МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж



ЗВІТ

про виконання лабораторної роботи № 2

«Блокове шифрування інформації та шифри моноалфавітної заміни»

з дисципліни

«Технології захисту інформації»

Виконала:

студентка групи ІТ-31

Щербак Л. В. Прийняв викладач:

Досин Д.Г.

Львів 2022

**Мета роботи:** Навчитися опрацьовувати (шифрувати та дешифрувати) файли з допомогою блокового шифрування інформації, методів моноалфавітної заміни та методу гамування.

**Завдання роботи:**

індивідуальне завдання з детальним формулюванням розв’язуваної

задачі, використовувані (власні) теоретичні відомості;

1. Написати програму на мові С++ (чи іншій за згодою викладача) яка

виконує шифрування повідомлення методом накладення гамми

використовуючи таблицю 1. Гамма шифру і повідомлення вибираються

згідно з варіантом з табл. 3.





3. Використовуючи алфавіт (табл. 1), дешифрувати повідомлення, яке

зашифроване методом гамування. Гамма шифру та повідомлення

вибираються згідно з варіантом з таблиці 4.





4. Оформити алгоритм функціонування програми та здійснити його опис

5. Оформити звіт в згідно вимог, результати подати у вигляді текстів

програми, відповідних пояснень та скріншотів.

6. Додаткове завдання. Здійснити написання програми яка виконує

шифрування даних над файлами за одним з методів відповідно до заданого

варіанту приведеного в таблиці. Інформацію (текстовий блок для

шифрування) взяти у відповідності до вибраної предметної галузі

(узгоджується з викладачем) але не менше ніж 200 слів.



**Відповіді на контрольні питання:**

1. Поняття блокового шифрування інформації?

Робота блокового шифру в найпростішому режимі — застосування функції, що шифрує, до блоку даних (проста заміна) викликає серйозну проблему: статистичні властивості відкритих даних частково зберігаються, тому що кожному однаковому блоку даних однозначно відповідає зашифрований блок даних. При великій кількості даних (відео, звук) це може дати деякі відомості для криптоаналізу про зміст даних.

2. Мета та особливості використання операції зсуву?

Логічний зсув

При логічному зсуві значення останнього біта за напрямом зсуву втрачається (копіюємо в біт перенесення), а перший набуває нульове значення.Логічні зсуви на одиницю вліво та вправо використовуються для множення та ділення на 2, відповідно. І на степінь двійки якщо зсув не на одиницю. Завдяки тому, що зсув займає менше тактів процесора ніж множення, його використовували замість множення чи ділення на степінь двійки. Але з сучасними оптимізованими компіляторами така діяльність небажана через можливість помилок.

Арифметичний зсув

Арифметичний зсув аналогічний логічному, але значення слова вважається знаковим числом, представленим в додатковому коді. Так, при правому зсуві старший біт зберігає своє значення. Лівий арифметичний зсув ідентичний логічному.

Циклічний зсув

При циклічному зсуві, значення останнього біта за напрямом зсуву копіюється в перший біт (і копіюється в біт переносу). Також розрізняють циклічний зсув через біт переносу — при ньому перший біт за напрямом зсуву отримує значення з біта переносу, а значення останнього біта зсувається в біт переносу. У наступній таблиці для деяких мов програмування наведено вбудовані оператори та функції, що реалізують побітові логічні операції.

3. Режими шифрування методом Скремблера?

Скремблювання — шифрування потоку даних, в результаті якої він виглядає як потік випадкових бітів. Послідовність вхідних бітів шифрується за допомогою згенерованої послідовності шифруючих бітів, де імовірність появи одиниці або нуля однакова. Під час шифрування зберігається можливість дешифрувати послідовність. Стосовно до телекомунікаційних систем скремблювання підвищує надійність синхронізації пристроїв, підключених до протилежних сторонах лінії зв'язку, і зменшує рівень перешкод, випромінюваних на сусідні лінії багатожильного кабелю. Інша область застосування скремблерів - захист переданої інформації від несанкціонованого доступу.

4. Поняття гамування та його особливості?

Під гамуванням розуміють процес накладання за певним законом гами шифру на вхідні дані. Гама шифру – це псевдовипадкова послідовність, створена за заданим алгоритмом для шифрування вхідних даних і дешифрування зашифрованих даних.

