Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

# Лабораторна робота №6

з дисципліни «Безпека інформаційних систем»

«Шифрування з відкритим ключем на основі задачі рюкзака»

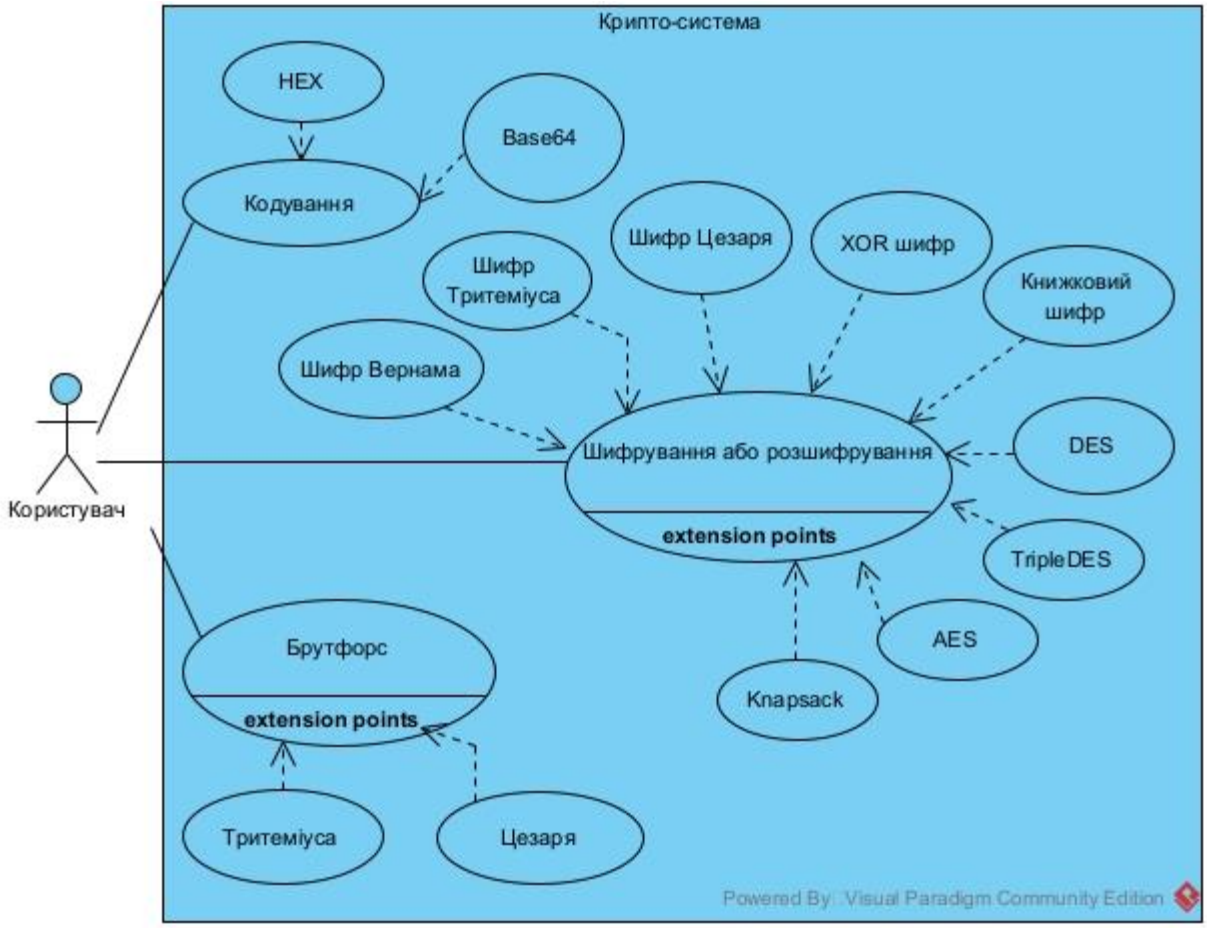
Варіант № 22

Виконав: Студент групи ТР-12
Ковальов Олександр
Перевірив: доцент, к.ф.-м.н.
Тарнавський Ю. А.

