6. Lớp luồng mã hóa CryptoStream

An ninh trong truyền gửi dữ liệu luôn cần được cân nhắc.

.NET thực sự cung cấp lớp CryptoStream qua lớp cha Stream. Tuy nhiên cũng có thể dùng nó như lớp con của không gian System.IO.

CryptoStream cần ba tham số. Thứ nhất là luồng cần đưa vào mã hóa. Thứ hai, thuật toán mã hóa. Tham số sau cùng chỉ ra cách truy xuất đọc hay ghi tới luồng.

Có nhiều thuật toán mã hóa khác nhau được cung cấp qua giao tiếp ICryptoTransform. Ví dụ sau đây lại lấy các thuật toán mã hóa từ không gian System.Security.Cryptography.

Ta bắt đầu ứng dụng bằng đoạn mã sau đây.

```
using System;
using System.IO;
using System.Text;
using System.Security.Cryptography;

public class crypt
{
   public static void Main()
   {
```

Trước hết, ta yêu cầu người dụng chọn một con số mà ứng dụng sẽ liệt kê. Các dịch vụ mã hóa thường được các nhà cung cấp dịch vụ (Service Provider) cung cấp qua lớp SymmetricAlgorithm và nó cho phép thi hành cả mã hóa và giải mã kèm theo một mật khóa.

```
        Console.WriteLine("Moi ban chon dich vu Ma Hoa:");
        Console.WriteLine("1 = DES");
        Console.WriteLine{"2 = RC2");
        Console.WriteLine("3 = AES");
        Console.WriteLine("4 = 3DES");
        Console.WriteLine("5 = Ma hoa doi xung co dien");


        // Tạo biến đối tượng mã hóa des, nó sẽ nhận một kiểu từ người dùng
        SymmetricAlgorithm des = null;
```

```
        switch (Console.ReadLine())
        {
          case "1": des = new DESCryptoServiceProvider();           break;
          case "2": des = new RC2CryptoServiceProvider();           break;
          case "3": des = new RijndaelManaged();                    break;
          case "4": des = new TripleDESCryptoServiceProvider();     break;
          case "5": des= SymmetricAlgorithm.Create();               break;
          default:
                Console.WriteLine ("Moi chon so tu 1 den 5 !!!");
          return;
        }
```

Tạo một đối tượng FileStream để lưu trữ dữ liệu mã hóa và bao gói nó trong đối tượng CryptoStream. Tạo giao tiếp ICryptoTransform bằng phương thức CreateEncryptor (thuộc lớp SymmetricAlgorithm) để xác định các hoạt động mã hóa cơ bản.

```
        FileStream fs = new FileStream("SecretFile.dat", FileMode.Create,
                                       FileAccess.Write);
        ICryptoTransform desencrypt = des.CreateEncryptor();
        CryptoStream cryptostream = new CryptoStream(fs, desencrypt,
                                            CryptoStreamMode.Write);
```

Bây giờ ta bắt đầu mã hóa một thông điệp đơn giản. Trước hết thông điệp được chuyển thành mảng byte bằng phương thức GetByte() của lớp Encoding trong không gian System.Text. Mảng này rồi sẽ được ghi vào CryptoStream bằng phương thức Write().

```
        string theMessage = "A top secret message";
        byte[] bytearrayinput = Encoding.Unicode.GetBytes(theMessage);
        Console.WriteLine("Original Message : {0} ", theMessage);
        cryptostream.Write(bytearrayinput, 0, bytearrayinput.Length);
        cryptostream.Close();
        fs.Close();
```

Sau khi đã đóng luồng fs thành công, ta có thể thực hiện giải mã thông điệp đã mã hóa. Dữ liệu mã hóa được lần lượt đọc từ file và chuyển đổi ngược lại như bản gốc.

```
/***********Thi hành giai mã***********/
        // Tạo luồng FileStream để đọc file mã hóa
```

```
        FileStream fsread = new FileStream("SecretFile.dat", FileMode.Open,
                                FileAccess.ReadWrite);
        byte [] encByte = new byte[fsread.Length ];
        fsread.Read(encByte,0,encByte.Length);
```

Ở đây, ta xác định chiều dài cần thiết của mảng byte bằng thuộc tính Length có trong luồng FileStream nhằm mục đích lấy hết dữ liệu có trong file. Lần lượt đọc dữ liệu từ file vào mảng bằng phương thức Read(). Trước khi giải mã, ta hiển thị toàn bộ nội dung mã hóa ra màn hình. Sau khi in ra màn hình, con trỏ luồng file ở cuối cùng, ta bắt buộc nó trở về vị trí đầu luồng.

```
        Console.WriteLine ("Encrypted Message : " +
                                Encoding.ASCII.GetString(encByte));
        fsread.Position = 0;
```

Thủ tục giải mã dữ liệu không khác nhiều so với mã hóa. Sự khác biệt chủ yếu ở đây chỉ là cách dùng phương thức CreateDecryptor(), nó tạo đối tượng giải mã. Tạo mới mảng byte để lưu trữ kết quả giải mã. Phương thức GetString() của lớp Encoding sẽ biến đổi mảng byte thành chuỗi.

```
      // Tạo đối tượng giải mã từ đối tượng des
      ICryptoTransform desdecrypt = des.CreateDecryptor();
      CryptoStream cryptostreamDecr = CryptoStream(fsread, desdecrypt,
                                                   CryptoStreamMode.Read);
      byte[] decrByte = new byte[fsread.Length];
      cryptostreamDecr.Read(decrByte,0,(int)fsread.Length);
      string output = Encoding.Unicode.GetString(decrByte);
      Console.WriteLine("Decrypted Message : {0}" ,output);
      cryptostreamDecr.Close();
      fsread.Close();
   }
}
```