7} (\bmod 23).$$

Знаходимо в полі $G(23)$ обернений елемент Z, розв'язавши порівняння $7 \cdot Z \equiv 1 \pmod{23}$.

Це порівняння має розв'язок при $Z = 10$, тому

$\lambda = 15 \cdot 10 \pmod{23} = 12$.

$x_3 = 12^2 - 12 - 5 = 144 - 12 - 5 \pmod{23} = 127 \bmod 23 = 12 \bmod 23$.

$y_3 = 12 \cdot (12 - 12) - 19 (\bmod 23) = 4 \bmod 23$.

Таким чином:

$P_1 + P_2 = (x_1, y_1) + (x_2, y_2) = P_3 = (x_3, y_3) = (12, 4)$.

P_3=(12,4).

Задача 3.

Знайти точку, яка дорівнює $3 \cdot P_1$. $P_1 = (5,4)$. Еліптична крива має вигляд $y^2 = x^3 + x + 1 (\mathrm{mod}\, 23)$, тобто a=1, b=1.

Розв'язок задачі:

Спочатку подвоїмо точку P_1, тобто знайдемо $P_2 = 2 \cdot P_1$.

Знайдемо λ, використовуючи (3.13)

$$\lambda = \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} (\mathrm{mod}\, 23) = \frac{3 \cdot 25 + 1}{2 \cdot 4} (\mathrm{mod}\, 23) = \frac{76}{2 \cdot 4} (\mathrm{mod}\, 23) = \frac{19}{2} (\mathrm{mod}\, 23).$$

Знайдемо зворотний елемент

$2 \cdot Z \equiv 1 (\mathrm{mod}\, 23)$; $Z = 12$.

Отже $\lambda = 19 \cdot 12 (\mathrm{mod}\, 23) = 21$.

Використовуючи (3.10) – (3.11) та враховуючи, що x_1=x_2, знайдемо координати x_2 і y_2 точки $P_2(x_2,y_2)$

$x_2 = (21^2 - 2 \cdot 5) \mathrm{mod}\, 23 = (441 - 10) \mathrm{mod}\, 23 = 431 \mathrm{mod}\, 23 = 17$;

$y_2 = (21(5 - 17) - 4) \mathrm{mod}\, 23 = -256 \mathrm{mod}\, 23 = 20$.

Отже: $P_2 = 2 \cdot P_1 = 2 \cdot (5,4) = (17,20)$.

Далі знайдемо суму точок P_1+P_2=P_1+2P_1=P_3=(x_3, y_3).

$P_3 = 3 \cdot P_1 = P_1 + 2P_1 = (5,4) + (17,20)$.

Знайдемо λ, використовуючи (1.87)

$$\lambda = \frac{20 - 4}{17 - 5} = \frac{16}{12} (\mathrm{mod} 23) = \frac{4}{3} (\mathrm{mod} 23).$$

Знайдемо зворотний елемент

$3 \cdot Z \equiv 1 (\mathrm{mod}\, 23)$; $Z = 8$.

Отже $\lambda = 4 \cdot 8 (\mathrm{mod}\, 23) = 32 \mathrm{mod}\, 23 = 9$.

Використовуючи (3.10) і (3.11), знайдемо x_3 та y_3

$x_3 = (9^2 - 5 - 17) \mathrm{mod} 23 = (81 - 22) \mathrm{mod} 23 = 13$;

$y_3 = (9(5 - 13) - 4) \mathrm{mod}\, 23 = (9 \cdot (-8) - 4) \mathrm{mod}\, 23 = -76 \mathrm{mod}\, 23 = 16$.

Отже: $3P_1$=3(5,4)=(13,16).

Задача 4.

Нехай є ЕК з рівнянням: $y^2 = (x^3 + x + 1) \mathrm{mod}\, 23$, a=1, b=1, p=23.

Перевірити, чи належать такі точки ЕК:

(1,7), (1,16).

Розв'язок задачі:

$7^2 = (1+1+1) \bmod 23$, точка (1,7) належить ЕК;

$16^2 \equiv (1+1+1) \bmod 23$, точка (1,16) належить ЕК.

Задача 5.

Знайти порядок базової точки (17,20). Якщо рівняння еліптичної кривої має вигляд $y^2 = x^3 + x + 1 (\bmod 23)$.

Розв'язок задачі:

Для визначення порядку точки (17,20) на еліптичній кривій

$$y^2 = x^3 + x + 1 (\bmod 23)$$

знайдемо таке значення d, щоб $d^*(17,20)(\bmod 23) = 0$.

Для цього підставимо в останнє порівняння $d = 1, 2, 3, \ldots k$, та визначимо значення скалярного добутку.

$$\lambda = \frac{3 \cdot 289 + 1}{40} (\bmod 23) = \frac{868}{40} (\bmod 23) = 1;$$

$$x_3 = (\lambda^2 - 2x_1) \bmod p = (1 - 34) \bmod 23 = 13;$$

$$y_3 = (\lambda(x_1 - x_3) - y) \bmod p = (1 \cdot 4 - 20) \bmod 23 = 7.$$

При $d = 2$ маємо

$Q_1 = (13,7)$.

При $d = 3$

$P_1 + Q_1 = Q_2$;

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\bmod p) = \frac{7 - 20}{13 - 17} \bmod 23 = \frac{-13}{-4} \bmod 23 = 9;$$

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p = (81 - 17 - 13) \bmod 23 = 5;$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p = (9 \cdot 2 - 20) \bmod 23 = 19;$$

$Q_2 = (5,19)$.

При $d = 4$ маємо:

$P_1 + Q_2 = Q_3$;

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\bmod p) = \frac{19 - 20}{5 - 17} \bmod 23 = \frac{-1}{-12} \bmod 23 = 2;$$

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p = (4 - 17 - 5) \bmod 23 = 5;$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p = (2 \cdot 12 - 20) \bmod 23 = 4;$$

$Q_3 = (5,4)$.

При $d = 5$

$P_1 + Q_3 = Q_4$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\, p) = \frac{4-20}{5-17} \mathrm{mod} 23 = \frac{-16}{-12} \mathrm{mod} 23 = 9;$$

$$x_3 = (\lambda^2 - x_1 - x_2) \mathrm{mod}\, p = (81 - 17 - 5) \mathrm{mod} 23 = 13;$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \mathrm{mod}\, p = (9 \cdot 4 - 20) \mathrm{mod} 23 = 16;$$

$Q_4 = (13,16)$.

При $d = 6$

$P_1 + Q_4 = Q_5$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\, p) = \frac{16-20}{13-17} \mathrm{mod}\, 23 = \frac{-4}{-4} \mathrm{mod}\, 23 = 1;$$

$$x_3 = (\lambda^2 - x_1 - x_2) \mathrm{mod}\, p = (1 - 17 - 13) \mathrm{mod}\, 23 = 17;$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \mathrm{mod}\, p = (1 \cdot 0 - 20) \mathrm{mod}\, 23 = 3;$$

$Q_5 = (17,3)$.

При $d = 7$

$P_1 + Q_5 = Q_6$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\, p) = \frac{3-20}{17-17} \mathrm{mod}\, 23 = \infty;$$

$Q_6 = (0,0)$.

Таким чином порядок n точки $G = (17, 25)$ на еліптичній кривій дорівнює 7.

Задача 6.

Побудувати $GF(2^m)$, m=4, $\alpha = x$, $f(x) = x^4 + x + 1$.

Розв'язок задачі:

Поле $GF(2^4)$ – може бути задано 16 поліномами не вище третього ступеня, наприклад, поліномами над полем $GF(2)$.

$$0, x, x^2, x^3, x+1, x^2+1, x^3+1, x^2+x, x^2+x+1,$$

$$x^3+x, x^3+x+1, x^3+x^2, x^3+x^2+x,$$

$$x^3+x^2+x+1, 1$$

Елементи поля можуть бути задані у двійковому вигляді, тоді, наприклад,

$x^3 + x^2 + 1 \Rightarrow 1101$

$x^3 + 1 \Rightarrow 1001$.

