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

Лабораторна робота №5

з дисципліни «Безпека інформаційних систем»

«Шифр DES»

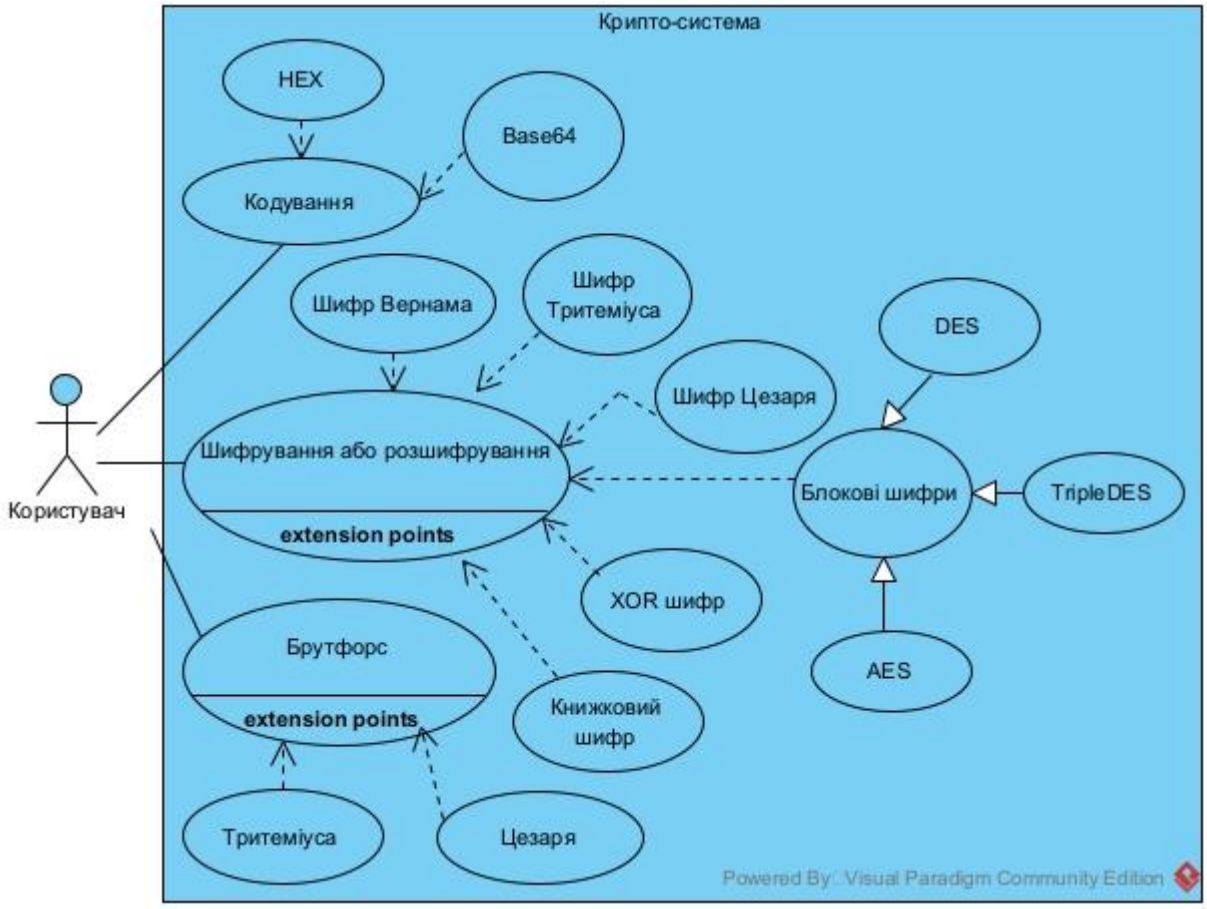
Варіант № 22

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

КИЇВ 2023

Мета роботи. Ознайомитись з використанням криптопровайдерів в прикладному програмуванні.

Діаграма прецедентів.

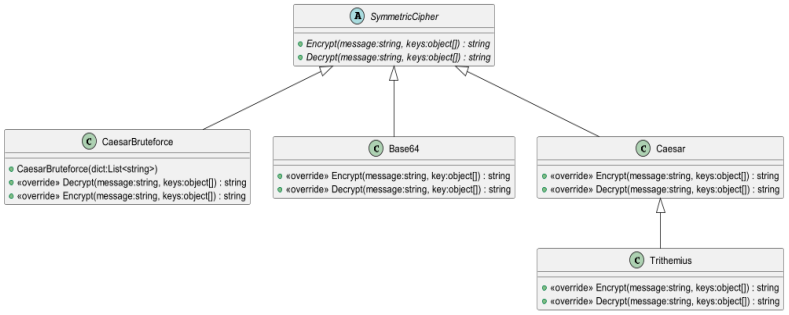


Діаграма класів.

