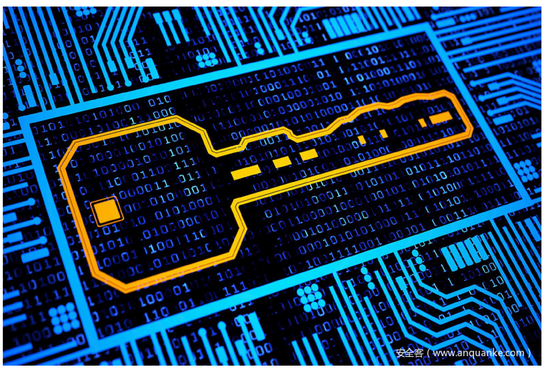
.net中各种加解密用这一个库都够了，BouncyCastle支持常用的各种加密解密算法，快试试吧

作者：微信公众号：【架构师老卢】 11-20 7:41



概述：BouncyCastle 是一个流行的 Java 加解密库，也支持在 .NET 平台上使用。下面是 BouncyCastle 在 .NET 下使用的一些常见功能，包括 AES、RSA、MD5、SHA1、DES、SHA256、SHA384、SHA512 等。

BouncyCastle 是一个流行的 Java 加解密库，也支持在 .NET 平台上使用。下面是 BouncyCastle 在 .NET 下使用的一些常见功能，包括 AES、RSA、MD5、SHA1、DES、SHA256、SHA384、SHA512 等。

在开始之前，请确保你已经将 BouncyCastle 的 NuGet 包安装到你的项目中。你可以通过 NuGet 包管理器控制台或 Visual Studio 中的 NuGet 包管理器进行安装。

```
Install-Package BouncyCastle
```

接下来，我将演示如何使用 BouncyCastle 实现一些常见的加解密操作。

1. AES 加解密

```
using System;
using System.Text;
using Org.BouncyCastle.Crypto;
using Org.BouncyCastle.Crypto.Engines;
using Org.BouncyCastle.Crypto.Modes;
using Org.BouncyCastle.Crypto.Parameters;

public class AesExample
{
    public static byte[] Encrypt(string plaintext, byte[] key, byte[] iv)
    {
        CipherEngine engine = new CipherEngine();
        CipherParameters keyParam = new KeyParameter(key);
        ParametersWithIV keyParamWithIV = new ParametersWithIV(keyParam, iv);

        engine.Init(true, keyParamWithIV);

        byte[] input = Encoding.UTF8.GetBytes(plaintext);
        byte[] output = new byte[engine.GetOutputSize(input.Length)];

        int len = engine.ProcessBytes(input, 0, input.Length, output, 0);
        engine.DoFinal(output, len);

        return output;
    }

    public static string Decrypt(byte[] ciphertext, byte[] key, byte[] iv)
    {
        CipherEngine engine = new CipherEngine();
        CipherParameters keyParam = new KeyParameter(key);
        ParametersWithIV keyParamWithIV = new ParametersWithIV(keyParam, iv);

        engine.Init(false, keyParamWithIV);

        byte[] output = new byte[engine.GetOutputSize(ciphertext.Length)];

        int len = engine.ProcessBytes(ciphertext, 0, ciphertext.Length, output, 0);
        engine.DoFinal(output, len);

        return Encoding.UTF8.GetString(output);
    }
}

// 示例用法
byte[] aesKey = new byte[16]; // AES 128-bit key
byte[] aesIV = new byte[16];  // AES 128-bit IV
string plaintext = "Hello, BouncyCastle!";

byte[] ciphertext = AesExample.Encrypt(plaintext, aesKey, aesIV);
string decryptedText = AesExample.Decrypt(ciphertext, aesKey, aesIV);

Console.WriteLine($"Plaintext: {plaintext}");
Console.WriteLine($"Ciphertext: {Convert.ToBase64String(ciphertext)}");
Console.WriteLine($"Decrypted Text: {decryptedText}");
```