Процес шифрування полягає в генеруванні гами шифру і накладанні отриманої гами на початковий вхідний текст у зворотний спосіб, наприклад, з використанням операції додавання за модулем 2.

6. Особливості реалізації шифрування методом біграм та біграм з

подвійним квадратом?

На початку XVI століття абат з Німеччини Йоганн Трісемус

запропонував шифрування двох букв одночасно. Шифри, які почали

застосовувати даний підхід отримали назву біграмних. Зазвичай такі шифри використовують таблиці заповнені символами використовуваного алфавіту. Найбільш відомий шифр біграм отримав назву Playfair. Він застосовувався Великобританією в Першій світовій війні. Для використання шифру Трісемуса зазвичай використовувалися таблиці для записування літер абетки й ключове слово. У таблицю спочатку вписувалося по рядках ключове слово, причому повторювані літери відкидалися. Потім ця таблиця доповнювалася літерами абетки, які не увійшли до неї одна за одною.

**Текст програми реалізації:**

package lab2;

import java.util.Arrays;

public class Gamma {

// Буква А Б В Г Ґ Д Е Є Ж З И І

// Код 01 02 03 04 05 06 07 08 09 10 11 12

// Буква Ї Й К Л М Н О П Р С Т У

// Код 13 14 15 16 17 18 19 20 21 22 23 24

// Буква Ф Х Ц Ч Ш Щ Ь Ю Я Пробіл

// Код 25 26 27 28 29 30 31 32 33 34

// Цифра 0 1 2 3 4 5 6 7 8 9

// Код 35 36 37 38 39 40 41 42 43 44

static char[] gSymbol = new char[] {

'А','Б','В','Г','Ґ','Д','Е','Є','Ж','З','И','І',

'Ї','Й','К','Л','М','Н','О','П','Р','С','Т','У',

'Ф','Х','Ц','Ч','Ш','Щ','Ь','Ю','Я','\_',

'0','1','2','3','4','5','6','7','8','9'

};

static int[] gCode = new int[] {

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,

13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,

25, 26, 27, 28, 29, 30, 31, 32, 33, 34,

35, 36, 37, 38, 39, 40, 41, 42, 43, 44

};

// static String gamma = "ТИГР";

// static String message = "ЛЕГІОН\_27";

static String gamma;// = "ВЕЛИЧАР";

static String message;// = "АЯНОМАЙСКІЙ\_200\_Т";

static char[] gammaCh;

static char[] messageCh;

// static String gamma = "ДОБРОСЛАВ";

static String encryptedMessage;// = "ПФХ0ЧГПШ7Й73568ДЧ";

//get gamma code by char

public static int getGammaCodeByCh(final char c) {

final int length = gSymbol.length;

for(int i = 0; i < length; i++) {

if (gSymbol[i] == c) {

return gCode[i];

}

}

return 0;

}

// get gamma char by code

public static char getGammaChByCode(final int c) {

final int length = gSymbol.length;

for(int i = 0; i < length; i++) {

if (gCode[i] == c) {

return gSymbol[i];

}

}

return ' ';

}

// encrypt

public static char[] encrypt(char[] msg) {

int j = 0;

int code;

char[] result = new char[msg.length];

for(int i = 0; i < msg.length; i++) {

code = getGammaCodeByCh(msg[i]) + getGammaCodeByCh(gamma.toCharArray()[j]);

code = code % (gSymbol.length);

result[i] = getGammaChByCode(code);

j++;

if(j == gamma.toCharArray().length)

j = 0;

}

return result;

}

//decrypt

public static char[] decrypt(char[] msg) {

int j = 0;

int code;

char[] result = new char[msg.length];

for(int i = 0; i < msg.length; i++) {

code = getGammaCodeByCh(msg[i]) - getGammaCodeByCh(gamma.toCharArray()[j]) + gSymbol.length;

code = code % (gSymbol.length);

result[i] = getGammaChByCode(code);

j++;

if(j == gamma.toCharArray().length)

j = 0;

}

return result;

}

// print

public static void print(final char[] arr) {

for(int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

}

System.out.println();

}

public static void main(String[] args) {

System.out.println();

gamma = "ВЕЛИЧАР";

message = "АЯНОМАЙСКІЙ\_200\_Т";

gammaCh = gamma.toCharArray();

messageCh = message.toCharArray();

System.out.println("Gamma: "+gamma);

