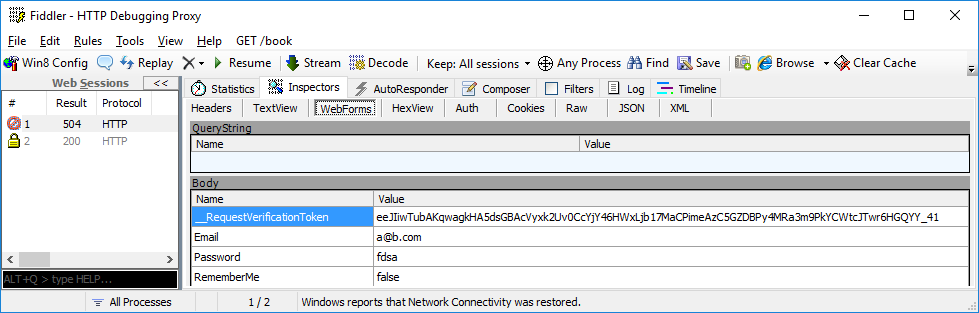
**ASP.NET MVC中使用JS实现不对称加密密码传输**

摘要： ASP.NET MVC中登录页面中点击登录后，用户名、密码将被明文传输到Controller中，使用Fiddler等工具可以轻松截获并获取密码， 这是不安全的。 使用对称加密，如AES，密钥将被暴露前端代码，也是不安全的。使用不对称加密能够较好解决这个问题。本文以RSA不对称加密的形式，在JS端通过公钥对密码进行加密，将密文传输到后端后通过密钥进行解密。

关键字： 不对称加密; 对称加密; RSA 算法; AES; 密钥; 公钥

**0 背景**

登录是最常见的需求之一，在这个环节，安全问题不可避免，明文传输很容易被截获并暴露密码原文。如下图使用Fiddle中出现的情况。



为了避免这种情况，通常办法有1 使用HTTPS形式解决； 2 使用公钥和不对称加密对密文进行加密; 3使用对称加密，比如AES。

这3种方案中，方案1是终极方案，但是需要克服证书获取和配置的问题， 本方案不是本文讨论重点，请有兴趣的自行查阅<https://letsencrypt.org/>。方案3， 以AES加密为例，必须把加密密钥存放在前端。 而前端对用户来说是开源的，很多开发者尝试把密钥藏的路径很深，但无疑这还是自欺欺人的。

方案2中，在JS端进行密码的RSA加密是有必要的，因为密码需要在用户点击“登录”按钮后被提交到服务器，这个过程被截获是很容易的。同时，防范CSRF类型攻击的特性也必须保留。这就要求：必须使用AJAX在JS端对密码加密，并向后台的Account Controller中的Login Action发起Post请求。而不能使用传统的Form Submit的方案。

