

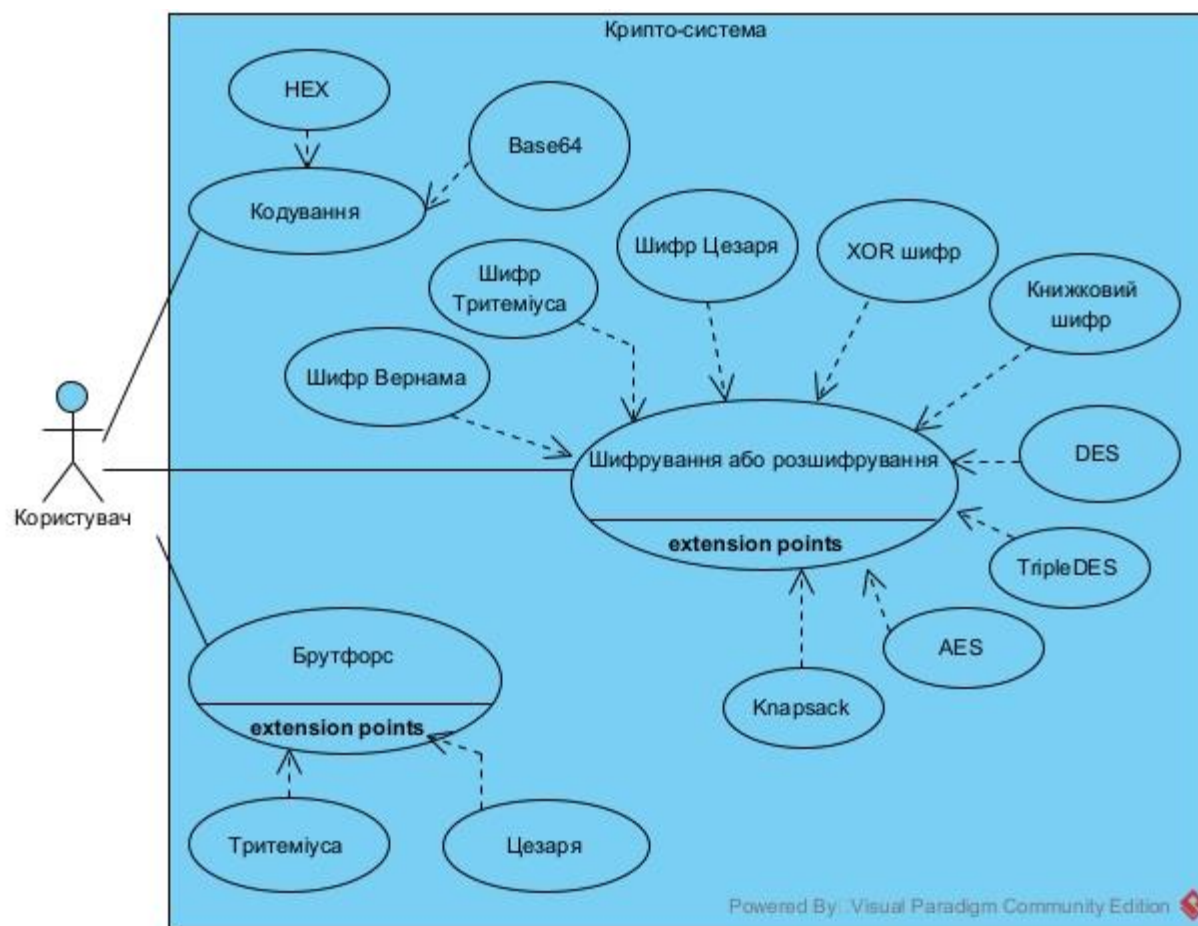
Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

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
Перевірів: доцент, к.ф.-м.н.
Тарнавський Ю. А.

Мета роботи. Ознайомитись з принципами побудови асиметричних криптосистем.

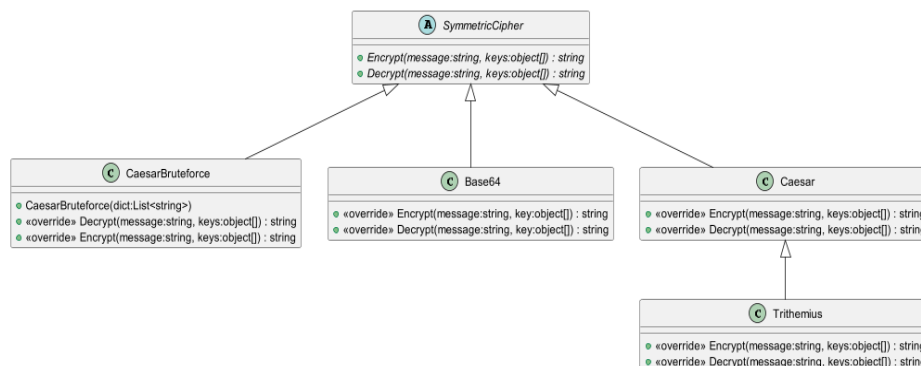
Діаграма прецедентів.



