

软件安全之道：利用RSA加密打造强大License验证，确保软件正版合法运行！

作者：微信公众号：【架构师老卢】

11-29 10:0

1240



概述： C#软件开发中， License扮演着确保软件合法使用的重要角色。采用RSA非对称加密方案， 服务端生成带签名的License， 客户端验证其有效性， 从而实现对软件的授权与安全保障。

License应用场景：

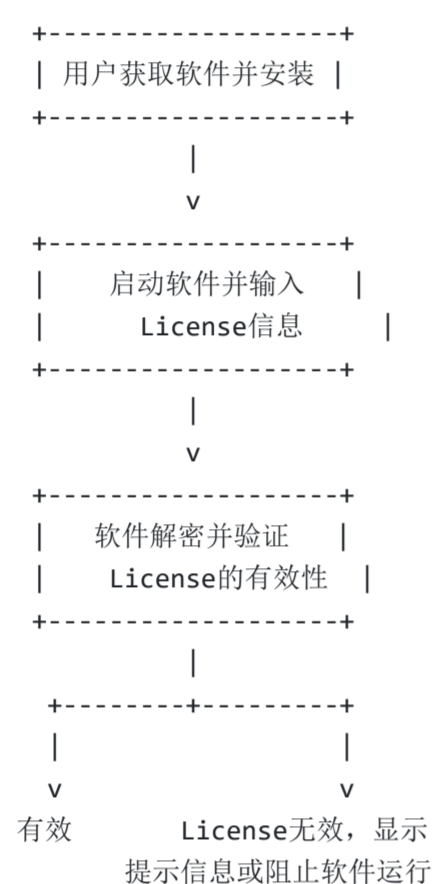
License（许可证）在C#软件开发中被广泛应用，以确保软件在合法授权的环境中运行。常见场景包括商业软件、桌面应用、服务端应用等。

License实现方案：

一种常见的License实现方案是使用非对称加密技术，将License信息加密，并在软件中内置公钥，从而确保只有使用私钥签名的License才会被验证通过。

License验证流程图：

以下是一个简单的License验证流程图：



主要功能代码：

以下是一个简单的C#示例，演示了使用RSA非对称加密进行License验证的基本实现。示例中包含服务端和客户端的代码。

服务端（生成License）：

```
1 using System.Security.Cryptography;
2 using System.Text;
3
4 public class LicenseGenerator
5 {
6     // 生成License的方法
7     public string GenerateLicense()
8     {
9         using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider())
10        {
11            // 生成公钥和私钥
12            string publicKey = rsa.ToXmlString(false);
13            string privateKey = rsa.ToXmlString(true);
14
15            // License信息（模拟）
16            string licenseInfo = "ValidLicenseInfo";
17
18            // 使用私钥对License信息进行签名
19            byte[] signature = rsa.SignData(Encoding.UTF8.GetBytes(licenseInfo), new SHA256CryptoServiceProvider());
20
21            // 将公钥、License信息和签名组合成License
22            string license = $"{publicKey};{licenseInfo};{Convert.ToBase64String(signature)}";
23
24            return license;
25        }
26    }
27 }
```

客户端（验证License）：

```
1 using System.Security.Cryptography;
2 using System.Text;
3
4 public class LicenseValidator
5 {
6     // 验证License的方法
7     public bool ValidateLicense(string userEnteredKey)
8     {
9         // 将License拆分成公钥、License信息和签名
10        string[] parts = userEnteredKey.Split(';');
11        string publicKey = parts[0];
12        string licenseInfo = parts[1];
13        byte[] signature = Convert.FromBase64String(parts[2]);
14
15        using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider())
16        {
17            // 设置公钥
18            rsa.FromXmlString(publicKey);
19
20            // 使用公钥验证License信息的签名
21            return rsa.VerifyData(Encoding.UTF8.GetBytes(licenseInfo), new SHA256CryptoServiceProvider(), signature);
22        }
23    }
24 }
```

使用示例：

```
1 public class Application
2 {
3     public static void Main()
4     {
5         LicenseGenerator licenseGenerator = new LicenseGenerator();
6         LicenseValidator licenseValidator = new LicenseValidator();
7
8         // 服务端生成License
9         string generatedLicense = licenseGenerator.GenerateLicense();
10
11        // 客户端输入License
12        Console.WriteLine("请输入License: ");
13        string userEnteredLicense = Console.ReadLine();
14
15        // 客户端验证License
16        if (licenseValidator.ValidateLicense(userEnteredLicense))
17        {
18            Console.WriteLine("License验证通过，软件已启动。");
19            // 软件正常运行逻辑...
20        }
21        else
22        {
23            Console.WriteLine("License验证失败，无法启动软件。");
24        }
25    }
26 }
```

```
24     }  
25 }  
26 }
```

上述代码演示了使用RSA非对称加密进行License的生成和验证。上只是提供一个思路，在实际应用中，公钥和私钥需要安全存储，以确保系统的安全性。