

ASP.NET Core 数据保护系统

2019年3月20日 11:23

■ 解决何种问题？

真实性、保密性和隔离性。要求：简单，可灵活配置，可扩展。

■ 微软数据保护源码

来自 <<https://github.com/aspnet/AspNetCore/tree/master/src/DataProtection>>

抽象：IDataProtectionProvider 接口和 IDataProtector 接口

最终：密钥存储、可扩展 和 具体实现。

■ 最简单的使用方法

1. 在依赖注入容器中注入数据保护相关实现（可选）
2. 使用数据保护提供程序创建数据保护程序
3. 调用 Protect 加密数据
4. 调用 Unprotect 解密数据

```
Install-Package Microsoft.Extensions.DependencyInjection -Version 2.2.0
```

```
Install-Package Microsoft.AspNetCore.DataProtection -Version 2.2.0
```

```
// add data protection services
var serviceCollection = new ServiceCollection();
serviceCollection.AddDataProtection();
var services = serviceCollection.BuildServiceProvider();

var provider = services.GetRequiredService<IDataProtectionProvider>();
var protector = provider.CreateProtector("xcode.test");

//var provider = services.GetDataProtectionProvider();
//var protector = services.GetDataProtector("xcode.test");

string input = "this is xcode.me test";

// protect the payload
string protectedPayload = protector.Protect(input);

// unprotect the payload
string unprotectedPayload = protector.Unprotect(protectedPayload);
```

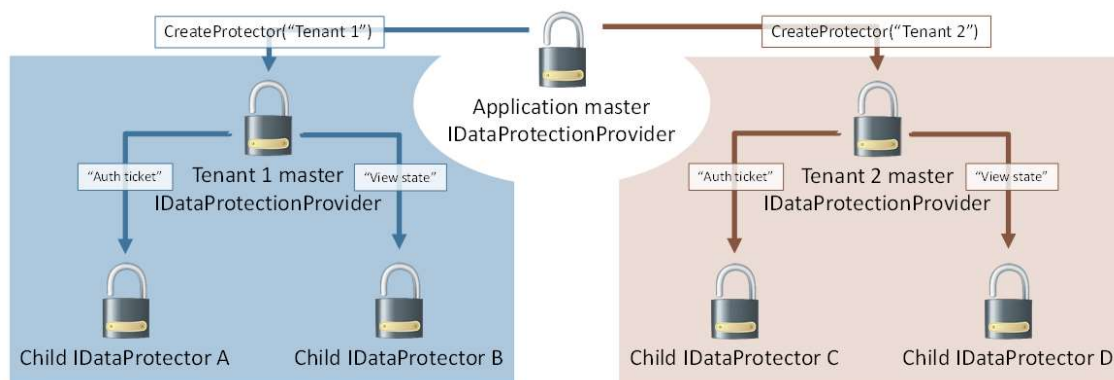
密钥保存位置：%LOCALAPPDATA%\ASP.NET\DataProtection-Keys

■ 加密器名称的层级结构

为了加密器的隔离性，加密器名称支持层级结构，可创建层级结构划分加密用途。



微软推荐：使用命名空间和类型名称是一个很好的经验法则。



```
provider.CreateProtector("purpose1").CreateProtector("purpose2")
```

■ 哈希密码最佳实践

[Password-Based Cryptography Specification Version 2.0](#)

Install-Package Microsoft.AspNetCore.Cryptography.KeyDerivation

微软开发的单独包，依赖于 PBKDF2 算法实现，不依赖于数据保护系统。

```
string password = "www.xcode.me";

// generate a 128-bit salt using a secure PRNG
byte[] salt = new byte[128 / 8];
using (var rng = RandomNumberGenerator.Create())
{
    rng.GetBytes(salt);
}
string saltString = Convert.ToBase64String(salt);

// derive a 256-bit subkey (use HMACSHA1 with 10,000 iterations)
var hashedBytes = KeyDerivation.Pbkdf2(password, salt, prf:
```

```
KeyDerivationPrf.HMACSHA1, 10000, 256 / 8);  
string hashedString = Convert.ToBase64String(hashedBytes);
```

微软在 ASP.NET CORE Identity 中就是用 PBKDF2 算法哈希密码，可参见源码：

来自 <<https://github.com/aspnet/AspNetCore/blob/master/src/Identity/Extensions.Core/src/PasswordHasher.cs>>

零度提取主要方法便于使用：

来自 <<https://github.com/dreamdancer/CommonClass/blob/master/PasswordHasher.cs>>

■ 加密器的生命周期

ITimeLimitedDataProtector 接口实现

```
// convert the normal protector into a time-limited protector  
var timeLimitedProtector = baseProtector.ToTimeLimitedDataProtector();  
  
// get some input and protect it for five seconds  
string protectedData = timeLimitedProtector.Protect(input, lifetime:  
    TimeSpan.FromSeconds(5));
```