КИЇВ 2023

**Мета роботи.** Ознайомитись з принципами побудови асиметричних криптосистем.

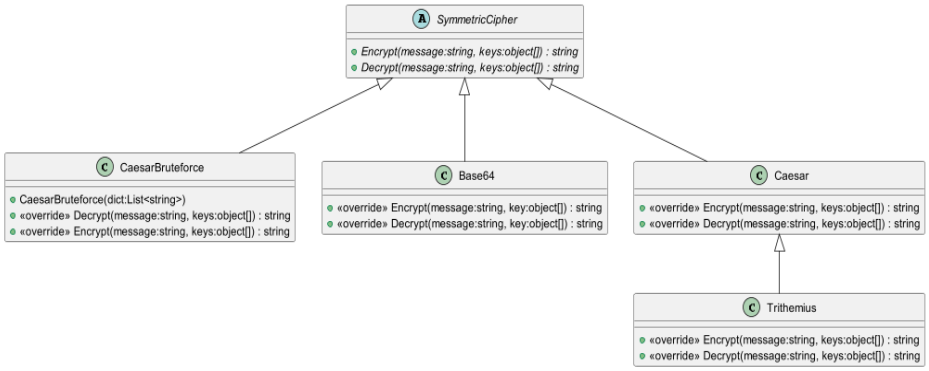
**Діаграма прецедентів.**




**Діаграма класів.**

В просторі імен Cryptography знаходяться всі шифри та супутні до них алгоритми. Також, там є перелік CipherEnum. Клас Knapsack знаходиться в просторі імен Cryptography.Asymmetric. Алгоритми (Евкліда, рюкзака) знаходяться в просторі імен Algorithms.

В класі Knapsack знаходяться основні методи для шифрування та розшифрування даних цим методом. API класу складається з двох основних методів – Encrypt та Decrypt. В них викликаються приватні методи. Наприклад, там ще є методи для генерації секретного та публічного ключа.