Операція множення елементів поля виконується в полі GF(P).

Наприклад:

$$(1101) \cdot (1001) \Leftrightarrow (x^3 + x^2 + 1) \cdot (x^3 + 1) (\mathrm{mod}\, f(x), 2) =$$

$$= (x^6 + x^5 + x^2 + 1) (\mathrm{mod}\, f(x), 2).$$

Зведемо цей поліном за модулем $f(x) = x^4 + x + 1$. У результаті маємо:

$$\begin{array}{l|l} X^6 + X^5 + X^2 + 1 & X^4 + X + 1 \\ \underline{X^6 + X^3 + X^2} & \\ X^5 + X^3 + 1 & X^2 + X \\ \underline{X^5 + X^2 + X} & \\ X^3 + X^2 + X + 1 & \end{array}$$

Отже: $(x^3+x^2+1)(x^3+1)(\mathrm{mod}\,(x^4+x+1),2)=x^3+x^2+x+1\Rightarrow 1111$.

В табл. 3.1 наведено елементи α^j для $j-\overline{0,7}$, якщо $\alpha=x$.

Таблиця 3.1 – Значення α^j

j	0	1	2	3	4	5	6	7
α^j	1	x	x^2	x^3	$x+1$	x^2+x	x^3+x^2	x^3+x^2+x+1

Елементи поля можна отримати як $\alpha^j=\alpha^j(\mathrm{mod}\,f(x),2),\ j-\overline{0,n}$.

Задача 7.

Нехай задано поле $GF\ (2^m)$, m=4, $\alpha=x$, $f(x)=x^4+x+1$.

Знайти $P_3=2P_1+P_2$, якщо $P_1=(0111,0010), P_2=(1100,1001)$.

Розв'язок задачі:

Знайдемо $2P_1$:

$$\lambda=0111+\frac{0010}{0111}.$$

Далі знайдемо зворотний елемент для числа 0111, яке подамо поліномом x^2+x+1.

Враховуючи, що:

$g^{p-2}=g^{-1}$;

$$(x^2+x+1)^{2^4-2}=(x^2+x+1)^{14}=((x^2+x+1)^2)^2\cdot((x^2+x+1)^2)^2\cdot((x^2+x+1)^2)^2\cdot(x^2+x+1)^2;$$

$$(x^2+x+1)^2=x^4+2x^3+3x^2+2x+1(\mathrm{mod}\,f(x),2)=x^2+x;$$

$$(x^2+x)^2=x^4+2x^3+x^2(\mathrm{mod}\,f(x),2)=x^2+x+1;$$

$$(x^2+x+1)^{14}=(x^2+x+1)(x^2+x+1)(x^2+x+1)(x^2+x)=(x^2+x)(x^2+x)(x^2+x+1)=$$

$$=(x^2+x+1)(x^2+x+1)=x^2+x;$$

$$\lambda=x^2+x+1+x(x^2+x)=x^2+x+1+x^3+x^2(\mathrm{mod}\,f(x),2)=x^3+x+1;$$

$$x_3=(x^3+x+1)^2+x^3+x+1+x+1=x^3+1+x^3+2x+2(\mathrm{mod}\,f(x),2)=1;$$

$$(x^3+x+1)^2=x^6+2x^4+2x^3+x^2+2x+1(\mathrm{mod}\,f(x),2)=x^3+1;$$

$$y_3=(x^3+x+1)^2+(x^3+x+1+1)\cdot 1=x^2+x+x^3+x(\mathrm{mod}\,f(x),2)=x^3+x^2;$$

Отже

$2P_1=(1,x^3+x^2)$.

Знайдемо $2P_1+P_2$:

$$\lambda = \frac{x^3 + 1 + x^3 + x^2}{x^3 + x^2 + 1};$$

$$(x^3 + x^2 + 1)^{14} = ((x^3 + x^2 + 1)^2)^2((x^3 + x^2 + 1)^2)^2((x^3 + x^2 + 1)^2)^2(x^3 + x^2 + 1)^2;$$

$$(x^3 + x^2 + 1)^2 = x^6 + 2x^5 + 2x^3 + x^4 + 2x^2 + 1(\mathrm{mod}\, f(x), 2) = x^3 + x^2 + 1;$$

$$(x^3 + x^2 + 1)^{14} = (x^3 + x^2 + 1)(x^3 + x^2 + 1)(x^3 + x^2 + 1)(x^3 + x^2 + 1)(\mathrm{mod}\, f(x), 2) = x^3 + x^2 + 1;$$

$$\lambda = (x^3 + x^2 + 1)(x^2 + 1) = x^5 + x^3 + 2x^2 + x^4 + 1(\mathrm{mod}\, f(x), 2) = x^3 + x^2;$$

$$x_3 = (x^3 + x^2)^2 + x^3 + x^2 + 1 + x^3 + x^2 + x + 1(\mathrm{mod}\, f(x), 2) = x^3 + x^2 + 1;$$

$$(x^3 + x^2)^2 = x^6 + x^4(\mathrm{mod}\, f(x), 2) = x^3 + x^2 + x + 1;$$

$$y_3 = (x^3 + x^2)(x^3 + x^2 + 1 + 1) + x^3 + x^2(\mathrm{mod}\, f(x), 2) = x^3 + x^2 + x^3 + x^2 + x + 1(\mathrm{mod}\, f(x), 2) = x + 1.$$

Отже

$$2P_1 + P_2 = (x^3 + x^2 + 1, x + 1).$$

3.4 Задачі для самостійного розв'язання

1. Нехай є ЕК з рівнянням: $y^2 = (x^3 + x + 1) \bmod 23$, a=1, b=1, p=23.
Перевірити, чи належать точки, що наведені в табл. 3.2, еліптичній кривій:

Таблиця 3.2 – Значення P_n

N	1	2	3	4	5	6	7	8	9	10	11	12
P_n	(3,10)	(3,13)	(4,0)	(5,4)	(5,19)	(6,4)	(6,19)	(7,11)	(7,2)	(9,7)	(9,16)	(17,3)
N	13	14	15	16								
P_n	(17,20)	(18,20)	(19,5)	(13,16)								

Якщо $n - k\ (mod\ 17)$, k – номер у журналі.
Якщо k=17, то k=k+1.

2. Знайти $2P_n$ (P_n з попередньої задачі).

3. Знайти $P_n + Q$, якщо $Q = 2P_n$ (P_n з попередньої задачі).

4. Знайти $2P_n$ і $P_n + Q$, якщо $Q = 2P_n$, а еліптична крива має вигляд $(0001)y^2 + (0001)xy \equiv (0001)x^3 + (0011)x^2 + 1(\mathrm{mod}\, x^4 + x + 1)$, $\alpha = 1100$

n	1	2	3	4	5	6	7
P_n	(α^6, α^8)	(α^{12}, α^8)	(α^6, α^{12})	$(\alpha^{22}, \alpha^{10})$	$(\alpha^{10}, \alpha^{15})$	$(\alpha^{12}, \alpha^{15})$	$(\alpha^{17}, \alpha^{14})$
Q_n	(α^9, α^{10})	(α^{10}, α^8)	$(\alpha^{12}, \alpha^{20})$	(α^9, α^{12})	(α^{13}, α^7)	$(\alpha^{14}, \alpha^{12})$	(α^9, α^8)
n	8						
P_n	(α^9, α^8)						
Q_n	(α^6, α^8)						

k – номер за списком, якщо k–8, то $k = k + 1;\ n = k \bmod 9$.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Дати визначення базової точки ЕК.
2. Запишіть та поясніть формулу складання точок еліптичної кривої над простим полем.
3. Запишіть та поясніть формулу подвоєння точок еліптичної кривої над простим полем.
4. Запишіть та поясніть формулу складання точок еліптичної кривої над розширеним полем.
5. Запишіть та поясніть формулу подвоєння точок еліптичної кривої над розширеним полем.
6. Яким вимогам відповідає примітивний поліном f(x)?
7. Скільки елементів у полі $GF(2^m)$, якщо m = 4, 9, 81, 160?
8. Скільки елементів у полі GF(P), якщо P = 17, 31, 47, 89, 107, 257?
9. Запишіть елементи поля GF(7).
10. Запишіть елементи поля
11. Запишіть та поясніть рівняння еліптичної кривої над простим полем GF(P).
12. Запишіть та поясніть рівняння еліптичної кривої над розширеним полем $GF(2^m)$.
13. Запишіть та поясніть рівняння еліптичної кривої в проективному вигляді.
14. Що забезпечує використання проективного базису?
15. Порівняйте стійкість RSA криптоперетворення та перетворення в групі точок ЕК.
16. Як можна знайти зворотні елементи в полі GF(q)?