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

В класах DES, AES, TripleDES знаходяться основні методи для шифрування та розшифрування даних цим методом. Клас наслідується від класу SymmetricCipher. Це означає, що API класу складається з двох основних методів – Encrypt та Decrypt. В них викликаються приватні методи.

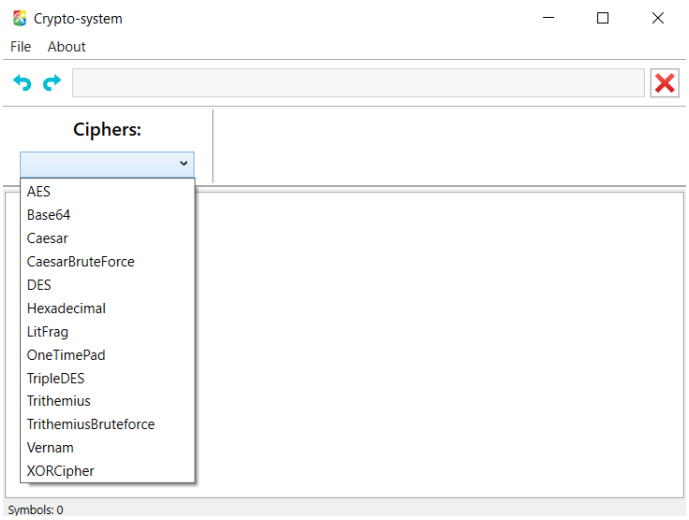
Методи приймають аргументи: повідомлення типу String, масив типу object[] keys. Перший аргумент – повідомлення, яке треба зашифрувати або розшифрувати. Друге – масив, помічений ключовим словом params. Це означає, що методу можна передавати будь-яку кількість аргументів. Вони автоматично запакуються в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



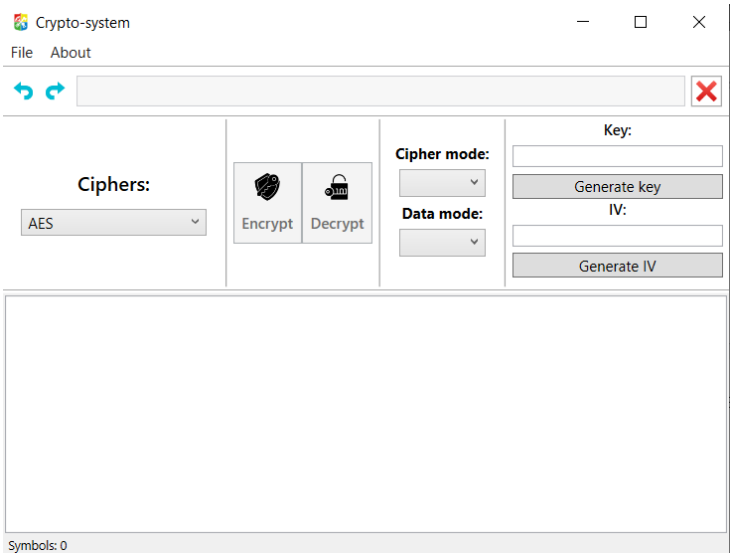
Фрагмент коду з реалізацією алгоритму шифрування/розшифрування.

```
public string Encrypt(CipherEnum type, string message, params object[] keys)
    {
        var provider = CryptoProvider(type, keys);

        var encryptor = provider.CreateEncryptor(provider.Key, provider.IV);

        var memoryStream = new MemoryStream();
        var cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write);

        using (var writer = new StreamWriter(cryptoStream))
        {
            writer.Write(message);
        }

        var encrypted = memoryStream.ToArray();

        cryptoStream.Close();
        memoryStream.Close();

        var dataMode = ValidateCastDataMode(keys[0]);

        return dataMode is DataMode.Base64 ?
            Convert.ToBase64String(encrypted)
            : Convert.ToHexString(encrypted);
    }

    public string Decrypt(CipherEnum type, string message, params object[] keys)
    {
        var provider = CryptoProvider(type, keys);

        var dataMode = ValidateCastDataMode(keys[0]);

        var bytes = dataMode is DataMode.Base64 ?
            Convert.FromBase64String(message)
            : Convert.FromHexString(message);

        var decryptor = provider.CreateDecryptor(provider.Key, provider.IV);

        var memoryStream = new MemoryStream(bytes);
        var cryptoStream = new CryptoStream(memoryStream, decryptor, CryptoStreamMode.Read);

        string plaintext;
        using (var reader = new StreamReader(cryptoStream))
        {
            plaintext = reader.ReadToEnd();
        }

        cryptoStream.Close();
        memoryStream.Close();

        return plaintext;
    }

    private SymmetricAlgorithm CryptoProvider(CipherEnum type, object[] keys)
    {
        if (keys.Length < 3) throw new ArgumentException("Not enough arguments!");

        SymmetricAlgorithm cipher = type switch
        {
            CipherEnum.AES => SysAES.Create(),
            CipherEnum.DES => SysDES.Create(),
            CipherEnum.TripleDES => SysTripleDES.Create(),

            _ => throw new ArgumentException("Wrong cipher type")
        };

        cipher.Mode = ValidateCastCipherMode(keys[1], keys.Length);

        cipher.Key = Encoding.UTF8.GetBytes(ValidateCastKey(keys[2], type));
        cipher.IV = Encoding.UTF8.GetBytes(ValidateCastIV(keys[3]));

        if (type is CipherEnum.DES && cipher.Mode is CipherMode.CFB) cipher.FeedbackSize = 8;

        return cipher;
    }
```

