КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №8

З курсу

«Безпека мереж і комп’ютерних систем»

студента 2 курсу
групи ПП-22
спеціальності 122 «Комп'ютерні науки»
ОП «Прикладне програмування»
Шевлюк Вікторії Віталіївни

Перевірив:
д.т.н, професор
Сайко В. Г.

Київ 2022

Тема: Односпрямовані хеш-функції.

Мета: 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.

Завдання:

1. Вивчити основні теоретичні положення стосовно використання і формування односпрямованих хеш-функциій.

2. Реалізувати програмно просту функцію хешування, основану на використанні операції XOR із застосуванням рандомізації:

- початкове повідомлення *M* береться з текстового файлу "text.txt";
- розрядність хеша вибрати рівною 16-ти бітам;
- значення хеш-функції зберегти у файлі "hash.txt".

Хід роботи:

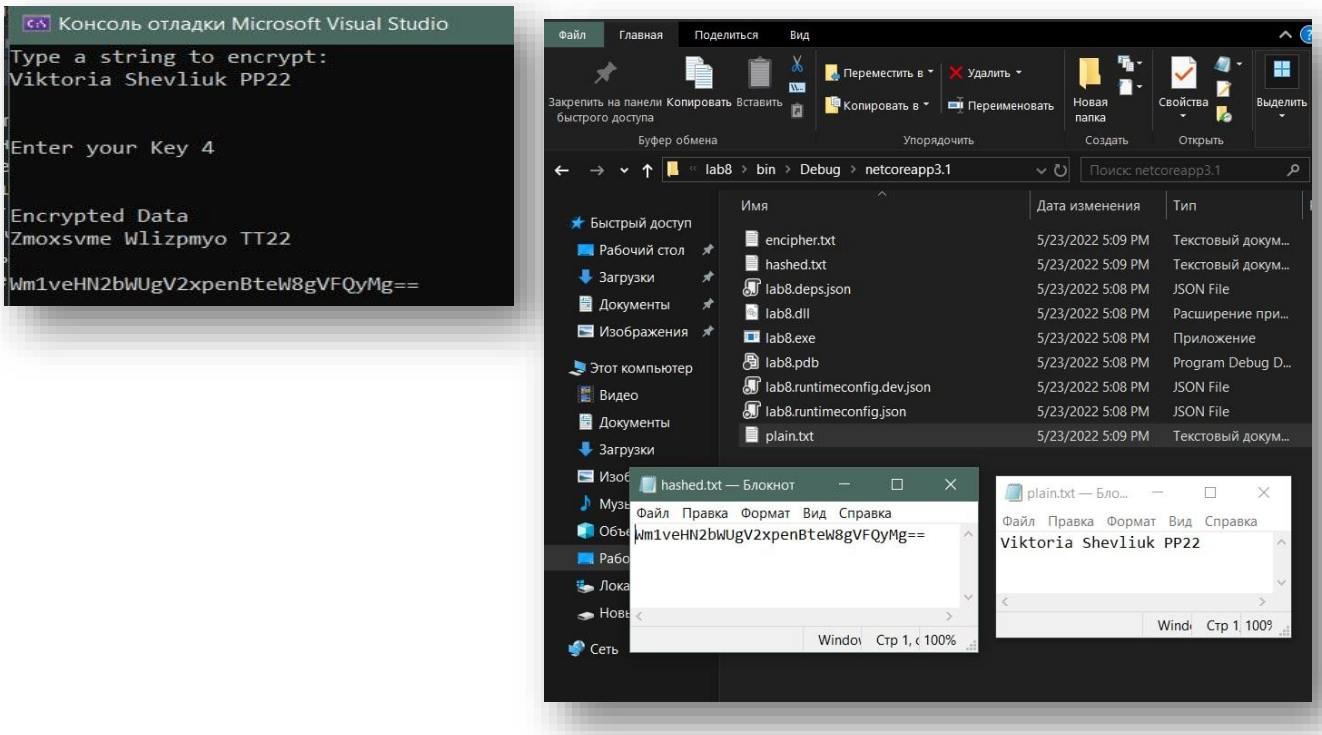
Створимо мовою C# програму, що буде реалізовувати функцію хешування, засновану на операції Ксор та рандомізаці.

Ось так виглядає код програми:

```
using System;
using System.IO;
using System.Text;


namespace lab8
{
    Ссылок: 0
    class Program
    {

        ссылка: 1
        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);


        }

        ссылка: 1
        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;
        }

        Ссылок: 2
        public static string Base16Encode(string plainText)
        {
            var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(plainText);
            return System.Convert.ToBase64String(plainTextBytes);
        }
```

```
Ссылок: 0
public static string Base16Decode(string base64EncodedData)
{
    var base64EncodedBytes = System.Convert.FromBase64String(base64EncodedData);
    return System.Text.Encoding.UTF8.GetString(base64EncodedBytes);
}


Ссылок: 0
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 програми:



Висновок: у ході даної лабораторної роботи я ознайомилася з різними алгоритмами формування хеш-функцій та простими хеш-функціями. Хеш-функція— функція, що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру. Хешування— перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Такі перетворення також називаються хеш-функціями, або функціями згортання, а їхні результати називають хешем, хеш-кодом, хеш-сумою, або дайджестом повідомлення Хеш-функція використовується зокрема у структурах даних — хеш-таблицях, широко вживаних у програмному забезпеченні для швидкого пошуку даних. Хеш-функції використовуються для оптимізації таблиць та баз даних користуючись з того, що в однакових записів однакові значення хеш-функції.

Контрольні питання:

1. ***Що таке хеш-функція? Для чого вона призначена?***

Функція, що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру.

2. ***Яки вимоги висуваються до хеш-функцій?***

Швидкість обчислення, мінімізація колізій

3. ***Що таке односпямованість?***

Односпрямована функція - це F(x), де:

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

4. ***Що таке сильна та слабка стійкість до колізій?***

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)

5. Опишіть принцип формування хеш-коду зо допомогою операції XOR. В чому недолік та переваги цього підходу?

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 T.

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 n-бітовий блок, який разом з новим повідомленням сформує бажаний хеш-код.

6. Опишіть алгоритм формування хеш-коду з використанням операції XOR з ефектом рандомізації.

Початкова ініціалізація n-бітового значення функції хешування нульовим значенням. Послідовна обробка n-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.