Лабораторна робота № 4

Тема: *АНАЛІЗ АЛГОРИТМІВ ФОРМУВАННЯ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ*

Загальні положення

$$n_{00} = 44;\quad n_{01} = 40;\quad n_{10} = 40;\quad n_{11} = 35.$$

$$\chi_2 = \frac{4}{159}(44^2 + 40^2 + 40^2 + 35^2) - \frac{2}{160}(84^2 + 76^2) + 1 = 0{,}62.$$

3. Тест Поккера

Нехай m=3 і k=53. Блоки 000, 001, 010, 011, 100, 101, 110, 111 з'являються відповідно 5, 10, 6, 4, 12, 3, 6 та 7 разів. Значення параметра

$$\chi_3 = \frac{2^3}{53}\left(5^2 + 10^2 + 6^2 + 4^2 + 12^2 + 3^2 + 6^2 + 7^2\right) - 53 = 9{,}6415.$$

4. Тест серій

$$l_i = (n - i + 3)/2^{i+2};$$

$$l_1 = (160 - 1 + 3)/2^3 = 20{,}25;$$

$$l_2 = (160 - 2 + 3)/2^4 = 10{,}0625;$$

$$l_3 = (160 - 3 + 3)/2^5 = 5.$$

Є 25, 4 та 5 блоків довжиною 1, 2, 3 відповідно, 8, 20 та 12 інтервалів довжиною 1, 2 та 3 відповідно. Параметр χ_4 приймає значення $\chi_4 = 31{,}7913$.

5. Автокореляційний тест.

Якщо d=8, то $R(8)$=100.

Значення статистичного параметра

$$\chi_5 = \left(2\left(100 - \frac{160 - 8}{2}\right)\right) \Big/ \sqrt{160 - 8} = 3{,}8933.$$

Для рівня значущості $\alpha = 0{,}05$ порогові значення χ_1, χ_2, χ_3, χ_4 та χ_5 дорівнюють 3.8415, 5.9915, 14.0671, 9.4877, 1.96 відповідно.

Бачимо, що послідовність проходить частотний (монобітний) тест, двобітовий тест та тест Поккера, але не проходить тест серій та автокореляційний тест.

Таблиця 4.1 – Значення інтервалів

Довжина серії	Відповідний інтервал
1	2267 – 2733
2	1079 – 1421
3	502 – 748
4	223 – 402
5	90 – 223
6	90 – 223

Тест довжин серій. Цей тест проходить, якщо не існує ніяких серій довжиною 34 або більше. Для високозахищених застосувань FIPS – 140 – 1 зобов'язує, щоб 4 тести виконувалися при кожній ініціалізації генератора випадкових бітів. В FIPS-140-2 максимальна довжина серії збільшена до 36 бітів.

2. Необхідно розробити такі процедури мовою C++ або іншою мовою:

а) генерація псевдовипадкової послідовності довільної довжини з використанням лінійного рекурентного регістру, закон генерації якої визначається відповідно до примітивних поліномів, наведених в табл. 4.2;

Таблиця 4.2 – Примітивні поліноми

№ п/п	1	2	3	4	5
1	$x^{31}+x^3+1$	$x^{61}+x+1$	$x^{95}+x^{11}+1$	$x^{127}+x^{63}+1$	$x^{31}+x^3+1$
2	x^6+x+1	x^7+x+1	x^9+x^4+1	$x^{10}+x^3+1$	x^5+x^2+1
№ п/п	6	7	8	9	10
1	$x^{100}+x^{15}+1$	$x^{63}+x^5+1$	$x^{41}+x^3+1$	$x^{257}+x^{52}+1$	$x^{52}+x^{21}+1$
2	x^9+x^5+1	x^7+x^3+1	$x^{10}+x^7+1$	x^6+x^5+1	x^5+x^3+1

б) реалізація монобітного тесту згідно з вищесказаною теорією для довільної довжини послідовності;

в) реалізація тесту Поккера згідно з вищесказаною теорією для довільної довжини послідовності;

г) реалізація тесту серій згідно з вищесказаною теорією для довільної довжини послідовності;

д) реалізація тесту довгих серій згідно з вищесказаною теорією для довільної довжини послідовності.

3. Розв'язати задачі 2 (а-д) з використанням ЕОМ при довжині послідовності $l = 2 \cdot 10^4$ бітів.

4. Розв'яжіть задачу 1 пункту 1.10.2, якщо псевдовипадкова послідовність довжиною n=128 бітів побудована шляхом чотириразового повторення такої 32-бітової послідовності:

$$S=\{0110\ 0100\ 0111\ 1010\ 1100\ 1000\ 1111\ 0101\}.$$

Відповідь: (χ_1=0,5; χ_2=2,295; χ_3=1,048; χ_4=5,99; χ_5=0,18).

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Поясніть алгоритм функціонування генератора псевдовипадкових послідовностей
2. Назвіть основні показники оцінки властивостей генератора псевдовипадкових послідовностей.
3. Визначте період повторення лінійної рекурентної послідовності, якщо m={7, 10, 12, 19, 31, 63, 89, 127, 257, 52}.
4. Визначте структурну скритність лінійної рекурентної послідовності періоду $L = 2^m$ -1.
5. Визначте значення безпечного часу, якщо в генератор псевдовипадкової послідовності може бути введено N_k (10^{20}, 10^{25}, 10^{30}, 10^{35}, 10^{40}, 10^{50}, 10^{70},
6. і л 80 і л 90 і л100 і л127ч
7. 10 , 10 , 10 , 10) ключів.
8. Визначте структурну скритність лінійної рекурентної послідовності періоду $L = 2^m$ -1, m=20, 25, 30, 35, 40, 61, 63, 85, 89, 127, 257, 507.
9. Поясніть сутність:
   монобітного тесту;
   тесту двобітових серій;
   тесту Поккера;
   тесту серій;
   автокореляційного тесту.

   Поясніть вихідний код процедури реалізації:
   монобітного тесту;
   тесту двобітових серій;
   тесту Поккера;
   тесту серій;
   автокореляційного тесту.
10. Дайте визначення лінійного конгруентного генератора.
11. Які вимоги висуваються до коефіцієнтів *a* та *b* рекурентного співвідношення, що визначає алгоритм функціонування лінійного конгруентного генератора.
12. За яких умов лінійний конгруентний генератор забезпечує максимальний період формувальної послідовності.
13. Зробіть пропозиції щодо реалізації та виберіть параметри конгруентного генератора, що забезпечує максимальний період послідовності.

Лабораторна робота № 5

Тема: *ОЦІНКА АВТЕНТИФІКАЦІЇ ІНФОРМАЦІЇ*

Загальні положення

Задача 1.

Оцініть імовірність обману, якщо для забезпечення цілісності та справжності використовується імітоприкладка згідно з ГОСТ 28147 – 89 або FIPS-197.

