Module 13

Encrypting and Decrypting Data

Module Overview

- Implementing Symmetric Encryption
- Implementing Asymmetric Encryption

Lesson 1: Implementing Symmetric Encryption

- What Is Symmetric Encryption?
- Encrypting Data by Using Symmetric Encryption
- Hashing Data
- Demonstration: Encrypting and Decrypting Data

What Is Symmetric Encryption?

- Symmetric encryption is the cryptographic transformation of data by using a mathematical algorithm
- The same key is used to encrypt and decrypt the data
- The **System.Security.Cryptography** namespace includes:
  - **DESCryptoServiceProvider** class
  - **AesManaged** class
  - **RC2CryptoServiceProvider** class
  - **RijndaelManaged** class
  - **TripleDESCryptoServiceProvider** class

Encrypting Data by Using Symmetric Encryption

To encrypt and decrypt data symmetrically, perform the following steps:

1. Create an **Rfc2898DeriveBytes** object
2. Create an **AesManaged** object
3. Generate a secret key and an IV
4. Create a stream to buffer the transformed data
5. Create a symmetric encryptor or decryptor object
6. Create a **CryptoStream** object
7. Write the transformed data to the buffer stream
8. Close the streams

Hashing Data

- A hash is a numerical representation of a piece of data
- Properties
  - Predictable hash size
  - Collision avoidance
- Application
  - Checksums
  - Hashtables
  - Storing passwords
  - Salting
  - Cryptocurrency

Hashing Data

- A hash can be computed by using the following code

```
public byte[] ComputeHash(byte[] dataToHash, byte[] secretKey)
{
   using (var hashAlgorithm = new HMACSHA1(secretKey))
   {
      using (var bufferStream = new MemoryStream(dataToHash))
      {
         return hashAlgorithm.ComputeHash(bufferStream);
      }
   }
}
```

Demonstration: Encrypting and Decrypting Data

In this demonstration, you will use symmetric encryption to encrypt and decrypt a message

Lesson 2: Implementing Asymmetric Encryption

- What Is Asymmetric Encryption?
- Encrypting Data by Using Asymmetric Encryption
- Creating and Managing X509 Certificates
- Managing Encryption Keys
- Demonstration: Encrypting and Decrypting Grade Reports Lab

What Is Asymmetric Encryption?

- To send a private message:
  - A public key to encrypt data
  - A private key to decrypt data
- Reverse to verify the sender
- Exchange public keys for private communication channel
- Slower than symmetric
- No need to send key!
- Hybrid – use asymmetric to encrypt a symmetric key

Encrypting Data by Using Asymmetric Encryption

To encrypt and decrypt data asymmetrically

```
var rawBytes = Encoding.Default.GetBytes("hello world..");
var decryptedText = string.Empty;

using (var rsaProvider = new RSACryptoServiceProvider())
{
    var useOaepPadding = true;

    var encryptedBytes =
        rsaProvider.Encrypt(rawBytes, useOaepPadding);

    var decryptedBytes =
        rsaProvider.Decrypt(encryptedBytes, useOaepPadding);

     decryptedText = Encoding.Default.GetString(decryptedBytes);
}
// decryptedText == hello world..
```

Creating and Managing X509 Certificates

- Use MakeCert to create certificates

```
makecert -n "CN=FourthCoffee" -a sha1 -pe -r -sr LocalMachine -
ss my -sky exchange
```

- Use the MMC Certificates snap-in to manage your certificate stores

Managing Encryption Keys

The **System.Security.Cryptography.X509Certificates** namespace contains classes that enable access to the certificate store and certificate metadata

```
var store = new X509Store(
   StoreName.My,
   StoreLocation.LocalMachine);

store.Open(OpenFlags.ReadOnly);

foreach (var storeCertificate in store.Certificates)
{
   // Code to process each certificate.
}

store.Close();
```

Demonstration: Encrypting and Decrypting Grade Reports Lab

In this demonstration, you will learn about the tasks that you will perform in the lab for this module

Lab: Encrypting and Decrypting the Grades Report

- Exercise 1: Encrypting the Grades Report
- Exercise 2: Decrypting the Grades Report

Estimated Time: 60 minutes

Lab Scenario

You have been asked to update the Grades application to ensure that reports are secure when they are stored on a user's computer. You decide to use asymmetric encryption to protect the report as it is generated, before it is written to disk. Administrative staff will need to merge reports for each class into one document, so you decide to develop a separate application that generates a combined report and prints it.

Module Review and Takeaways

- Review Questions

Course Evaluation

- Your evaluation of this course will help Microsoft understand the quality of your learning experience.
- Please work with your training provider to access the course evaluation form.
- Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.