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

Лабораторна робота №2

з дисципліни «Безпека інформаційних систем»

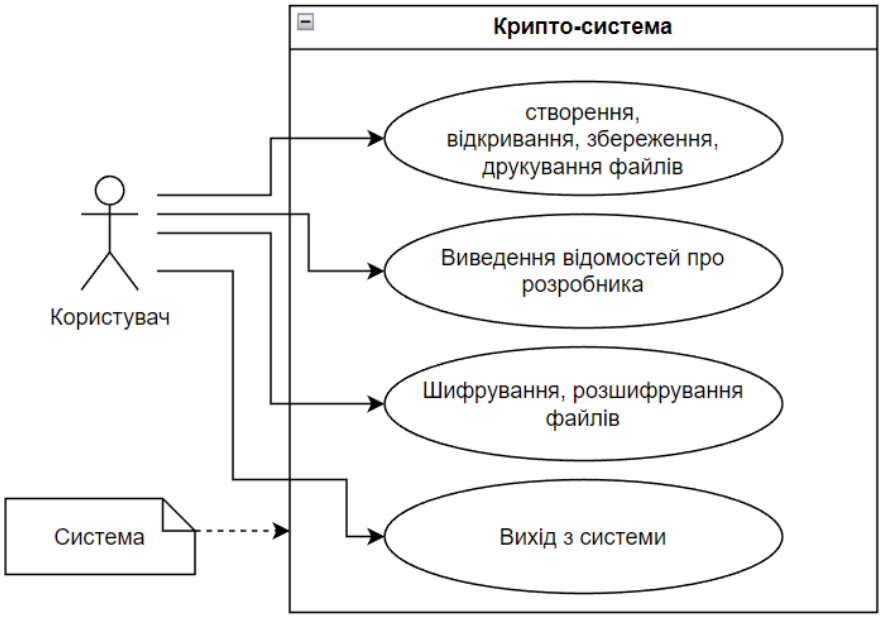
«Шифр Тритеміуса»

Варіант № 22

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

КИЇВ 2023

Мета роботи. Розробити криптосистему на основі шифру Тритеміуса.
Діаграма прецедентів.

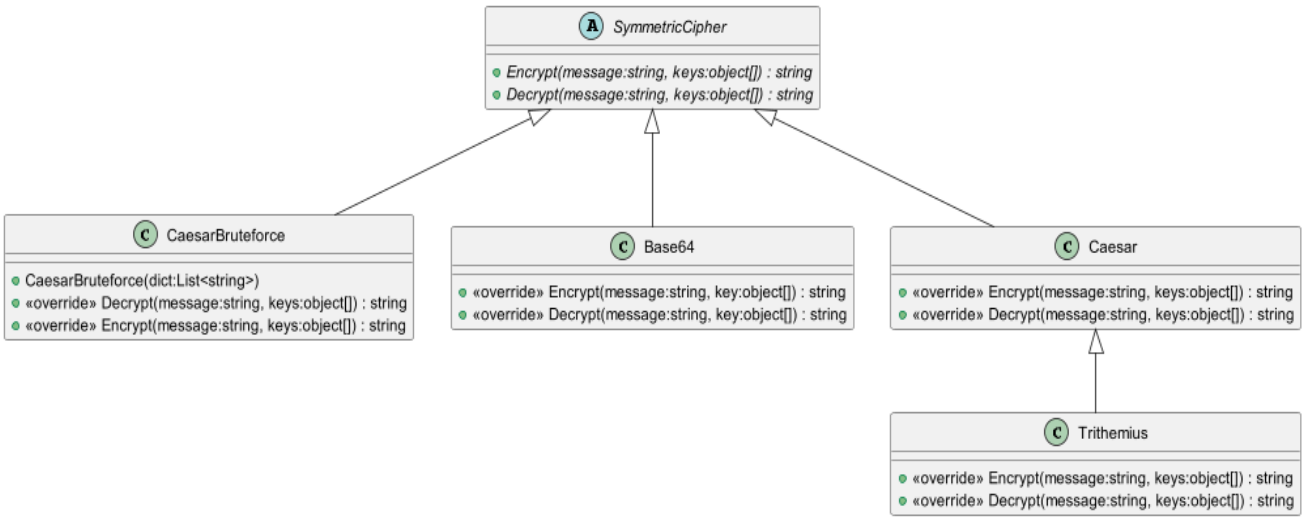


Діаграма класів.

В просторі імен Cryptography знаходяться всі шифри, які наслідуються від класу SymmetricCipher. Також там є перелік CipherEnum.

В класі Trithemius знаходяться основні методи для шифрування та розшифрування даних цим методом. Клас наслідується від класу Caesar, а той, відповідно, від класу SymmetricCipher. Це означає, що API класу складається з двох основних методів – Encrypt та Decrypt. В них викликаються приватні методи.

