Лабораторна робота №5

Тема: Шифр DES

Мета: Ознайомитись з використанням криптопровайдерів в прикладному програмуванні

Базові відомості

Алгоритм симетричного шифрування DES (Data Encryption Standard) – стандарт симетричного шифрування США, розроблений у 1977 році, який згодом набув міжнародного застосування. Зараз DES вважається ненадійним в основному через малу довжину ключа.

З метою забезпечення більш високого рівня криптостійкості був запропонований модифікований метод з послідовним трициклічним шифруванням за алгоритмом DES, який отримав назву 3-DES (TripleDES). Криптостійкість методу виявилась значно кращою (про реалізацію успішних атак на 3-DES не відомо), проте швидкість шифрування значно зменшилась у порівнянні з DES (приблизно в 3 рази).

На сьогодні алгоритми DES та 3-DES поступово витісняються новітнім алгоритмом шифрування AES (Advanced Encryption Standard), який забезпечує як високий рівень криптостійкості, так і прийнятну швидкість шифрування.

Нові можливості для розробки криптографічних додатків надає бібліотека класів .NET Framework, яка включає класи криптопровайдерів для реалізації симетричного шифрування за чотирма алгоритмами: DES, TripleDES і AES, а також RC2 (який є попередником AES і залишений для забезпечення сумісності з попередніми версіями додатків). Реалізуються вони за допомогою об'єктів двох класів з простору імен **System.Security.Cryptography**:

- **CryptographicServiceProvider** – клас, що надає криптопровайдери для кожного з вказаних алгоритмів.
- **CryptoStream**– клас для роботи з криптографічним потоком.

Застосування цих основних об'єктів вимагає використання об'єкту **FileStream** з простору імен **System.IO**. Крім того, для байтового представлення текстових даних необхідні об'єкти класів **UnicodeEncoding** (або **ASCIIEncoding**) з простору імен **System.Text**.

Порядок шифрування за їх допомогою такий:

1. Створюється потрібний криптопровайдер, задаються його ключ, режим роботи і вектор ініціалізації:

   DESCryptoServiceProvider cryptic = **new DESCryptoServiceProvider**();

   cryptic.**Key** = **ASCIIEncoding**.**ASCII**.**GetBytes**("ABCDEFGH");

   cryptic.**Mode** = **CipherMode**.**CBC**;

   cryptic.**IV** = **ASCIIEncoding**.**ASCII**.**GetBytes**("ABCDEFGH");

2. Відкривається звичайний файловий потік для запису зашифрованих даних:

   FileStream stream = **new FileStream**(@"d:\test.txt", FileMode.OpenOrCreate,FileAccess.Write)

3. Відкритий файловий потік трансформується в крипто потік для запису:

   CryptoStream crStream = **new CryptoStream**(fs, cryptic.**CreateEncryptor**(),**CryptoStreamMode**.**Write**);

4. Дані для шифрування перетворюються у байтову послідовність і всі байти записуються в криптопотік за допомогою методу Write ():

   byte[] data = **UnicodeEncoding**.**Utf8**.**GetBytes**("Hello World!");

   crStream.**Write**(data,0,data.Length);

5. Використані файловий і крипто – потоки закриваються:

   crStream.**Close**();

   stream .**Close**();

Порядок розшифрування полягає у такому:

1. Створюється потрібний крипто провайдер, задаються його ключ, режим роботи і вектор ініціалізації :

   DESCryptoServiceProvider cryptic = **new DESCryptoServiceProvider**();

   cryptic.**Key** = **ASCIIEncoding**.**ASCII**.**GetBytes**("ABCDEFGH");

   cryptic.**IV** = **ASCIIEncoding**.**ASCII**.**GetBytes**("ABCDEFGH");

   cryptic.**Mode** = **CipherMode**.**CBC**;

2. Відкривається звичайний файловий потік для читання зашифрованих даних:

   FileStream stream = **new FileStream**(@"d:\test.txt", **FileMode**.**Open**,**FileAccess**.**Read**)

3. Відкритий файловий потік трансформується в криптопотік для читання:

   CryptoStream crStream = **new CryptoStream**(stream, cryptic.**CreateDecryptor**(),**CryptoStreamMode**.**Read**).

4. Дані з крипто потоку зчитуються за допомогою об’єкта StreamReader і присвоюються текстовій змінній:

   StreamReader reader = **new StreamReader**(crStream);

   string data = reader.**ReadToEnd**();

5. Значення текстової змінної виводиться на екран і зчитувач та потік закриваються

   Console.**WriteLine**(data);

   reader.**Close**();

   stream.**Close**();

Хід виконання роботи

1. Розробіть інтерфейс криптографічної системи для шифрування за допомогою DES з використанням всіх можливих режимів.

2. Ознайомтесь з описом класів **CryptographicServiceProvider** і **CryptoStream** бібліотеки .NET Framework.
3. Реалізуйте шифрування DES, використовуючи класи .NET Framework.
4. Виконайте тестування роботи системи.

Додаткові завдання

1. Модифікуйте створений програмний код для здійснення шифрування за алгоритмом TripleDES .
2. Модифікуйте створений програмний код для здійснення шифрування за алгоритмом AES.