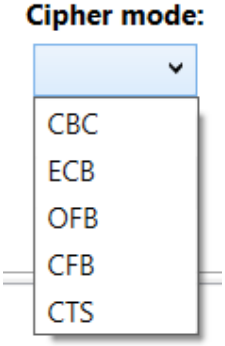
Скріншоти програми.
Головне вікно:



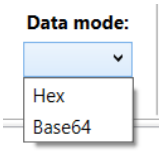
AES, DES, TripleDES. Є дві кнопки – шифрування та розшифрування. Вони заблоковані, якщо всі інші поля порожні:



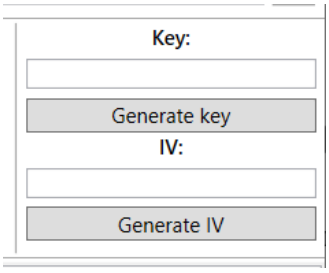
Для трьох шифрів є 5 режимів шифрування – ECB, CBC, CFB, OFB, та CTS. Але працюють лише три – окрім OFB та CTS. Ці методи не імплементовані в платформу .NET.



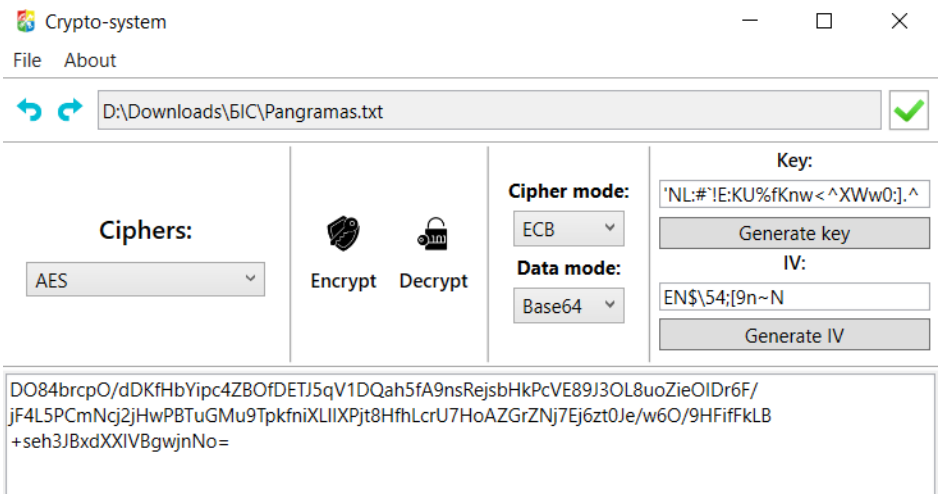
Всі дані формуються в двух форматах: або в шістнадцятковій системі, або в Base64 (на вибір):



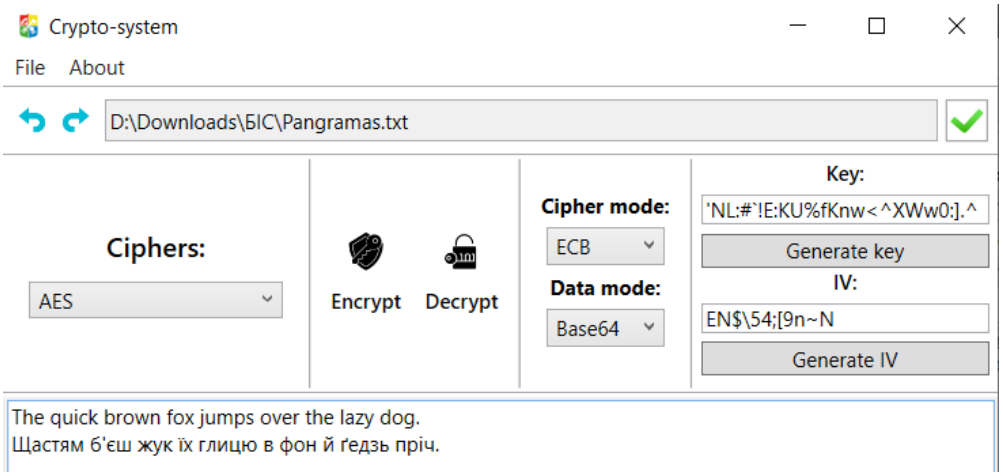
Для ключ та вектора ініціалізації наявні відповідні поля. Також, їх можна згенерувати автоматично.