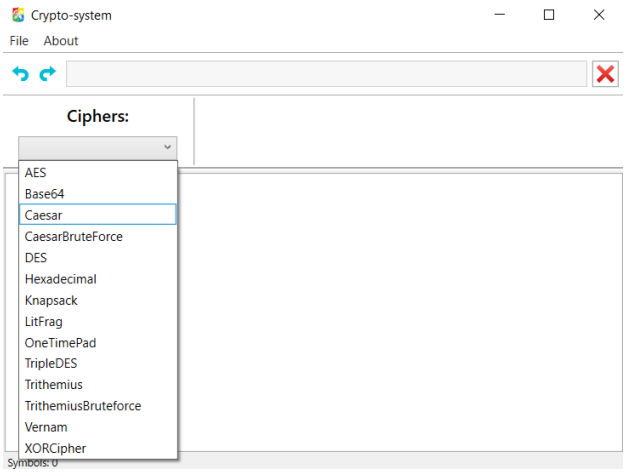


## Фрагмент коду з реалізацією алгоритму шифрування/розшифрування.

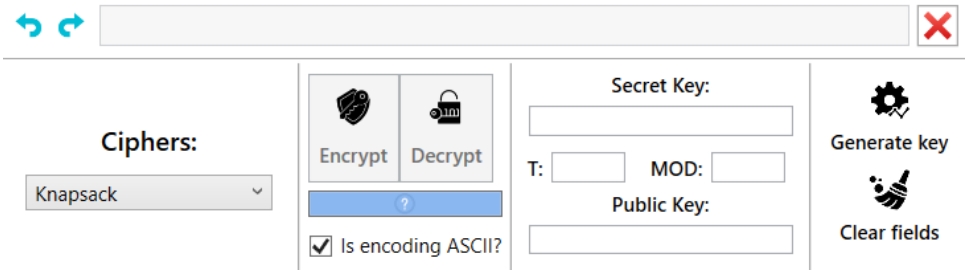
```
public string Encrypt(string message, long[] publicKey, bool isASCII)
    {
        var binary = BinaryConverter.StringToBinary(message,
            bits: isASCII ? 7 : 16);

        var padding = new string('0', publicKey.Length - (binary.Length % publicKey.Length));
        binary = string.Concat(binary, padding);

        var bitGroupLength = publicKey.Length;

        var encrypted = new long[binary.Length / publicKey.Length];

        for (int i = 0; i < encrypted.Length; i++)
        {
            var bitGroup = binary.Substring(i * bitGroupLength, bitGroupLength);
            long sum = 0;

            for (int j = 0; j < bitGroupLength; j++)
            {
                var bit = (int) char.GetNumericValue(bitGroup[j]);

                if (bit == 1) sum += publicKey[j];
            }

            encrypted[i] = sum;
        }

        return string.Join(", ", encrypted);
    }

    public string Decrypt(long[] encryptedSequence, long[] secretSequence, long tInverse, long mod,
bool isASCII)
    {
        var decryptedSequence = new long[encryptedSequence.Length];

        for (int i = 0; i < encryptedSequence.Length; i++)
        {
            decryptedSequence[i] = (encryptedSequence[i] * tInverse) % mod;
        }

        var sb = new StringBuilder();
        foreach (var num in decryptedSequence)
        {
            sb.Append(KnapsackAlg.GetBytes(num, secretSequence));
        }

        var result = BinaryConverter.BinaryToString(sb.ToString(),
            bits: isASCII ? 7 : 16);

        return result;
    }
```

## Скріншоти програми.

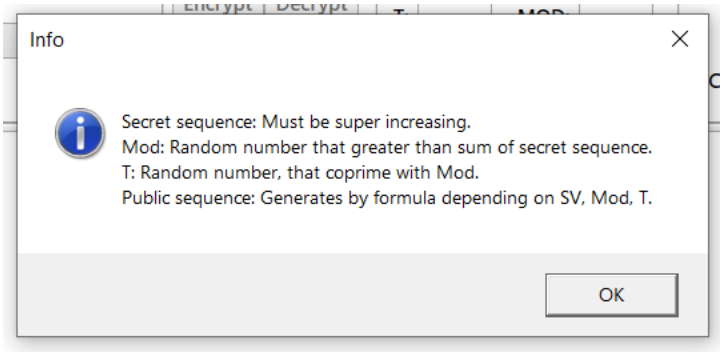
Головне вікно:



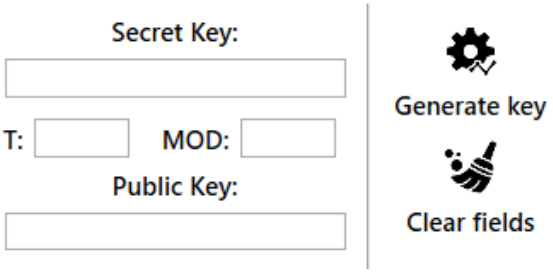
**Knapsack.** Головна панель має такий вигляд:



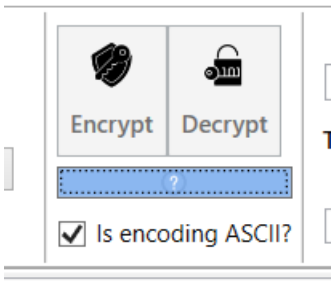

Якщо користувач не знає як користуватись шифром, то може натиснути на кнопку “Info” – вона знаходиться під кнопками для шифрування. Там він може дізнатись, які дані потрібно вводити в поля.