密钥销毁后如何找回丢失的数据：IPersistedDataProtector

短暂性数据保护程序：EphemeralDataProtectionProvider

■ 配置数据保护程序

初始化加密器时，系统会基于当前机器的运行环境默认配置，但是有些时候可能需要对这些配置做一些改变，比如：在分布式部署的时候，集中存储密钥。

Services.AddDataProtection() 返回 IDataProtectionBuilder 进行链式配置。

密钥存储位置配置

Azure: %HOME%\ASP.NET\DataProtection-Keys

windows: %LOCALAPPDATA%\ASP.NET\DataProtection-Keys

IIS: HKLM 注册表中

其它：如果以上都不符合，进程关闭的时候，生成的私钥就丢失了。

默认支持：PersistKeysToFileSystem 文件系统和 PersistKeysToRegistry 注册表

Microsoft.AspNetCore.DataProtection.StackExchangeRedis

Microsoft.AspNetCore.DataProtection.AzureStorage

Microsoft.AspNetCore.DataProtection.AzureKeyVault

Microsoft.AspNetCore.DataProtection.EntityFrameworkCore

[SqlServerDataProtectionProvider](#)

ProtectKeysTo* : 配置密钥的存储位置
ProtectKeysWith* : 对密钥进行加密
SetDefaultKeyLifetime: 默认密钥有效期 90 天
SetApplicationName: 应用程序隔离
DisableAutomaticKeyGeneration: 禁用自动
UseCryptographicAlgorithms: 更改算法
UseCustomCryptographicAlgorithms: 自定义算法

计算机范围的策略

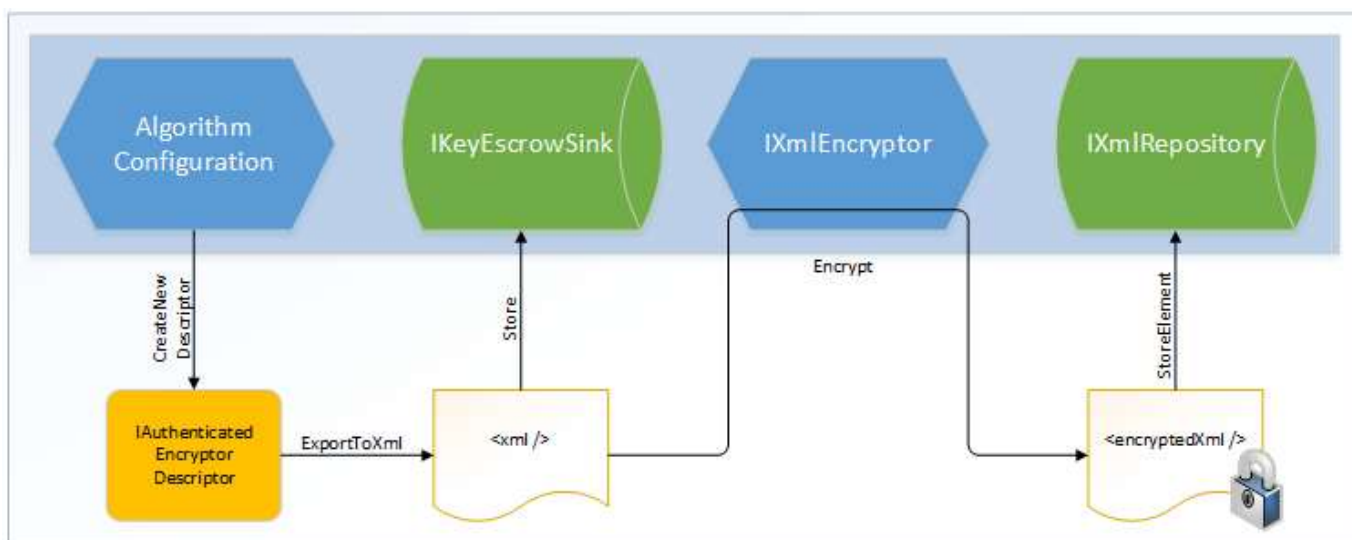
[Data Protection machine-wide policy support in ASP.NET Core](#)

非依赖注入场景下的数据保护

[Non-DI aware scenarios for Data Protection in ASP.NET Core](#)

■ 数据保护程序的扩展性

- IAuthenticatedEncryptor
- IAuthenticatedEncryptorDescriptor
- IAuthenticatedEncryptorDescriptorDeserializer
- AuthenticatedEncryptorConfiguration
- IKeyManager
- XmlKeyManager
- IXmlRepository
- IXmlEncryptor
- IXmlDecryptor
- IKeyEscrowSink



■ 扩展阅读资料

微软示例代码 <<https://github.com/aspnet/AspNetCore/tree/master/src/DataProtection/samples>>

中文博客学习 <<https://www.cnblogs.com/savorboard/tag/dotnet%20core%20data%20protection/>>

加密 ASP.NET CORE 参数扩展 <<https://github.com/WeihanLi/WeihanLi.DataProtection>>