2. RSA 加解密

```
using System;
using System.Text;
using Org.BouncyCastle.Crypto;
using Org.BouncyCastle.Crypto.Encodings;
using Org.BouncyCastle.Crypto.Engines;
using Org.BouncyCastle.Crypto.Parameters;
using Org.BouncyCastle.Security;

public class RsaExample
{
    public static byte[] Encrypt(string plaintext, AsymmetricKeyParameter publicKey)
    {
        CipherEngine engine = new CipherEngine();
        engine.Init(true, publicKey);

        byte[] input = Encoding.UTF8.GetBytes(plaintext);
        byte[] output = engine.ProcessBytes(input, 0, input.Length);

        return output;
    }

    public static string Decrypt(byte[] ciphertext, AsymmetricKeyParameter privateKey)
    {
        CipherEngine engine = new CipherEngine();
        engine.Init(false, privateKey);

        byte[] output = engine.ProcessBytes(ciphertext, 0, ciphertext.Length);

        return Encoding.UTF8.GetString(output);
    }
}

// 示例用法
RsaKeyPairGenerator rsaKeyPairGen = GeneratorUtilities.GetKeyPairGenerator("RSA");
rsaKeyPairGen.Init(new KeyGenerationParameters(new SecureRandom(), 2048)); // 2048-bit key size
AsymmetricCipherKeyPair keyPair = rsaKeyPairGen.GenerateKeyPair();

AsymmetricKeyParameter publicKey = keyPair.Public;
AsymmetricKeyParameter privateKey = keyPair.Private;

string plaintext = "Hello, BouncyCastle!";

byte[] ciphertext = RsaExample.Encrypt(plaintext, publicKey);
string decryptedText = RsaExample.Decrypt(ciphertext, privateKey);

Console.WriteLine($"Plaintext: {plaintext}");
Console.WriteLine($"Ciphertext: {Convert.ToBase64String(ciphertext)}");
Console.WriteLine($"Decrypted Text: {decryptedText}");
```

3. MD5、SHA1、SHA256、SHA384、SHA512

```
using System;
using System.Security.Cryptography;
using System.Text;
using Org.BouncyCastle.Crypto.Digests;

public class HashExample
{
    public static string ComputeMD5(string input)
    {
        MD5 md5 = MD5.Create();
        byte[] hashBytes = md5.ComputeHash(Encoding.UTF8.GetBytes(input));
        return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
    }

    public static string ComputeSHA1(string input)
    {
        SHA1 sha1 = SHA1.Create();
        byte[] hashBytes = sha1.ComputeHash(Encoding.UTF8.GetBytes(input));
        return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
    }

    public static string ComputeSHA256(string input)
    {
        Sha256Digest sha256 = new Sha256Digest();
        byte[] inputBytes = Encoding.UTF8.GetBytes(input);
        sha256.BlockUpdate(inputBytes, 0, inputBytes.Length);

        byte[] hashBytes = new byte[sha256.GetDigestSize()];
        sha256.DoFinal(hashBytes, 0);

        return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
    }

    public static string ComputeSHA384(string input)
    {
        Sha384Digest sha384 = new Sha384Digest();
        byte[] inputBytes = Encoding.UTF8.GetBytes(input);
        sha384.BlockUpdate(inputBytes, 0, inputBytes.Length);

        byte[] hashBytes = new byte[sha384.GetDigestSize()];
        sha384.DoFinal(hashBytes, 0);

        return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
    }

    public static string ComputeSHA512(string input)
    {
        Sha512Digest sha512 = new Sha512Digest();
        byte[] inputBytes = Encoding.UTF8.GetBytes(input);
        sha512.BlockUpdate(inputBytes, 0, inputBytes.Length);

        byte[] hashBytes = new byte[sha512.GetDigestSize()];
        sha512.DoFinal(hashBytes, 0);

        return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
    }
}

// 示例用法
string input = "Hello, BouncyCastle!";
Console.WriteLine($"MD5: {HashExample.ComputeMD5(input)}");
Console.WriteLine($"SHA1: {HashExample.ComputeSHA1(input)}");
Console.WriteLine($"SHA256: {HashExample.ComputeSHA256(input)}");
Console.WriteLine($"SHA384: {HashExample.ComputeSHA384(input)}");
```

```
Console.WriteLine($"SHA512: {HashExample.ComputeSHA512(input)}");
```

这些示例展示了在 .NET 下使用 BouncyCastle 实现 AES、RSA、MD5、SHA1、SHA256、SHA384、SHA512 加解密的基本操作。

请注意，具体的实现细节可能根据 BouncyCastle 版本略有变化，建议查阅 BouncyCastle 的官方文档以获取最新信息。