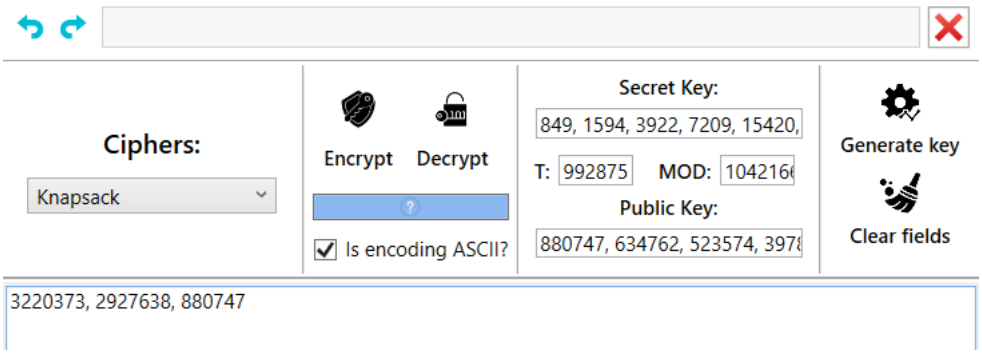

Наявні поля – секретний ключ (секретна послідовність, T, модуль) та публічний ключ – послідовність. Їх можна як вписати, і тоді останні дані згенеруються залежно від того що вже є, або все вписати самому. Для того щоб очистити поля, потрібно натиснути на “Clear fields”.



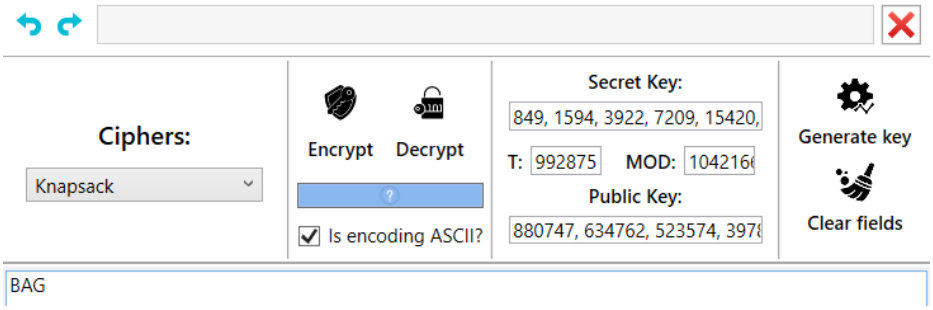

Також, результат різні відносно того, в якому кодуванні записане повідомлення. Для визначення встановлений CheckBox – якщо кодування ASCII, то потрібно встановити галочку.



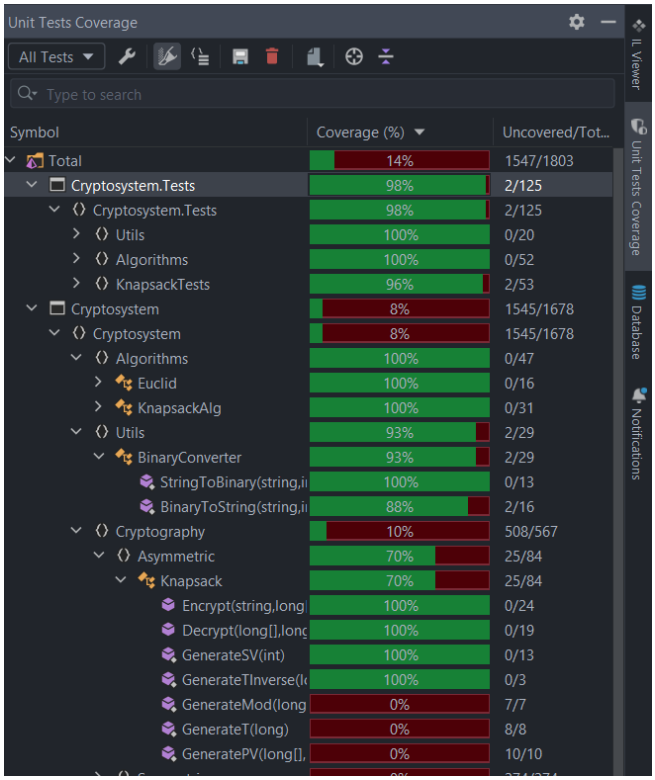
Приклад зашифрованого повідомлення:




Приклад розшифрованого повідомлення:




Новий код частково покритий юніт-тестами:




**Висновок:** за результатами виконання цієї лабораторної роботи було ознайомлено з принципом роботи асиметричних шифрів. В крипто-систему був імплементований шифр за алгоритмом рюкзака.