System.out.println("Message: "+message);

// encrypt message

messageCh = encrypt(messageCh);

System.out.print("Encrypted message: ");

print(messageCh);

// messageCh = decrypt(messageCh);

// System.out.print("Decrypted message: ");

// print(messageCh);

System.out.println();

gamma = "ДОБРОСЛАВ";

message = "ПФХ0ЧГПШ7Й73568ДЧ";

gammaCh = gamma.toCharArray();

messageCh = message.toCharArray();

System.out.println("Gamma: "+gamma);

System.out.println("Message: "+message);

//decrypt message

messageCh = decrypt(messageCh);

System.out.print("Decrypted message: ");

print(messageCh);

// messageCh = encrypt(messageCh);

// System.out.print("Encrypted message: ");

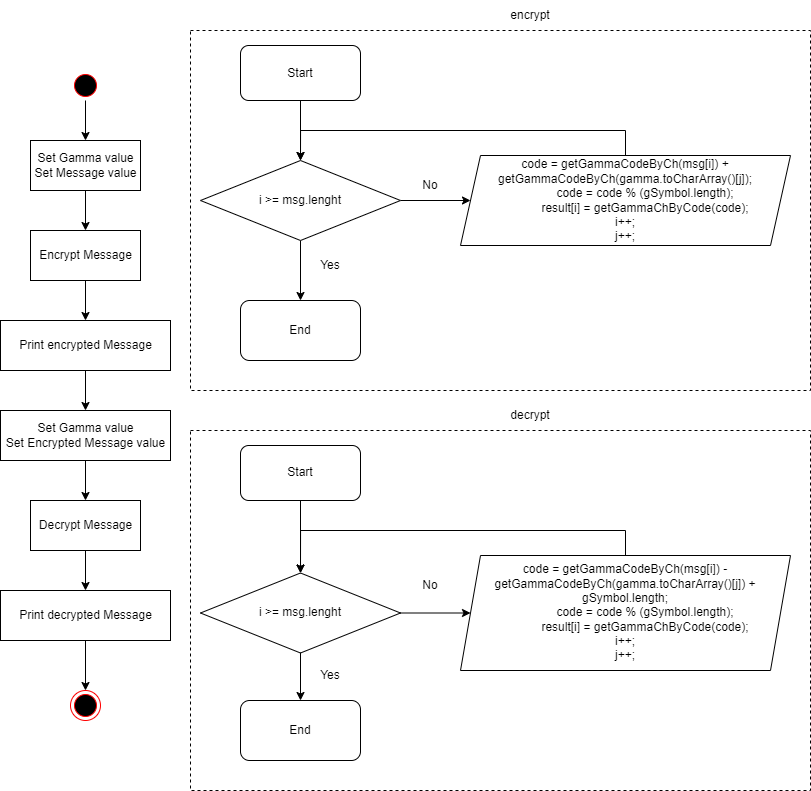
// print(messageCh);

System.out.println();

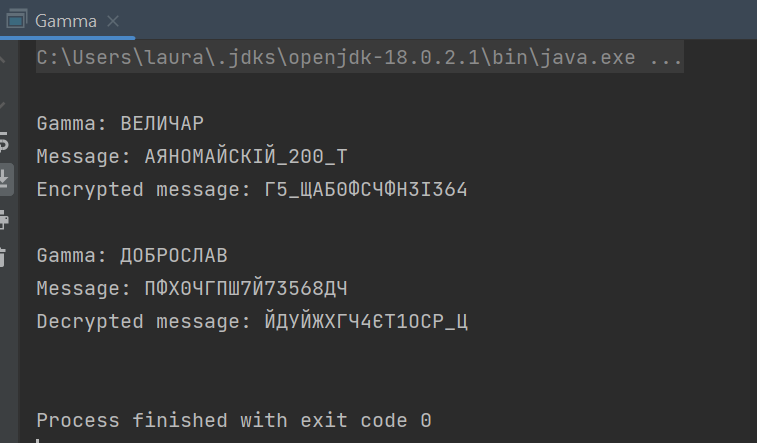
}

}

**Блок-схема:**



**Результати роботи програми:**



**Висновки:**

Навчилася опрацьовувати (шифрувати та дешифрувати) файли з допомогою блокового шифрування інформації, методів моноалфавітної заміни та методу гамування.