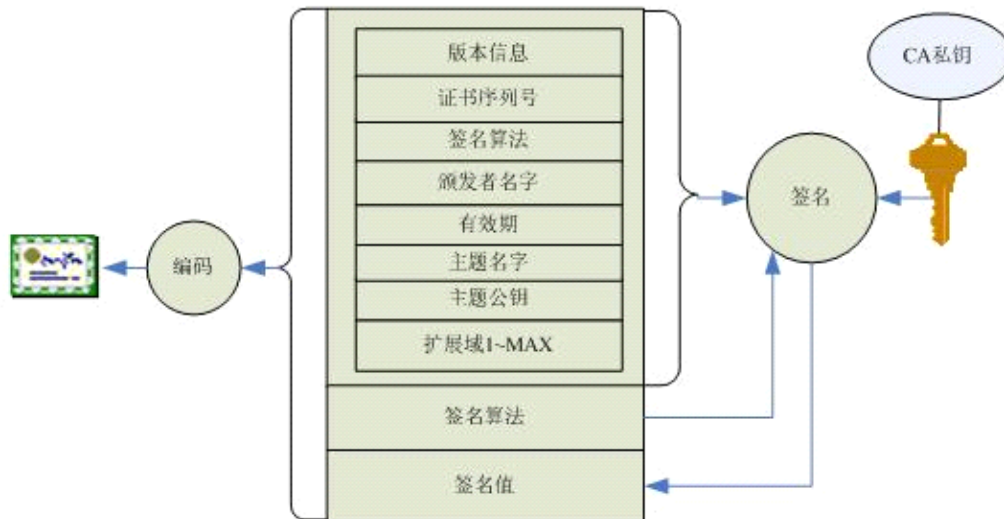


第86期-在gRPC中使用数字证书身份认证

2020年1月22日 11:52

证书管理工具



Certmgr.exe 是证书管理工具，安装 Visual Studio 时，将会自动安装证书管理器。若要启动该工具，在命令行键入 certmgr 命令。

MMC 可添加当前机器证书工作单元。

<https://docs.microsoft.com/zh-cn/dotnet/framework/tools/certmgr-exe-certificate-manager-tool>

证书创建工具

MakeCert

<https://docs.microsoft.com/zh-cn/windows/win32/seccrypto/makecert>

New-SelfSignedCertificate

<https://docs.microsoft.com/en-us/powershell/module/pkiclient/new-selfsignedcertificate>

OpenSSL

<https://github.com/openssl/openssl>

服务端验证证书

```
services.AddAuthentication(CertificateAuthenticationDefaults.AuthenticationScheme)
    .AddCertificate(options =>
```

```

    {
        // Not recommended in production environments. The example is
        using a self-signed test certificate.
        options.RevocationMode = X509RevocationMode.NoCheck;
        options.AllowedCertificateTypes = CertificateTypes.All;
    });

public class CertifierService : Certifier.CertifierBase
{
    [Authorize(AuthenticationSchemes =
CertificateAuthenticationDefaults.AuthenticationScheme)]
    public override Task<CertificateInfoResponse> GetCertificateInfo(Empty
request, ServerCallContext context)
    {
        var httpContext = context.GetHttpContext();
        var clientCertificate = httpContext.Connection.ClientCertificate;

        Console.WriteLine(clientCertificate);

        var name = string.Join(',',
context.AuthContext.PeerIdentity.Select(i => i.Value));
        var certificateInfo = new CertificateInfoResponse
        {
            HasCertificate = context.AuthContext.IsPeerAuthenticated,
            Name = name
        };

        return Task.FromResult(certificateInfo);
    }
}

```

客户端提供证书

```

private static HttpClient CreateHttpClient(bool includeClientCertificate)
{
    var handler = new HttpClientHandler();

    if (includeClientCertificate)
    {
        // Load client certificate
        var basePath =
Path.GetDirectoryPath(typeof(Program).Assembly.Location);
        var certPath = Path.Combine(basePath!, "Certs", "client.pfx");
        var clientCertificate = new X509Certificate2(certPath, "1111");
        handler.ClientCertificates.Add(clientCertificate);
    }

    // Create client
    return new HttpClient(handler);
}

```

```
var channel = GrpcChannel.ForAddress("https://localhost:5001", new
GrpcChannelOptions
    {
        HttpClient =
CreateHttpClient(includeClientCertificate)
    });
```