# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №7**

# з курсу

**«Безпека мереж і комп’ютерних систем»**

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

проф. Сайко В.Г.

## Київ – 202

**1.Назва роботи**

Дослідження принципів роботи найпростіших алгоритмів шифрування (шифр Цезаря, шифрування з використанням логічної операції XOR). Криптографічний аналіз даних алгоритмів.

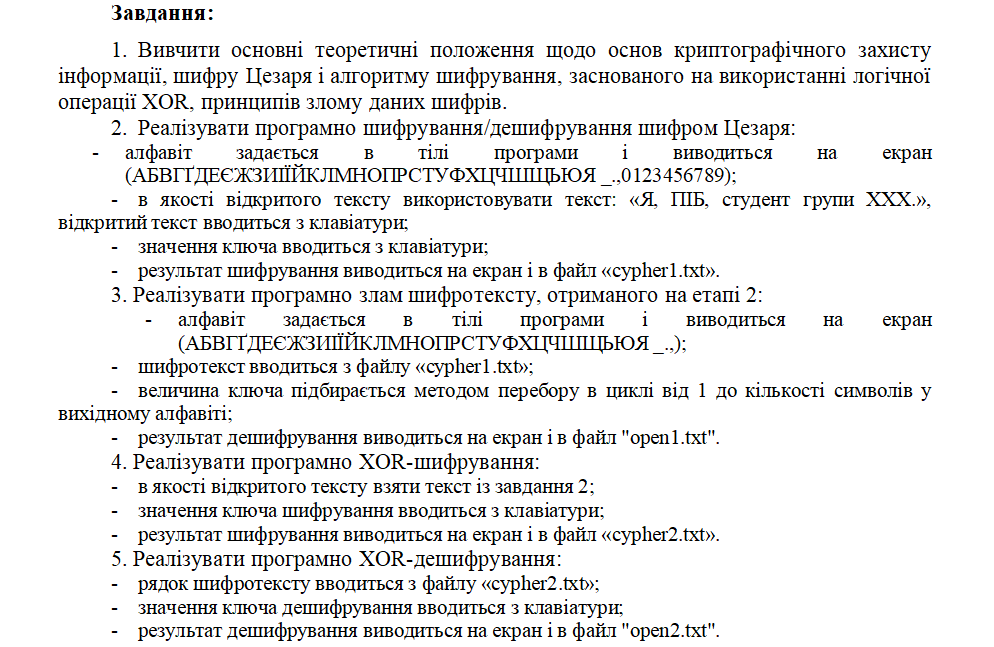
1. **Тема роботи**

Дослідження принципів роботи найпростіших алгоритмів шифрування (шифр Цезаря, шифрування з використанням логічної операції XOR). Криптографічний аналіз даних алгоритмів.

1. **Мета роботи**

Ознайомитися з найпростішими методами криптографічного захисту інформації. Вивчити шифр Цезаря і методи його криптоаналізу. Засвоїти алгоритм шифрування за використанням логічної операції XOR.

1. **Умова завдання**



1. **Рішення**

Код на С#:

using System;

using System.IO;

using System.Text;

namespace Lab7\_caesar\_

{

class Program

{

public static char cipher(char ch, int key)

{

if (!char.IsLetter(ch))

{

return ch;

}

char d = char.IsUpper(ch) ? 'A' : 'a';

return (char)((((ch + key) - d) % 26) + d);

}

public static string Encipher(string input, int key)

{

string output = string.Empty;

foreach (char ch in input)

output += cipher(ch, key);

File.WriteAllTextAsync("encipher.txt", output);

return output;

}

public static string Decipher(string input, int key)

{

File.WriteAllTextAsync("decipher.txt", Encipher(input, 26 - key));

return Encipher(input, 26 - key);

}

public static string Bruteforce(string input)

{

string output = string.Empty;

for (int bruteKey = 0; bruteKey < 26; bruteKey++)

{

foreach (char ch in input)

output += cipher(ch, bruteKey);

}

File.WriteAllTextAsync("bruteforce.txt", output);

return output;

}

public static string XOREncryptOrDecrypt(string text, string key)