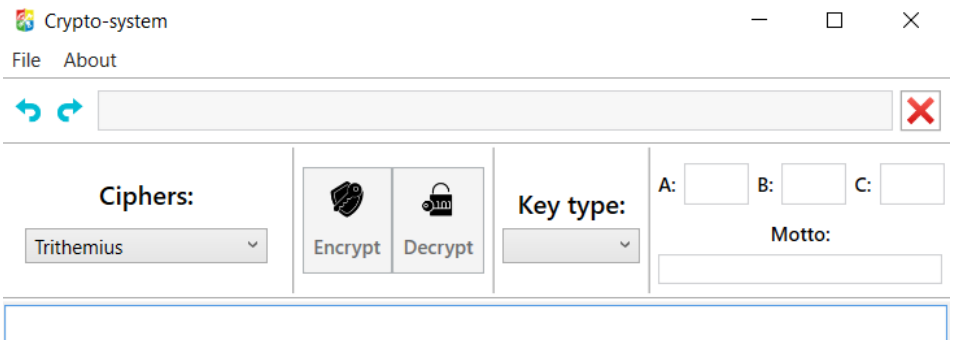
Методи приймають аргументи: повідомлення типу String, масив типу object[] keys. Перший аргумент – власне повідомлення, яке треба зашифрувати або розшифрувати. Друге – масив, помічений ключовим словом params. Це означає, що можна передавати будь-яку кількість аргументів методу. Вони автоматично запакуються в масив.



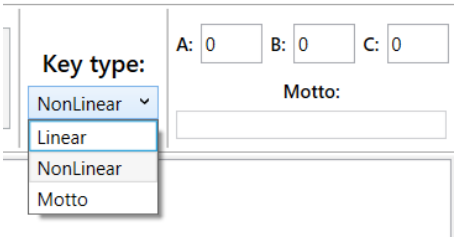
Фрагмент коду з реалізацією алгоритму шифрування/розшифрування.

```
public override string Encrypt(string message, params object[] keys)
    {
        Func<int, int> handler = ValidateAndGetHandler(message, keys);

        return TrithemiusCipher(message, handler);
    }

    public override string Decrypt(string message, params object[] keys)
    {
        Func<int, int> handler = ValidateAndGetHandler(message, keys);

        return TrithemiusCipher(message, DecryptionHandler(handler));
    }

    private string TrithemiusCipher(string message, Func<int, int> handler)
    {
        var sb = new StringBuilder();

        for (int i = 0; i < message.Length; i++)
        {
            char c = message[i];

            sb.Append((char) ((c + (char) handler(i)) % UnicodeCardinal));
        }

        return sb.ToString();
    }

    private Func<int, int> LinearHandler(int a, int b)
    {
        return (int position) => a * position + b;
    }

    private Func<int, int> NonLinearHandler(int a, int b, int c)
    {
        return (int position) => (a * position * position + b * position + c);
    }


    private Func<int, int> MottoHandler(string message, string motto)
    {
        return (int position) =>
        {
            if (message.Length <= motto.Length) return motto[position];

            var factor =
(int) Math.Ceiling((decimal) message.Length / motto.Length);

            var localMotto = string.Concat(Enumerable.Repeat(motto, factor));

            return localMotto[position];
        };
    }

    private Func<int, int> DecryptionHandler(Func<int, int> handler)
    {
        return (int position) => -1 * handler(position);
    }
```

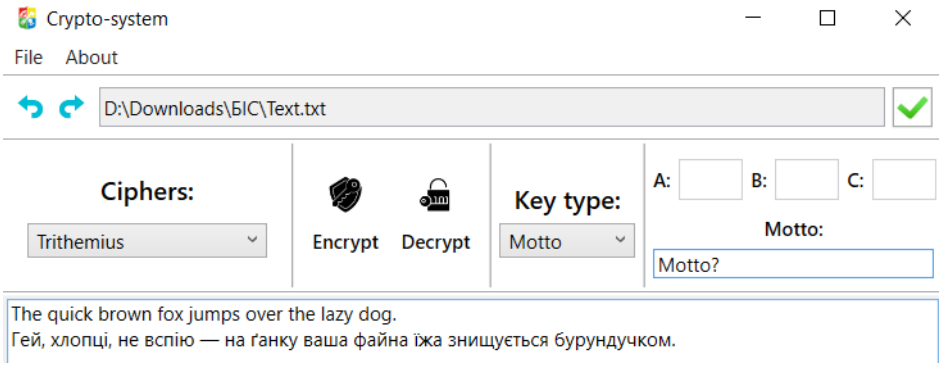
Скріншоти програми.
Панель для роботи з шифром Тритеміуса:



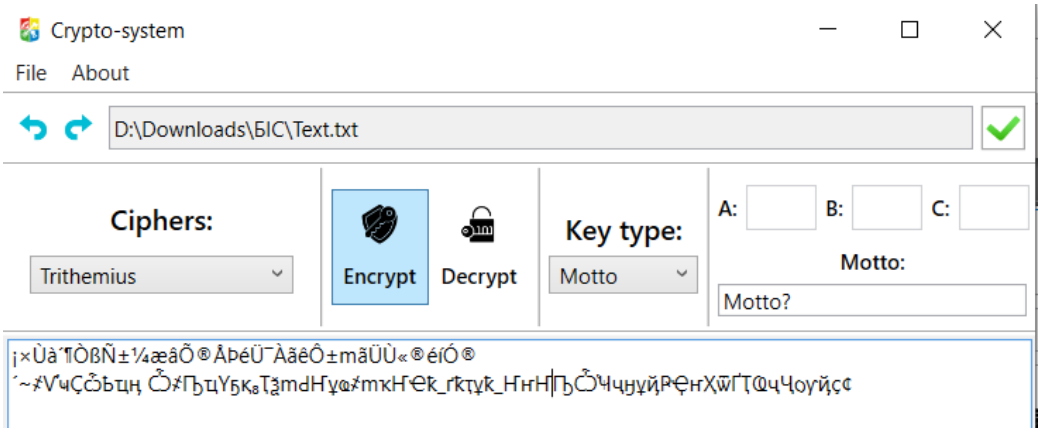
Всього є 3 типи ключів: лінійне рівняння, нелінійне та гасло:



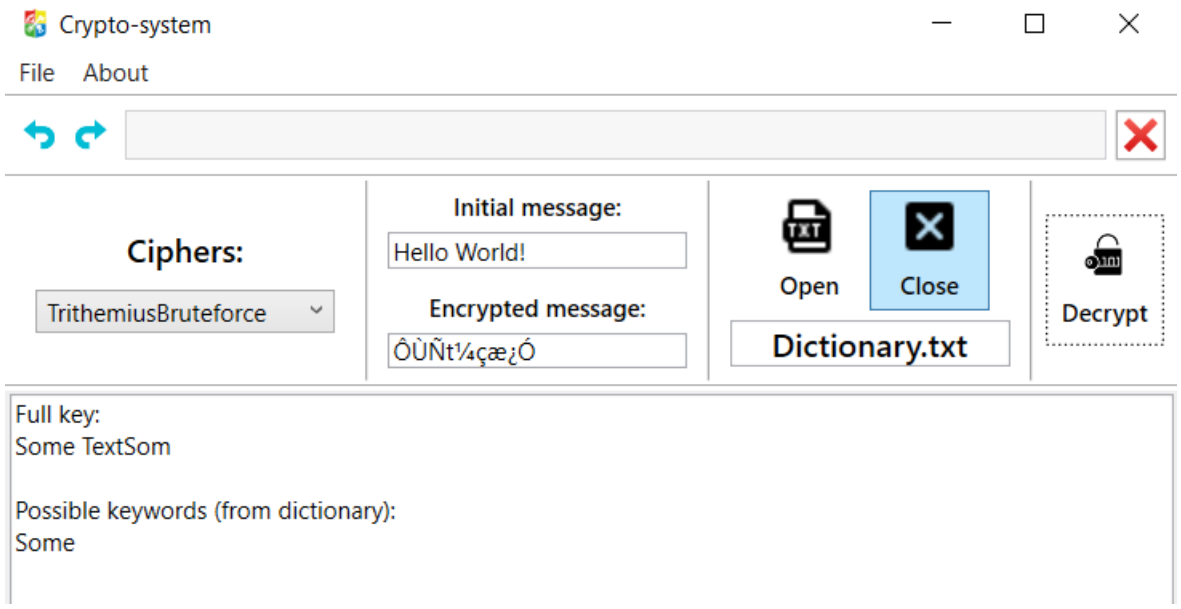
Кнопки Encrypt та Decrypt розблоковуються, коли визначений тип ключа, в робочій області введено текст, та заповнені поля пов’язані з ключем:



Приклад зашифрованого тексту:



Приклад роботи методу грубої сили для шифру Тритеміуса:



Висновок: за результатами виконання цієї лабораторної роботи було ознайомлено з принципом роботи шифру Тритеміуса та написано метод грубої сили. Але – брутфорс працює лише якщо перехватити пару повідомлень «Зашифроване» та «Розшифроване». Також, метод не працює з ключами у вигляді лінійних та нелінійних рівнянь. Але, це і не має сенсу – отримання ключа дозволило б розшифровувати повідомлення такої ж довжини, не більше (тому що далі послідовність інша, бо ключ без повторень)