Розв'язок задачі:

Імітоприкладка I_i виробляється згідно з ГОСТ 28147 – 89, 4-й режим. Вона є функцією повідомлення M, ключа зашифрування $K^з$ та ключа автентифікації $K^а$

$$I_м = f(M, K^з, K^а). \quad (5.23)$$

Згідно з теорією Сімонсена імовірність обману можна оцінити нерівністю

$$P_{іái} \geq 2^{-l_i},$$

де l_i – довжина імітоприкладки.

Оскільки l_i = 64 бітів, то

$$P_{обм} > 2^{-64} \approx 1{,}8 \cdot 10^{-19}.$$

Ми застосуємо в оцінці відповідно більше, оскільки не доведено, що в режимі виробки імітоприкладки забезпечується досконала автентичність.

Враховуючи, що довжина коду автентифікації для FIPS-197 l_i=128 біт, розв'яжіть задачу за аналогією з вищенаведеним.

Задача 2.

Розробіть та обґрунтуйте алгоритм блокового симетричного шифрування зі зв'язком по шифртексту, використавши алгоритм FIPS-197 з двома ключами – ключем зашифрування та ключем автентифікації.

Розв'язок задачі:

На рис. 5.5 наведено алгоритм зашифрування із зв'язком по шифрованому тексту. Інформація розбивається на блоки довжиною 128 бітів. Перший блок перед зашифруванням складається з ключем автентифікації K_a довжиною 128 бітів. Потім він зашифрується в блоковому режимі з використанням вихідного ключа зашифрування $K_з$. Процес повторюється для усіх блоків. При цьому пе-

ред зашифруванням M_i блоку він складається з C_{i-1} блоком криптограми. Отже, останній блок C_n залежить від усіх M_i блоків, ключа автентифікації та ключа зашифрування, тобто:

$$C_n = f(M_1, M_2, ..., M_n, K_a, K_з). \quad (5.24)$$

Тому C_n по суті є криптографічною контрольною сумою і може використовуватися як імітоприкладка $I_{мп}$ (згідно з міжнародною термінологією коду автентифікації повідомлень).

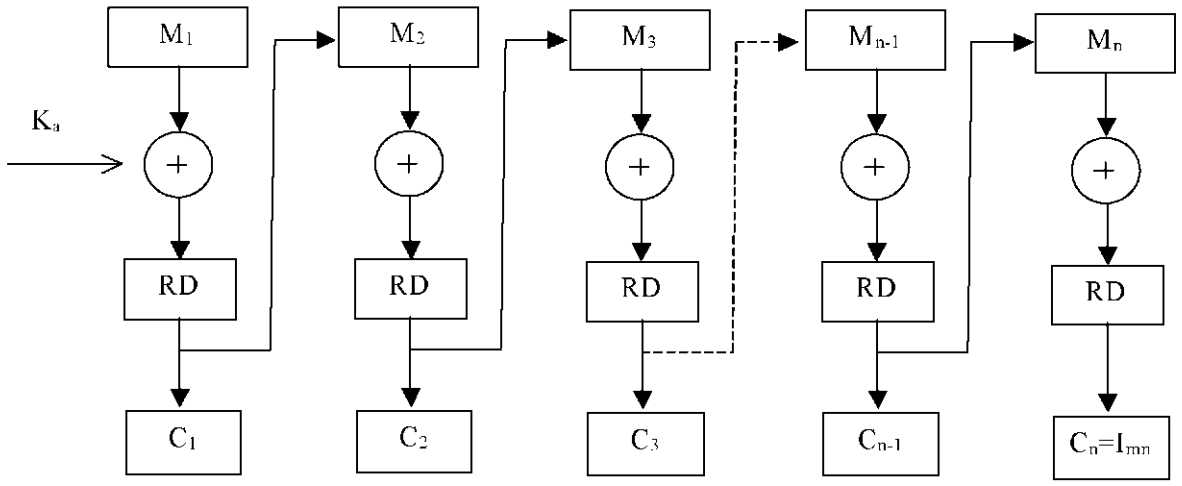


Рисунок 5.5 – Алгоритм зашифрування зі зв'язком по шифрованому тексту

Оскільки довжина імітоприкладки дорівнює $l_{імп}$=128 біт, то граничне значення ймовірності обману можна оцінити як:

$$P_{обм} \geq 2^{-128} \approx 2{,}9 \cdot 10^{-39}.$$

Необхідно зазначити, що одночасно з формуванням імітоприкладки здійснено зашифрування повідомлення М. При цьому довжина зашифрованого повідомлення дорівнює довжині вихідного повідомлення, тобто автентифікацію здійснено без збільшення довжини зашифрованого повідомлення. Особливістю цього режиму є те, що додатково використовується ключ автентифікації. Якщо повідомлення не має зашифровуватися, то в цьому випадку до нього додається імітоприкладка і довжина автентифікованого повідомлення збільшується на $l_б$=128 бітів, а саме автентифіковане повідомлення має вигляд $\{M, I_{імп}\}$. Основною перевагою такого методу автентифікації є висока швидкість перетворення і, як наслідок, можливість обчислення значення імітоприкладки в реальному плині часу. Основним недоліком є те, що використовувані ключі $K_з$ та $K_а$ є симетричними, а тому не дозволяють реалізувати захист на основі моделі взаємної недовіри.

Задача 3.

"Парадокс" днів народження. Ця задача може бути сформульована таким чином. Чому дорівнює мінімальне значення k, при якому ймовірність того, що, у крайній мірі, у двох осіб із групи в k осіб дні народження співпадуть, буде $P_з$.

Розв'язок задачі здійснити без урахування 29 лютого за умови, що кожний день народження рівноймовірний.

Розв'язок задачі:

Введемо ймовірність $P(365,k) \geq P_з$ та знайдемо ймовірність того, що в групі із k осіб дні народження не співпадуть, позначивши її як $Q(365,k)$. Зрозуміло, що:

$$P(365,k) + Q(365,k) = 1,$$

тому

$$P(365,k) = 1 - Q(365,k).$$

Зрозуміло також, що $k \leq 365$.

Знайдемо число різних способів N, якими можна отримати k значень без повторень. Для першого елемента ми маємо 365 значень без повторень, для другого 364, які залишилися, і так далі. Тому загальне число підходящих способів

$$N = 365 \cdot 364 \cdot \ldots \cdot (365 - k + 1) = \frac{365!}{(365-k)!}.$$

Якщо виключити умову відсутності співпадань днів народження, то кожний елемент (подія) може набувати будь-якого із 365 можливих значень. Загальна сума можливих подій дорівнює 365^k. Тому ймовірність відсутності співпадань дорівнює відношенню числа варіантів без співпадань до загального числа варіантів.

$$Q(365,k) = \frac{365!/(365-k)}{(365)^k} = \frac{365!}{(365-k)!(365)^k}. \qquad (5.25)$$

Тому

$$P(365,k) = 1 - Q(365,k) = 1 - \frac{365!}{(365-k)!(365)^k}.$$

В таблиці 5.1 наведено наближені значення $P(365,k)$.

Таблиця 5.1 – Значення P(365,k)

k	10	20	30	40	50	60	70
P	0,13	0,4	0,71	0,89	0,97	0,99	0,999

При інтуїтивному розгляді в розв'язку можна знайти парадокс. Це пов'язано з тим, що для кожної окремої особи в групі імовірність того, що з його днем народження співпадає день народження ще когось із групи, достатньо мала. Але тут необхідно розглядати усі пари людей. Наприклад, в групі із 40 осіб буде

$$\frac{40(40-1)}{2} = 780 \text{ різних пар,}$$

тому й імовірність для k=40 в таблиці достатньо велика.

Задача 4.

