RSA应用比较多的在于签名上

使用私钥进行签名，公钥验证签名

public class RSAHelper

{

    /// <summary>

    /// 签名

    /// </summary>

    /// <param name="privateKey"></param>

    /// <param name="content"></param>

    /// <returns></returns>

    public static string Signature (string privateKey, string content)

    {

        string signatureContent = string.Empty;

        using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider ()) {

            // 导入密钥

            rsa.ImportCspBlob (Convert.FromBase64String (privateKey));

            // 生成签名

            SHA1 sha1 = new SHA1CryptoServiceProvider ();

            byte[] signatureData = rsa.SignData (Encoding.Default.GetBytes (content), sha1);

            // 签名文本编码为 base64

            signatureContent = Convert.ToBase64String (signatureData);

        }

        return signatureContent;

}

    /// <summary>

    /// 验证签名

    /// </summary>

    /// <param name="publicKey"></param>

    /// <param name="content"></param>

    /// <param name="signatureText"></param>

    /// <returns></returns>

    public static bool VaildSignature (string publicKey, string content, string signatureText)

    {

        using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider ())

        {

            // 导入公钥

            rsa.ImportCspBlob (Convert.FromBase64String (publicKey));

            byte[] signatureData = Convert.FromBase64String (signatureText);

            // 验证签名

            SHA1 sha1 = new SHA1CryptoServiceProvider ();

            return rsa.VerifyData (Encoding.Default.GetBytes (content), sha1, signatureData);

        }

}

    /// <summary>

    /// 生成密钥

    /// </summary>

    /// <param name="keySize">384位到16384位，以8为倍数</param>

    /// <returns>密钥Key，公钥Value</returns>

    public static KeyValuePair<string, string> GenerateRSASecretKey (int keySize)

    {

        KeyValuePair<string, string> rsaKey;

        using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider (keySize)) {

            rsaKey = new KeyValuePair<string, string> (

                // 导出Blob，此结构包含了密钥信息

                Convert.ToBase64String (rsa.ExportCspBlob (true)),

                // 导出Blob，此结构包含了公钥信息

                Convert.ToBase64String (rsa.ExportCspBlob (false)));

        }

        return rsaKey;

    }

}

// 示例

static void Main(string[] args)

{

    var key = RSAHelper.GenerateRSASecretKey(384);

var signatureText = RSAHelper.Signature(key.Key, "hello world");

var isPass = RSAHelper.VaildSignature(key.Value, "hello world", signatureText);

    Console.WriteLine("Hello World!");

}