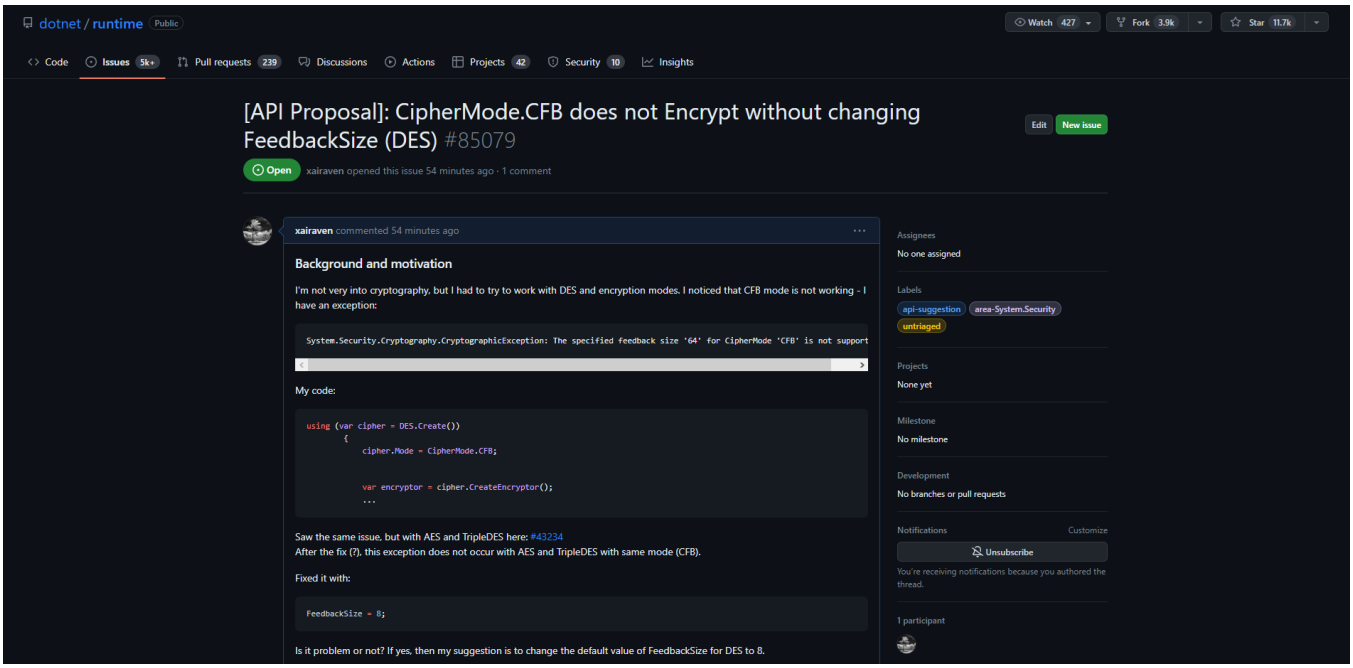
Приклад зашифрованого тексту:



Приклад розшифрованого тексту:



В процесі імплементації шифру DES виникла помилка – шифрорежим CFB працює лише коли змінна FeedbackSize встановлена як 8, а не 64 (за замовчуванням 64). Щодо знайденої помилки був сформований звіт та відправлений як тікет в Гітхаб репозиторій Microsoft (dotnet/runtime):



Також, варто зазначити, що запропонований шлях шифрування – за допомогою файлового потоку – є небезпечним. Файл може бути скомпрометований третьою стороною. В якості експерименту, було проведене підключення до віртуальної машини по SSH. Перша сторона зашифрувала певний текст, який був записаний у вигляді байтів в файл, і під час цього був проведений перехват за допомогою утиліти EasyUS Recovery Wizard. Щоб уникнути таких ситуацій, доцільно користуватись потоком MemoryStream.

Висновок: за результатами виконання цієї лабораторної роботи було ознайомлено з принципом роботи блокових шифрів. В крипто-систему були імплементовані такі шифри як AES, DES, TripleDES. Код був частково модифікований – були прибрані застарівші крипто-провайдери, також файловий потік був замінений на MemoryStream.