Нехай є деяка функція хешування $h=H(M)$, де M – інформація довільної довжини, причому h може приймати $n=2^m$ значень. Скільки випадкових повідомлень k треба подати на вхід перетворювача H, щоб відбулося з ймовірністю $P_з$ хоча б одне співпадання вигляду

$$H(M_i) = H(M_j),$$

тобто відбулася колізія.

Розв'язок задачі:

Розв'язок задачі ґрунтується на "парадоксі" про день народження (див. Задача 3). Але ця задача носить більш загальний характер.

У нашому випадку є цілочислова випадкова величина з рівноймовірним розподілом значень від 1 до n, та є вибірка із k значень випадкової величини ($k \le n$). Знайдемо ймовірність $P(n,k)$ того, що серед значень $H(M)$ у виборці, у крайньому випадку, дві співпадають

$$H(M_i) = H(M_j). \tag{5.26}$$

Використовуючи підхід, викладений вище під час розв'язку задачі 3, отримаємо узагальнення виразу (5.25)

$$P(n,k) = 1 - \frac{n!}{(n-k)!n^k}. \tag{5.27}$$

Для спрощення розрахунків вираз (5.27) можна спростити. Для цього використовуємо те, що справедливо

$$(1-x) \le e^{-x}, \text{ для усіх } x \ge 0. \tag{5.28}$$

Крім того, за малих значень x (наприклад, $x \le 0{,}1$) можна вважати, що:

$$(1-x) \approx e^{-x}. \tag{5.29}$$

Далі запишемо (5.27) у вигляді:

$$\begin{aligned} P(n,k) &= 1 - \frac{n(n-1)\cdot\ldots\cdot(n-k+1)}{n^k} = \\ &= 1 - \left[\frac{n-1}{n}\cdot\frac{n-2}{n}\cdot\ldots\cdot\frac{n-k+1}{n}\right] = \\ &= 1 - \left[\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot\ldots\cdot\left(1-\frac{k-1}{n}\right)\right]. \end{aligned} \tag{5.30}$$

Оскільки в нашому випадку $\frac{1}{n},..\frac{k-1}{n} \le 0{,}1$, то зробимо в (5.30) заміну, використовуючи (5.28).

У результаті маємо

$$P(n,k) = 1 - \left[\left(e^{-1/n}\right)\cdot\left(e^{-2/n}\right)\cdot\ldots\cdot\left(e^{-k-1/n}\right)\right] =$$
$$= 1 - e^{-\left[(1/n)+(2/n)+\ldots+((k-1)/n)\right]} = \qquad (5.31)$$
$$= 1 - e^{-(k(k-1))/2n}.$$

Розв'язок можна отримати, розв'язавши (5.31)
$1 - e^{-(k(k-1))/2n} = P_з$, оскільки $P_з$ відомо. Оскільки реально $P_з \le 1$,
то

$$1 - P_з = e^{-(k(k-1))/2n}$$

або

$$\ln(1 - P_з) = -k(k-1)/2n,$$

і далі

$$\frac{k(k-1)}{2n} = -\ln(1 - P_з);$$
$$k(k-1) = -2n\ln(1 - P_з).$$

У кінцевому вигляді маємо рівняння

$$k^2 - k + 2n\ln(1 - P_з) = 0. \qquad (5.32)$$

Нехай $P_з$=0,5, тоді маємо

$$k^2 - k + 2n\ln 0{,}5 = k^2 - k - 2n\ln 2 = 0.$$

При $n=2^m$ рівняння набуває вигляду

$$k^2 - k - 2^{m+1}\ln 2 = 0. \qquad (5.33)$$

Дамо оцінку значення k, враховуючи, що k достатньо велике і $k^2 >> k$.
Тоді із (5.32) маємо

$$k^2 = -2n\ln(1 - P_з).$$

При $P_з$ =0,5 маємо

$$k^2 = -2n\ln\left(1 - 1/2\right) = 2n\ln 2$$

і

$$k = \sqrt{2(\ln 2)\cdot n} = 1{,}18\sqrt{n}. \qquad (5.34)$$

При $n=2^{160}$, маємо $k=\sqrt{2^{160}}=2^{80}$.
При $n=2^{256}$, $k=2^{128}$.
При довільному значенні $P_з$

$$k = \sqrt{2\ln\left(\frac{1}{1-P_з}\right)\cdot n} = 1{,}41\sqrt{\ln\left(\frac{1}{1-P_з}\right)\cdot n}. \qquad (5.35)$$

Співвідношення (5.35) дозволяє оцінити число експериментів, які необхідно виконати для здійснення колізії типу (5.26).

1. Оцініть імовірність обману в режимі виробки та використання для забезпечення цілісності та справжності кодів автентифікації повідомлень (КАП) з довжиною $L_{КАП}$ = 14, 32, 64, 128, 192 та 256 бітів. Визначте, які криптоалгоритми для автентифікації із зазначеною довжиною можна застосувати.

2. Визначіть мінімальне значення k, при якому ймовірність того, що у крайній мірі у двох осіб із групи в k осіб дні народження співпадуть з ймовірністю $P_з$=0,5+0,01·r , де r – номер за журналом реєстрації.

3. Визначіть скільки випадкових повідомлень необхідно подати на вхід засобу розрахунку хеш-функції $H(Mi)$, щоб з ймовірністю $P_з$=0,5+0,01·r була здійснена колізія, якщо n=2^{m+k}, m=192, r – номер реєстрації за журналом.

4. Розв'яжіть рівняння

$$k^2 - k + 2n\ln(1 - P_з) = 0,$$

якщо $P_ç = 0{,}5 + 0{,}01k$, $n = 2^{192+r}$.

Визначте складність та безпечний час здійснення колізії для отриманого значення k, якщо потужність криптоаналітичної системи складає 10^8,10^{10}, 10^{12} та 10^{16} опер./с., а r – номер реєстрації за журналом.

5. Дайте оцінку числа експериментів k здійснення колізій, використовуючи співвідношення (1.185)

$$k = 1{,}41\sqrt{\ln\left(\frac{1}{1 - P_ç}\right)n},$$

якщо $P_з$=0,5+0,01r, n=2^{192+r}, де r – номер реєстрації за журналом.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Визначте поняття цілісності та справжності, яким чином вони забезпечуються?
2. Для чого здійснюється автентифікація повідомлень?
3. Як здійснити виробку імітовкладки з використанням симетричного криптоалгоритму?
4. Оцініть ймовірність обману в інформаційній технології, якщо для цього використовується алгоритм AES FIPS-197 з довжиною блоку 128 бітів.

5. Які основні переваги та недоліки методу автентифікації, що базується на використанні імітовкладки?

6. В чому суть парадоксу дня народження?

7. В чому суть явища виникнення колізій та як його можна реалізувати.

8. За якими показниками можна оцінити складність створення колізій?

9. Як залежить складність створення колізій від довжини хеш-функції.

10. Побудуйте графіки залежності

$$f_1(x) = 1 - x \text{ та } f_2(x) = e^{-x}$$

для x=0;0,1;0,2;0,3;0,4;0,5;0,6;0,7;0,8;0,9;1,0 та знайдіть значення параметра, при якому похибка складає не більше 10%.

11. Яка ймовірність того, що в групі студентів із 50 осіб двоє з них народилися в один день?

12. Яка ймовірність колізії на виході функції хешування, якщо довжина вихідного значення 128, 160, 192, 224, 256, 512 бітів і зроблено $k=2^{128}$ експериментів?

13. В чому суть парадоксу дня народження?

14. Знайдіть ймовірність перекриття шифруючої гами з періодом 2^{128},

о ^{100} якщо згенеровано відрізок 2 .

15. Знайдіть довжину відрізка гами шифруючої потокового шифру, за якої ймовірність перекриття не перевищує 0,6 , якщо період гами 2^{128}.

16. Як обчислити ймовірність колізії, якщо в (5.28) ^{x} - ^{0,5}.

Лабораторна робота № 6

Тема: *ОЦІНКА АВТЕНТИЧНОСТІ ІНФОРМАЦІЇ*

Загальні положення

Задача 1.