AJAX post请求中，需要注意问题: 由于需要防范CSRF攻击的同时保障密文传输安全。需要同时顾及如下问题

问题1： 如何通过AJAX向Controller发起ajax请求？

问题2：如何在ajax请求中加入AntiForgeryToken？

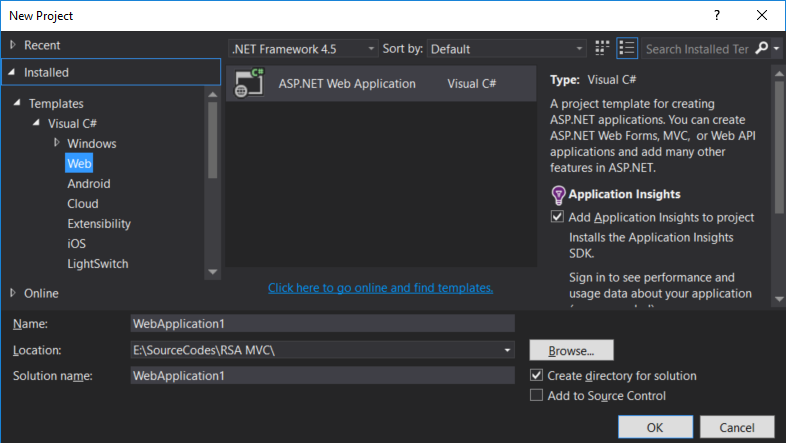
问题3： AJAX请求前，如何对密码进行RSA加密？

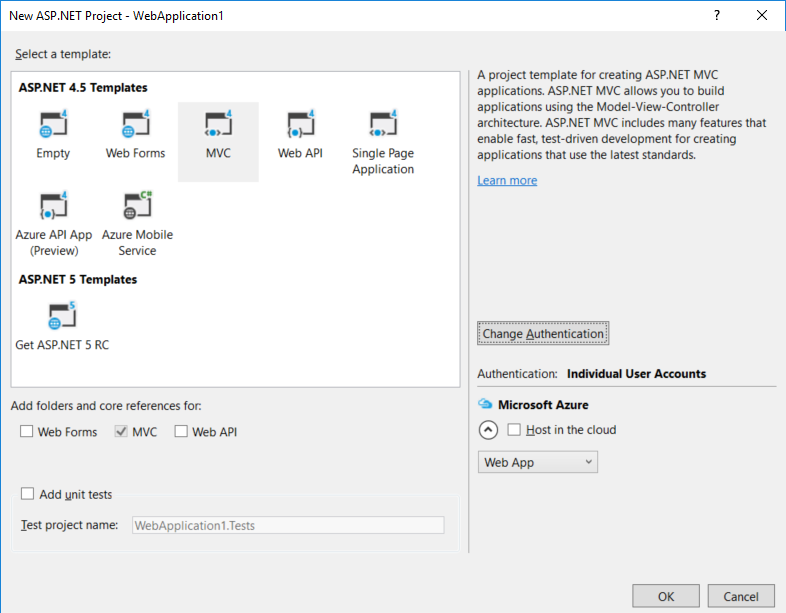
问题4： RSA的key format 有两种， pem格式和C#所支持的XML格式，通常JS支持pem, C#支持xml， 如何转换？

带着问题，进入操作步骤；

详细代码，请参考GitHub:

**1 操作步骤**

* 1. 新建Web Application
  2. 选择MVC， Authentication中选择”Individual User Accounts”



* 1. **Views** -> Account->Login.cshtml中代码修改如下

**@using WebApplicationRSA.Models**

**@model LoginViewModel**

**@{**

**ViewBag.Title = "Log in";**

**}**

**<h2>@ViewBag.Title.</h2>**

**<div class="row">**

**<div class="col-md-8">**

**<section id="loginForm">**

**<input type="hidden" id="AccountLoginURL" value=@Url.Action("Login", "Account") />**

**@Html.AntiForgeryToken()**

**<h4>Use a local account to log in.</h4>**

**<hr />**

**@Html.ValidationSummary(true, "", new { @class = "text-danger" })**

**<div class="form-group">**

**@Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })**

**<div class="col-md-10">**

**@Html.TextBoxFor(m => m.Email, new { @class = "form-control",id= "userNameTextBox" })**

**@Html.ValidationMessageFor(m => m.Email, "", new { @class = "text-danger" })**

**</div>**

**</div>**

**<div class="form-group">**

**@Html.LabelFor(m => m.Password, new { @class = "col-md-2 control-label" })**

**<div class="col-md-10">**

**@Html.PasswordFor(m => m.Password, new { @class = "form-control", id = "passwordTextBox" })**

**@Html.ValidationMessageFor(m => m.Password, "", new { @class = "text-danger" })**

**</div>**

**</div>**

**<div class="form-group">**

**<div class="col-md-offset-2 col-md-10">**

**<div class="checkbox">**

**@Html.CheckBoxFor(m => m.RememberMe)**

**@Html.LabelFor(m => m.RememberMe)**

**</div>**

**</div>**

**</div>**

**<div class="form-group">**

**<div class="col-md-offset-2 col-md-10">**

**<input type="submit" value="Log in" class="btn btn-default" id="login" />**

**</div>**

**</div>**

**<p>**

**@Html.ActionLink("Register as a new user", "Register")**

**</p>**

**<p>**

**@Html.Label("PublicKey", File.ReadAllText(AppDomain.CurrentDomain.BaseDirectory + @"/Content/PublicKey.pem"), new { id= "PublicKey" })**

**</p>**

**</section>**

**</div>**

**<div class="col-md-4">**

**<section id="socialLoginForm">**

**@Html.Partial("\_ExternalLoginsListPartial", new ExternalLoginListViewModel { ReturnUrl = ViewBag.ReturnUrl })**

**</section>**

**</div>**

**</div>**

**@section Scripts {**

**@Scripts.Render("~/bundles/jqueryval")**

**}**

**<script src="~/Scripts/jquery-1.10.2.min.js"></script>**

**<script src="~/Scripts/jsencrypt.js"></script>**

**<script src="~/Scripts/Login.js"></script>**

1.4 Scripts文件夹下新增 Login.js, jsencrypt.js文件