{

var result = new StringBuilder();

for (int c = 0; c < text.Length; c++)

result.Append((char)((uint)text[c] ^ (uint)key[c % key.Length]));

File.WriteAllTextAsync("xor.txt", Convert.ToString(result));

return result.ToString();

}

static void Main(string[] args)

{

Console.WriteLine("Type a string to encrypt:");

string UserString = Console.ReadLine();

Console.WriteLine("\n");

Console.Write("Enter your Key");

int key = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("\n");

Console.WriteLine("Encrypted Data");

string cipherText = Encipher(UserString, key);

Console.WriteLine(cipherText);

Console.Write("\n");

Console.WriteLine("Decrypted Data:");

string t = Decipher(cipherText, key);

Console.WriteLine(t);

Console.Write("\n");

string bruteForceCipherText = File.ReadAllText("encipher.txt");

string y = Bruteforce(bruteForceCipherText);

Console.WriteLine(y);

Console.Write("\n");

Console.WriteLine("Enter the key for XOR encryption");

string xorKey = Console.ReadLine();

Console.WriteLine(XOREncryptOrDecrypt(UserString, xorKey));

Console.WriteLine("Enter the key for XOR decryption");

string xorDecryptKey = Console.ReadLine();

string xorEncrypted = File.ReadAllText("xor.txt");

Console.WriteLine(XOREncryptOrDecrypt(xorEncrypted, xorDecryptKey));

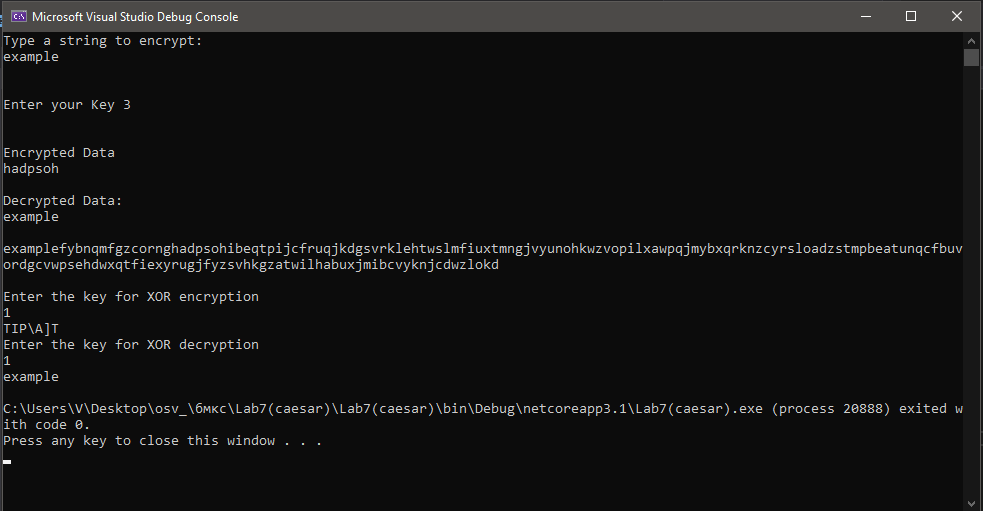
Console.ReadKey();

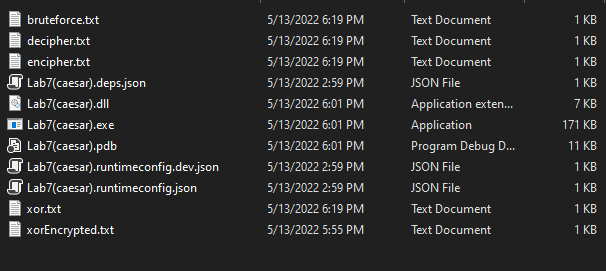
}

}

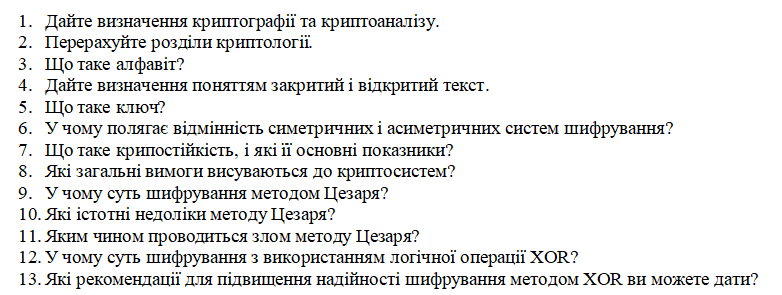
}

Результат роботи:





**Контрольні запитання:**



1. Наука про математичні методи забезпечення конфіденційності, цілісності і автентичності інформації.
2. Розділ криптології, що займається математичними методами порушення конфіденційності і цілісності інформації без знання ключа.
3. Набір символів, з яких скаладається відкритий/закритий текст.
4. Відкритий текст - нешифроване повідомлення; закритий - шифроване
5. Секретний параметр (в ідеалі, відомий лише двом сторонам) для окремого контексту під час передачі повідомлення.
6. У симетричному і відправник, і отримувач мають однаковий ключ. В асиметричному використовують пару ключів - закритий (відомий власнику) та відкритий (відомий усім).
7. Здатність криптографічного алгоритму протистояти криптоаналізу.

Число операцій, необхідних для визначення використаного ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, повинно бути не менше загального числа можливих ключів.

Число операцій, необхідних для розшифрування інформації шляхом перебору всіляких ключів повинно мати строгу нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень).

Знання алгоритму шифрування не повинно впливати на надійність захисту.

Незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні одного і того ж ключа.

Структурні елементи алгоритму шифрування повинні бути незмінними.

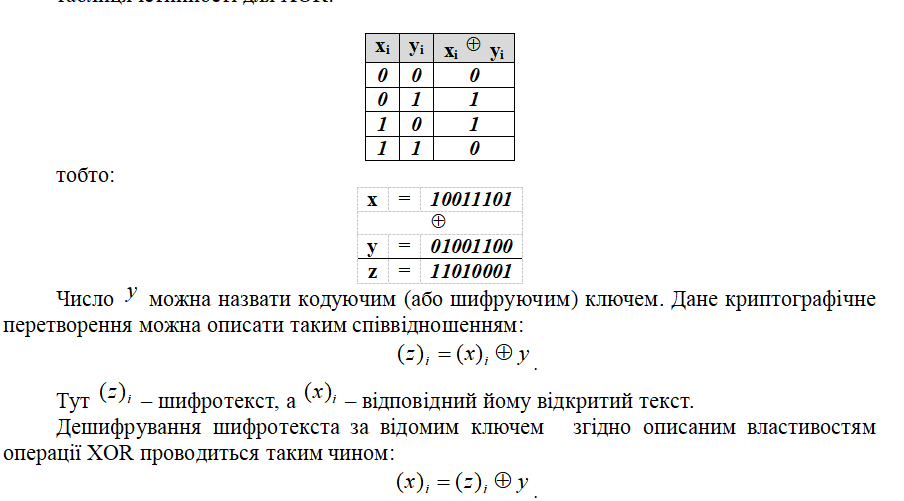
Додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути повністю та надійно сховані в зашифрованому тексті.

Довжина шифрованого тексту повинна бути рівною довжині вихідного тексту.

Не повинно бути простих і легко встановлюваних залежностей між ключами, які послідовно використовуються в процесі шифрування.

Будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації.

Алгоритм повинен допускати як програмно, так і апаратну реалізацію, при цьому зміна довжини ключа не повинно вести до якісного погіршення алгоритму шифрування.

1. Зсув букв в повідомленні на певну кількість символів за алфавітом.
2. Загально відомий, нестійкий.
3. Перебором у циклі зі зміною зсуву
4. 

**Висновки**

В результаті виконання даної лабораторної роботи я ознайомився з найпростішими методами криптографічного захисту інформації, вивчив шифр Цезаря і методи його криптоаналізу, засвов алгоритм шифрування за використанням логічної операції XOR. Створив програмний засіб для шифрування повідомлення введеного з клавіатури, шифром Цезаря та алгоритмом шифрування за використанням логічної операції XOR з подальшим дешифруванням з файлу. Вважаю дану лабораторну роботу виконаною в повному обсязі.