Оцініть імовірність обману, якщо для забезпечення цілісності та справжності використовується електронний цифровий підпис (ЕЦП) DSA.

Розв'язок:

Спочатку зробимо оцінку, використовуючи границю Сімонсена

$P_{обм} > 2^{-l_{ЦП}}$, де $l_{ЦП} = 160$ – довжина ЕЦП, який використовується.

$P_{обм} > 2^{-160} \approx 6{,}8 \cdot 10^{-49}$.

Відомо також, що імовірність обману можна визначити як

$$P_{обм} > \frac{2^{l_M}}{2^{l_M + l_{ЦП}}},$$

де l_M – довжина інформації, яка підписується, а $l_{ЦП}$ – довжина ЕЦП.

З цього співвідношення маємо

$$P_{обм} > 2^{l_M} \cdot 2^{-l_M - l_{ЦП}} = 2^{-l_{ЦП}}.$$

Отже, оцінка Сімонсена співпадає з останньою, отриманою через розмір множин повідомлень та автотентифікованих повідомлень за допомогою ЕЦП.

Задача 2.

Нехай ЕЦП здійснюється з використанням ECDSA, причому використовується крива $y^3 \equiv x^3 + x + 1 (\mathrm{mod}\, 23)$.

Як базова використовується точка $G = (13,7)$ з порядком $n = 7$.

Необхідно:

– виробити особистий та відкритий ключі;

– виробити цифровий підпис згідно з ECDSA, якщо $l = H(M) = 6$;

– перевірити цілісність та справжність повідомлення, у якого хеш-функція $l = H(M) = 6$.

Розв'язок прикладу:

Виробка ключів. Оскільки $n = 7$, то ключем може бути будь-яке ціле $1 \le d \le n - 1 = 6$.

Виберемо $d = 3$. Тоді відкритий ключ

$$Q = d \cdot G (\mathrm{mod}\, p) = 3 \cdot (13,7) (\mathrm{mod}\, 23),$$

тобто

$$Q = ((13,7) + (13,7) + (13,7)) \mathrm{mod}\, 23.$$

Використовуючи афінний базис, спочатку визначаємо (13,7)+(13,7) = 2(13,7), тобто здійснимо подвоєння.

Якщо R_1 та R_2 є точками на еліптичній кривій відповідно з коефіцієнтами (x_1, y_1) та (x_2, y_2), то

$$R_1 + R_2 = (x_1, y_1) + (x_2, y_2) = R_3 = (x_3, y_3),$$

причому

$$x_3 = \lambda^2 - x_1 - x_2 (\mathrm{mod}\, p),$$
$$y_3 = \lambda (x_1 - x_3) - y_1 (\mathrm{mod}\, p),$$
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\, p).$$

При подвоєнні $R_3 = 2R_1 = 2(x_1, y_1) = (x_3, y_3)$, причому

$$x_3 = \left(\lambda^2 - 2x_1 \right) (\mathrm{mod}\, p),$$

$$y_3 = (\lambda(x_1 - x_3) - y_1)(\mathrm{mod}\ p),$$

$$\lambda = \frac{3x_1^2 + a}{2y_1}(\mathrm{mod}\ p).$$

При подвоєнні точки (13,7) маємо (a=1)

$$\lambda = \frac{3 \cdot 13^2 + 1}{2 \cdot 7}(\mathrm{mod}23) = \frac{508}{14}(\mathrm{mod}23) = \frac{1}{7}(\mathrm{mod}23).$$

Знайдемо для 7 зворотний елемент, розв'язавши порівняння

$$7 \cdot z \equiv 1(\mathrm{mod}\,23)$$

або еквівалентне цьому діафантове рівняння

$$23(-k) + 7z = 1.$$

Позначивши x – -k та y–z, маємо $23x + 7y = 1$.

Розв'язок шукаємо у вигляді

$$x = (-1)^{\mu+1} b_{\mu-1},$$

$$y = (-1)^{\mu} a_{\mu-1},$$

де μ – кількість членів ланцюгового дробу розкладу $23/7$.

Подамо a/b у вигляді ланцюгового дробу

$$\frac{a}{b} = r_0 + \cfrac{1}{r_1 + \cfrac{1}{r_2 + \cfrac{1}{r_3 + \ddots \cfrac{1}{r_\mu + 0}}}}.$$

В результаті маємо

$$\frac{23}{7} = 3 + \cfrac{1}{3 + \cfrac{1}{2 + 0}}.$$

Отже $r_0 = 3$, $r_1 = 3$, $r_2 = 2$, тоді $\mu = 2$

$$a_{\mu-1} = a_1 = r_0 r_1 + 1;$$

$$b_{\mu-1} = b_1 = r_1.$$

Дійсно, $\frac{a_1}{b_1} = r_0 + \frac{1}{r_1} = \frac{r_0 r_1 + 1}{r_1}$.

Тоді

$$z = (-1)^2 a_1 = r_0 r_1 + 1 = 3 \cdot 3 + 1 = 10.$$

Отже

$$\lambda = 1 \cdot 10(\mathrm{mod}\,23) = 10,$$

$$x_3 = \left(\lambda^2 - 2x_1\right)\mathrm{mod}23 = (10^2 - 2 \cdot 13)\,\mathrm{mod}23 = 74(\mathrm{mod}23) = 5,$$

$$y_3 = \left(\lambda(x_1 - x_3) - y_1\right)\mathrm{mod}\,23 = (10(13 - 7)\,\mathrm{mod}\,23 = 4.$$

Отже

$$2(x_1, y_1) = 2(13,7) = (x_3, y_3) = (5,4).$$

Далі знайдемо

$$Q = 2(x_1, y_1) + (x_3, y_3) = (5,4) + (13,7).$$

Використовуючи формули для складання точок, маємо

$$\lambda = \frac{y_3 - y_1}{x_3 - x_1} (\text{mod} 23) = \frac{7-4}{13-5} (\text{mod} 23) = \frac{3}{8} (\text{mod} 23) = 3 \cdot 3 = 9,$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \text{mod } p = (9(5-17) - 4) \text{mod } 23 = -112 (\text{mod } 23) = 3.$$

Аналогічно x_3 дорівнює 17.

Отже, $R_3 = (x_3, y_3) = (17,3)$.

Результат:

особистий ключ $d = 3$;

відкритий ключ $Q = (17, 3)$.

Виробку ЕЦП виконаємо згідно з [25], в результаті маємо:

1.Виробимо $k = 4 \in [1,6]$.

2.Обчислимо $(x_1, y_1) = k \cdot G = 4(13,7) = (17,20)$.

3.Знаходимо $x_1 = 17$.

4.Обчислимо перший компонент ЕЦП

$$r = x_1 (\text{mod } n) = 17 (\text{mod } 7) = 3.$$

5.Обчислимо другий компонент ЕЦП

$$s = k^{-1}(l + dr) \text{mod } n = 4^{-1}(6 + 3 \cdot 3) \text{mod } 7 = 2(6 + 3 \cdot 3) \text{mod } 7 = 2.$$

Отже

$$(r, s) = (3, 2).$$

Здійснимо перевірку ЕЦП.

Нехай $(r', s') = (3,2)$.

1.Відкритий ключ $Q = (17, 3)$.

2.Обчислимо $l' = H(M') = 6$.

3.Обчислимо

$$w = (s')^{-1} \text{mod } n = (2^{-1}) \text{mod } 7 = 4.$$

4.Обчислимо

$$u_1 = l'w (\text{mod } n) = 6 \cdot 4 (\text{mod } 7) = 3 \quad \text{і} \quad u_2 = r'w (\text{mod } n) = 3 \cdot 4 (\text{mod } 7) = 5.$$

5.Знаходимо точку

$$u_1 G + u_2 Q = 3G + 5Q = 3(13,7) + 5(17,3) = (17,20).$$

6.Знайдемо $x_1 = 17$.