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;
    }

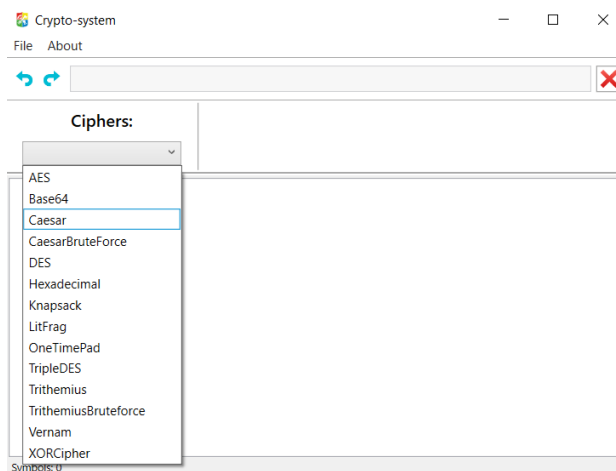
    var sb = new StringBuilder();
    foreach (var num in decryptedSequence)
    {
        sb.Append(KnapsackAlg.GetBytes(num, secretSequence));
    }

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

    return result;
}
```

Скріншоти програми.

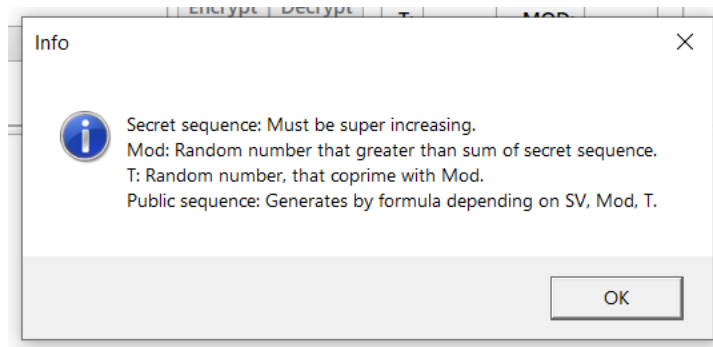
Головне вікно:



Knapsack. Головна панель має такий вигляд:

The main panel of the Knapsack application features a top bar with a refresh icon, a text input field, and a close button. Below this, the interface is divided into several sections: a 'Ciphers' dropdown menu currently set to 'Knapsack'; a central area with 'Encrypt' and 'Decrypt' buttons, a question mark icon, and a checked checkbox labeled 'Is encoding ASCII?'; a 'Secret Key' section with a text input field and 'T:' and 'MOD:' labels; a 'Public Key' section with a text input field; and a right-hand sidebar with 'Generate key' and 'Clear fields' buttons, each accompanied by an icon.

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, Т, 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”.

A close-up view of the key input section of the application. It shows the 'Secret Key' label above a text input field, followed by 'T:' and 'MOD:' labels above their respective input fields. Below these is the 'Public Key' label above another text input field. To the right of these fields are two buttons: 'Generate key' with a gear icon and 'Clear fields' with an eraser icon.

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

A close-up view of the 'Is encoding ASCII?' checkbox, which is currently checked. It is located below the 'Encrypt' and 'Decrypt' buttons and above the 'Public Key' input field.

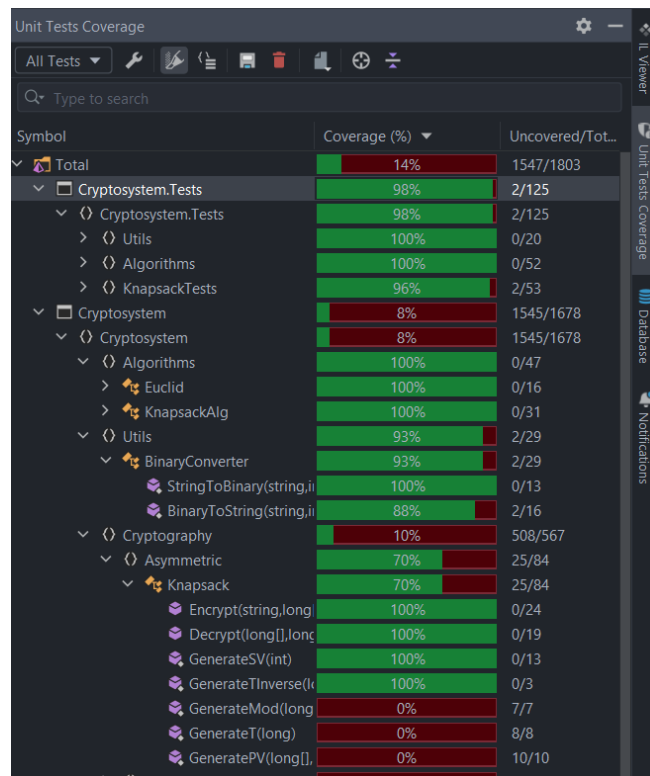
Приклад зашифрованого повідомлення:

The screenshot shows a web-based cryptographic application. At the top, there is a text input field containing the encrypted message "3220373, 2927638, 880747". Below this, the interface is divided into several sections. On the left, under "Ciphers:", a dropdown menu shows "Knapsack". In the center, there are buttons for "Encrypt" and "Decrypt", with "Decrypt" being active. Below these buttons is a checkbox labeled "Is encoding ASCII?" which is checked. On the right, there are fields for "Secret Key:" (849, 1594, 3922, 7209, 15420), "T:" (992875), "MOD:" (104216), and "Public Key:" (880747, 634762, 523574, 3978). Further right are buttons for "Generate key" and "Clear fields". At the bottom, a large text area displays the encrypted message "3220373, 2927638, 880747".

Приклад розшифрованого повідомлення:

This screenshot shows the same application as the previous one, but with the "Decrypt" button active. The text input field at the top is empty. The "Is encoding ASCII?" checkbox remains checked. The key fields and buttons on the right are the same. The large text area at the bottom now displays the decrypted message "BAG".

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.