# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №9**

# 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**

Односпрямовані хеш-функції.

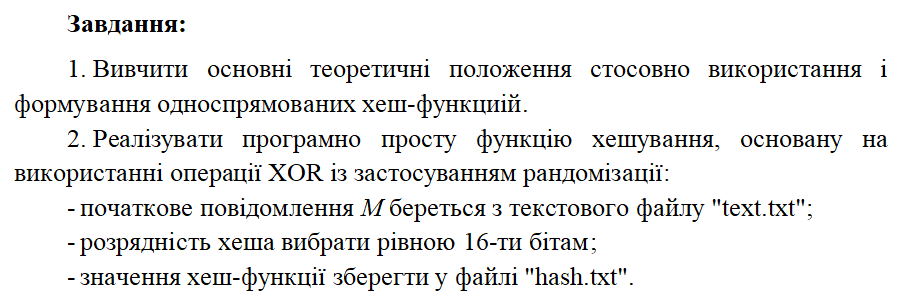
1. **Тема роботи**

Односпрямовані хеш-функції

1. **Мета роботи**

Ознайомитися з різними алгоритмами формування хеш-функцій. Вивчити їх сфери застосування та основні властивості. Розглянути класи атак, спрямованих на аналіз хеш-функцій. Розглянути прості хеш-функції.

1. **Умова завдання**



1. **Рішення**

Код на С#:

using System;

using System.IO;

using System.Text;

namespace Lab9

{

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 Base16Encode(string plainText)

{

var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(plainText);

return System.Convert.ToBase64String(plainTextBytes);

}

public static string Base16Decode(string base64EncodedData)

{

var base64EncodedBytes = System.Convert.FromBase64String(base64EncodedData);

return System.Text.Encoding.UTF8.GetString(base64EncodedBytes);

}

static void Main(string[] args)

{

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

string UserString = Console.ReadLine();

File.WriteAllText("plain.txt", UserString);

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");

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

Console.WriteLine(Base16Encode(plainTextForHash));

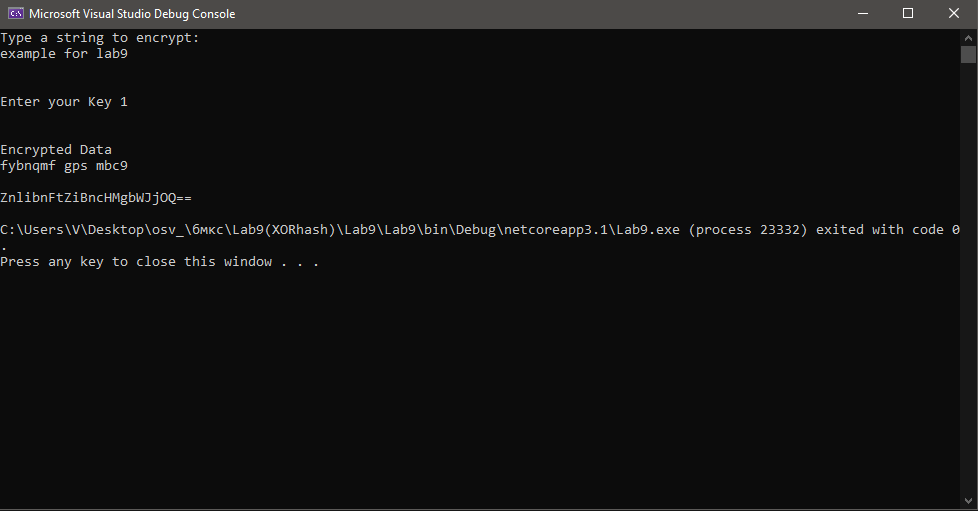
File.WriteAllText("hashed.txt", Base16Encode(plainTextForHash));

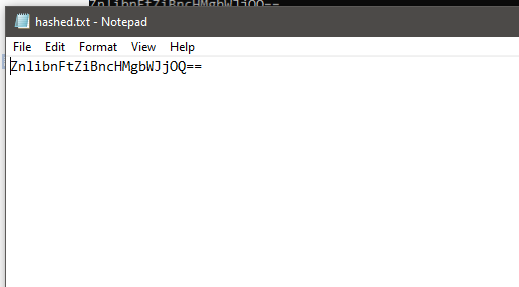
}

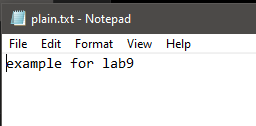
}

}

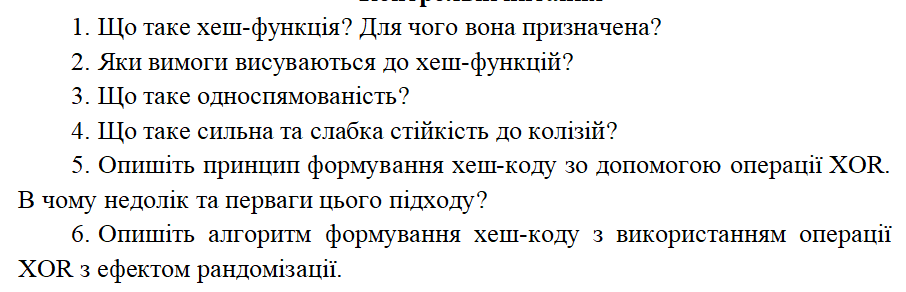
Результат роботи:







**Контрольні запитання:**



1. [Функція](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0" \o "Підпрограма), що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру.
2. Швидкість обчислення, мінімізація колізій.
3. Односпрямована функція - це F(x), де:

Існує поліноміальний алгоритм для обчислення F(x), але не існує поліноміального алгоритму для інвертування F(x) = y

1. Хеш функція – це деяка функція h(K), яка бере ключ K і повертає адресу, по якому проводиться пошук в хеш-таблиці, щоб отримати інформацію, пов'язану з K. Колізія — це ситуація, коли h(K1) = h(K2)
2. На вхід функція отримує слово W  що складається з n символів, кожен розміром 1 байт, і повертає значення в діапазоні від 0 до 255. При цьому значення геш-коду залежить від кожного символу вхідного слова.

Алгоритм можна описати таким псевдокодом, який отримує на вхід рядок W та використовує таблицю перестановок Т:

h := 0

For each c in W loop

index := h xor c

h := T [index]

End loop'

Return h

Переваги - простота обчислення; не існує таких вхідних даних, для яких імовірність колізії найбільша; можливість модифікації в ідеальну геш-функцію, але вона неефективна для захисту даних, коли з відкритим повідомленням передається шифрований хеш-код. Маючи деяке повідомлення, зовсім неважко створити нове повідомлення, якому відповідатиме той же самий хеш-код: просто підготуйте будь-яке необхідне альтернативне повідомлення і приєднаєте до нього відповідний -бітовий блок, який разом з новим повідомленням сформує бажаний хеш-код.

1. Початкова ініціалізація -бітового значення функції хешування нульовим значенням.

Послідовна обробка -бітових блоків даних за наступним правилом:

виконання циклічного зсуву поточного значення функції хешування вліво на один біт;

додавання поточного блоку до значення функції хешування з допомогою операції XOR.

**Висновки**

В результаті виконання даної лабораторної роботи я ознайомився з різними алгоритмами формування хеш-функцій. Вивчив їх сфери застосування та основні властивості, розглянув класи атак, спрямованих на аналіз хеш-функцій, розглянув прості хеш-функції. Створив програмний засіб для хешування повідомлення, записаного у текстовий файл з подальшим виводом результату роботи хеш-функції в інший текстовий файл. Вважаю дану лабораторну роботу виконаною в повному обсязі.