7.Обчислимо $v = x_1 (\text{mod } n) = 17 (\text{mod } 7) = 3$.

8.Оскільки $v = r' = 3$, то повідомлення цілісне та справжнє.

1. Оцініть імовірність обману, якщо для ЕЦП використовуються криптографічні алгоритми ГОСТ 34.310 – 95 або ГОСТр.34.10 – 2001.

2. Оцініть імовірність обману, якщо довжина ЕЦП $L_{ЦП}$ = 64, 96, 128, 160, 192, 224, 256 та 512 бітів.

3. Розв'яжіть приклад 2 при значенні $l = H(M) = 1,2,3,4$ та 5.

4. Розв'яжіть приклад 2 в частині перевірки ЕЦП, якщо:

1) значення хеш – функції $l' = l + 1 (\mathrm{mod}\, n)$;

2) значення r вибрано довільним чином $(1 \le r \le n-1)$;

3) значення s вибрано довільним чином $(1 \le s \le n-1)$;

4) значення r, s та l вибрано довільно.

5. Розрахуйте складність криптоаналізу дискретного логарифму в групі точок еліптичної кривої, якщо

$$n = 2^{192}, 2^{224}, 2^{256} \text{ та } 2^{512}.$$

6. Знайдіть безпечний час криптосистеми ЕЦП у групах точок ЕК, якщо $\gamma = 10^5, 10^6$ та 10^7 *оп.ск/ЕК*, а $n = 2^{192}$, $2^{224}, 2^{224}, 2^{256}$ та 2^{512}.

7. Порівняйте цифрові підписи DSA та ECDSA за показниками криптографічної стійкості та складності виконання.

8. Виробіть цифровий підпис в групі точок еліптичної кривої, використовуючи алгоритм задачі 2, якщо $e' = k (\mathrm{mod}\ n)$, де k – номер реєстрації.

Якщо $e' = 0$, то e' =4.

9. Перевірте правильність ЕЦП, у якого хеш-функція має значення e', прийнявши як параметри значення із задачі 2, послідовно довільним чином:

1. Параметр *r* та здійсніть перевірку ЕЦП відповідно до задачі 2.

2. Параметр *s* та здійсніть перевірку ЕЦП відповідно до задачі 2.

3. Обидва параметри *r* і *s* та здійсніть перевірку ЕЦП згідно з задачею 2.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Дайте визначення ЕЦП.
2. Яким вимогам має відповідати ЕЦП?
3. З використанням яких перетворень можна здійснити ЕЦП?
4. З використанням якого ключа та яким чином здійснюється виробка ЕЦП?
5. Яким чином здійснюється перевірка ЕЦП?

6. Порівняйте можливості існуючих ЕЦП.
7. Оцініть складність криптоаналізу DSA ЕЦП.
8. Оцініть складність криптоаналізу ЕЦП ECDSA.
9. Чому складність криптоаналізу ECDSA на багато більша, ніж DSA?
10. Поясніть алгоритм виробки підпису ECDSA.
11. Поясніть алгоритм перевірки підпису ECDSA.
12. Порівняйте ЕЦП ECDSA та DSA. Порівняйте складність криптоаналізу в полях та в групах точок еліптичних кривих.
13. Як формуються відкриті ключі?
14. Як пов'язані між собою порядок кривої та порядок базової точки?
15. Дайте загальну характеристику електронного цифрового підпису, що реалізується в групі точок еліптичної кривої.

Лабораторна робота № 7

Тема: *КРИПТОАНАЛІЗ RSA ТА ДИСКРЕТНИХ ЛОГАРИФМІВ МЕТОДАМИ ρ –ПОЛАРДА*

Загальні положення

Задача 1.
Факторизувати модуль N, використавши метод ρ-Поларда.
Дано, що $N = 221$, $a = 11$, $x_0 = 127$, $C = 1$.

Розв'язок задачі:
$x_1 = (127 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$,
$x_1 = 118$; НСД(127-118, 221) = (9, 221)=1, тобто спільних дільників немає.
$x_2 = (118 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$,
$x_2 = 19$; НСД (118-19,221) = НСД (99,221) = 1.
Таким чином НСД ($x_1 - x_2$, N) знову не має сильного дільника.
$x_3 = (19 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$.
Розрахуємо послідовно x_i доти, доки НСД ($x_1 - x_2$, N) різнитиметься від 1 або N.
$x_3 = 210$; НСД (191,221) = 1,
$x_4 = (210 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$,
$x_4 = 7$; НСД (203 , 221) =1,
$x_5 = (7 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$,
$x_5 = 78$; НСД (71,221) = 1,

$x_6 = (78 \cdot 11 + 1)(\mathrm{mod}\, 2^8)$,

$x_6 = 91$. При x_6 маємо НСД $(x_6 - x_5, N)$ = НСД (91 – 78, 221) =НСД (13, 221) = 13. Таким чином один із співмножників модуля N є 13, наприклад, P=13. Тоді Q=N/P=221/13=17. Перевіряємо правильність розв'язку, знайшовши P*Q = 221.

Задача 2.

Факторизувати число N = 209, якщо $x_i = x^2 + 1(\mathrm{mod}\, N)$.

$f(x) = x_{i-1}^2 + 1(\mathrm{mod}\, 209) = x_i$, прийнявши, що $x_0 = 17$.

Розв'язок задачі:

$x_1 = (17^2 + 1)\mathrm{mod}\ 209 = 81$,

$x_2 = (81^2 + 1)\mathrm{mod}\ 209 = 83$,

$d_1 = |x_2 - x_1| = |83 - 81| = 2$;

НОД (2, 209) = 1 розв'язку немає.

$x_3 = (83^2 + 1)\mathrm{mod}\ 209 = 202$,

$x_4 = (202^2 + 1)\mathrm{mod}\ 209 = 50$,

x_3 - x_4 = |202-50| = 152.

НОД (152, 209) = 19.

Таким чином один із співмножників, наприклад, P=19, тоді Q=209/19=11.

Задача 3.

Розв'язати порівняння $15^x \equiv 9(\mathrm{mod}\ 23)$ методом ρ-Поларда. Нехай c = 6.

Розв'язок задачі:

Використовуючи (7.9), (7.10) розрахунки зведемо в таблицю. Знайдемо

$$x_{i+1} \equiv x_i \cdot 9(\mathrm{mod}\, 23), x_i \le 6;$$

$$x_{i+1} = x_i \cdot 15(\mathrm{mod}\, 23), x_i > 6.$$

Розрахунки зведемо в табл. 7.1.

Таблиця 7.1 – Розрахунки

Цикл	Довжина	x_i	u	V	x_i	t	v
1	2	$X_0 = 9$	0	1	$X_1 = 20$	1	1
		$X_1 = 20$	1	1	$X_2 = 1$	2	1
		$X_2 = 1$	2	1	$X_3 = 9$	2	2
		$X_3 = 9$	2	2	$X_4 = 20$	3	2
					$X_5 = 1$	4	2
					$X_6 = 9$	4	3

Таким чином при u=2,v=2, і u=4,v=3 отримуємо одне і теж саме значення x_6=x_3=9.

Підставивши значення в ($v = 2$, $w = 3$, $t = 4$, $u = 2$), отримаємо

$$x = (4-2)\cdot(2-3)^{-1}(\text{mod } 22) = 2\cdot\frac{1}{-1}(\text{mod } 22) = -2(\text{mod } 22) = 20(\text{mod } 22).$$

Задача 4.

Розв'язати рівняння методом ρ-Поларда $20 \equiv 7^x \pmod{23}$.

$$\begin{cases} r_i = r_{i-1}b(\text{mod } p), \textit{якщо} & r_{i-1} < c; \\ r_i = r_{i-1}a(\text{mod } p), \textit{якщо} & r_{i-1} \geq c; \end{cases}$$

$c = 10$, $r_0 = b = 20$, $a = 7$.

Розв'язок задачі:

$$r_0 = b^1 \cdot a^0(\text{mod } p) = 20^1 \cdot 7^0(\text{mod } 23) = 20;$$

$$r_1 = r_0 \cdot a(\text{mod } p) = 140(\text{mod } 23) = 2;$$

$$r_1 = 20^1 \cdot 7^1;$$

$$r_2 = r_1 \cdot b(\text{mod } p) = 2 \cdot 20^1(\text{mod } 23) = 17;$$

$$r_2 = 20^2 \cdot 7^1;$$

$$r_3 = r_2 \cdot a(\text{mod } p) = 119(\text{mod } 23) = 4;$$

$$r_3 = 20^2 \cdot 7^2;$$

$$r_4 = r_3 \cdot b(\text{mod } p) = 11;$$

$$r_4 = 20^3 \cdot 7^2;$$

$$r_5 = r_4 \cdot a(\text{mod } p) = 8;$$

$$r_5 = 20^3 \cdot 7^3;$$

$$r_6 = r_5 \cdot b(\text{mod } p) = 22;$$

$$r_6 = 20^4 \cdot 7^3;$$

$$r_7 = r_6 \cdot a(\text{mod } p) = 16;$$

$$r_7 = 20^4 \cdot 7^4;$$

$$r_8 = r_7 \cdot a(\text{mod } p) = 20.$$

Таким чином $r_8 = r_0$, причому r_8 можна подати як

$$u = 0, t = 5;$$

$$v = 1, w = 4;$$

$$x = \frac{5}{1-4}(\text{mod } 22) = \frac{17}{3}(\text{mod } 22) = 17 \cdot 15(\text{mod } 22) = 13;$$

$$x = 13.$$

1. Оцініть імовірність обману, якщо для ЕЦП використовуються криптографічні алгоритми ГОСТ 34.310 – 95 або ГОСТр.34.10 – 2001.

2. Оцініть імовірність обману, якщо довжина ЕЦП $L_{ЦП}$ = 64, 96, 128, 160, 192, 224, 256 та 512 бітів.

3. Розв'яжіть приклад 2 при значенні $l = H(M) = 1,2,3,4$ та 5.

4. Розв'яжіть приклад 2 в частині перевірки ЕЦП, якщо:

1) значення хеш – функції $l' = l + 1 (\mathrm{mod}\, n)$;

2) значення r вибрано довільним чином $(1 \le r \le n-1)$;

3) значення s вибрано довільним чином $(1 \le s \le n-1)$;

4) значення r, s та l вибрано довільно.

5. Розрахуйте складність криптоаналізу дискретного логарифму в групі точок еліптичної кривої, якщо

$$n = 2^{192}, 2^{224}, 2^{256} \text{ та } 2^{512}.$$

6. Знайдіть безпечний час криптосистеми ЕЦП у групах точок ЕК, якщо $\gamma = 10^5, 10^6$ та 10^7 *оп.ск/ЕК*, а $n = 2^{192}$, $2^{224}, 2^{224}, 2^{256}$ та 2^{512}.

7. Порівняйте цифрові підписи DSA та ECDSA за показниками криптографічної стійкості та складності виконання.

8. Виробіть цифровий підпис в групі точок еліптичної кривої, використовуючи алгоритм задачі 2, якщо $e' = k (\mathrm{mod}\ n)$, де k – номер реєстрації.

Якщо $e' = 0$, то $e' = 4$.

9. Перевірте правильність ЕЦП, у якого хеш-функція має значення e', прийнявши як параметри значення із задачі 2, послідовно довільним чином:

1. Параметр r та здійсніть перевірку ЕЦП відповідно до задачі 2.

2. Параметр s та здійсніть перевірку ЕЦП відповідно до задачі 2.

3. Обидва параметри r і s та здійсніть перевірку ЕЦП згідно з задачею 2.

Завдання:

Використовуючи будь-яку мову програмування на ваш вибір, скласти алгоритм та програму розв'язання задач для самостійного розв'язку. Бути готовим прокоментувати роботу програми та відповісти на контрольні запитання. Результат виконання лабораторної роботи викласти у звіті, який повинен містити: титульну сторінку, номер та тему лабораторної роботи, мету, зміст задачі, програмний код та прінт-скрін виконання програми, відповідь на контрольне запитання (відповідно номеру в журналі академгрупи), висновок.

Контрольні запитання

1. Охарактеризуйте сутність методу ρ -Поларда?
2. Чому метод Поларда називають ρ -методом?
3. Яка ознака того, що розв'язок знайдено?
4. Що таке «факторизація модуля»?
5. Які вимоги до модуля *N*?
6. Як розраховується значення функції Ойлера для RSA перетворення?

7. Дайте оцінку складності RSA криптоаналізу методом ρ -Поларда?
8. Які прості числа називаються «сильними»?
9. В чому полягає криптоаналіз дискретних логарифмів методом ρ -Поларда?
10. Як обирають коефіцієнти *c* при розв'язку дискретного логарифмічного порівняння?
11. Як знайти НСД заданих двох чисел?
12. Які обмеження притаманні методу ρ-Поларда при факторизації та розв'язку дискретних логарифмічних рівнянь?
13. Визначте складність криптоаналізу RSA методом ρ-Поларда, якщо довжина модуля дорівнює 1024 або 2048 бітів.
14. Визначте складність криптоаналізу дискретного логарифму методом ρ -Поларда, якщо довжина модуля дорівнює 1024 або 2048 бітів.
15. Дайте характеристику алгоритму Евкліда знаходження НСД.

Порівняйте складність розв'язку задач факторизації та розв'язку дискретного логарифму

Список рекомендованої літератури

1. Нечаев В. И. Элементы криптографии М.: Высшая школа, 1999.
2. Ященко В.В. Введение в криптографию М.: МЦНМО—ЧеРо, 2000
3. Шапошников И.В Web- сервисы Microsoft .Net СПб.: БХВ-Петербург, 2002.-336 с
4. Кормич В.А. Інформаційна безпека: організаційно-правові основи: Навчальний посібник./МОН.-К.:Кондор, 2008.-283с.
5. Юдін О.І., Корченко О.Г., Конахович Г.Ф., Захист інформації в мережах передачі даних – К.: Вид-во ТОВ "НВП"Інтерсервіс", 2009.-716с.
6. Бардачов Ю. М., Соколова Н. А., Ходаков В. Є. Дискретна математика // Підручник / За ред. В. Є. Ходакова. – К. : Вища школа, 2002. – 287 с.
7. Борисенко О. А. Лекції з дискретної математики: (множини і логіка) // Навчальний посібник. – 3-є вид., випр. і доп. – Суми: ВТД “Університетська книга”, 2002 – 180 с.
8. Капітонова Д. В., Кривий С. Л., Летичевський О. А. Основи дискретної математики // Підручник / НАН України. МОН України – К. : Наукова думка, 2002. – 579 с.
9. Хенлі Ернест Джон, Кумамото Хиромицу Надійнісне проектування технічних систем і оцінка ризику./ пер. з англ. О.Ю.Зареніна, В.Ф. Хмеля; під ред. Ю.Г.Зареніна.-К.: Вища школа, 1987.-544с.
10. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. — М.: Триумф, 2002.
11. http://www.mat.net.ua/mat/index-teoriya-chisel.htm
12. http://www.dut.edu.ua/uploads/l_1359_89216009.pdf
13. http://www.dut.edu.ua/uploads/l_1134_27449793.pdf

Методичне видання

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
“ОСНОВИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ”

для студентів денної та заочної форми навчання
за напрямом підготовки
6.170103 “Управління інформаційною безпекою”,
спеціальністю 125 “Кібербезпека”

Видання оновлене та доповнене

Укладач
Лисенко Ірина Анатоліївна